Modelling of Complex Electromagnetic Structures with TLM
— Theory and Practice —

Poman Pok-Man So

B. Sc.	University of Toronto,	1985
B. A. Sc.	University of Ottawa,	1987
M. A. Sc.	University of Ottawa,	1989

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy

in the Department of Electrical and Computer Engineering

University of Victoria

We accept this dissertation as conforming to the required standard

Prof. W. J. R. Hoefer	Supervisor (Department of Electrical and Computer Engineering)
Prof. M.A. Stuchly	Member (Department of Electrical and Computer Engineering)
Prof. R. Vahldieck	Member (Department of Electrical and Computer Engineering)
Prof. D. M. Miller	Outside Member (Department of Computer Science)
Prof. P. Russer	External Examiner (Technische Universität München, Germany)

Abstract

To design high frequency complex electromagnetic structures with TLM is not trivial in spite of the simplicity of the fundamental TLM algorithm. This is because TLM is a time and space discretization method in which the entire computational domain must be filled with nodes. In three-dimensional cases, the computational effort to solve realistic problems would soon become intractable unless special techniques are used. To be attractive to the design engineer, field simulation tools must include these advanced techniques automatically in the models. New TLM features and computational techniques have been developed in this thesis to overcome the above mentioned problems. An experimental multi-purpose electromagnetic field simulation tool has also been created to demonstrate the features and techniques developed in this thesis can be easily integrated into a well designed tool.

Examiners

Prof. W. J. R. Hoefer	Supervisor (Department of Electrical and Computer Engineering)
Prof. M.A. Stuchly	Member (Department of Electrical and Computer Engineering)
Prof. R. Vahldieck	Member (Department of Electrical and Computer Engineering)
Prof. D. M. Miller	Outside Member (Department of Computer Science)
Prof. P. Russer	External Examiner (Technische Universität München, Germany)

Dedication

To my parents, wife and children.

Preface

The transmission line matrix (TLM) method has been around for more than two decades. But so far, to design high frequency complex electromagnetic structures with TLM is a challenging task in spite of the simplicity of the fundamental TLM algorithm. This is because TLM is a time and space discretization method in which the entire computational domain must be filled with TLM cells. In three-dimensional cases, the computational effort to solve realistic problems soon becomes intractable unless special techniques are used.

New theory for special nodes and meshes for the representation of fine geometrical features through local modification have been developed. This improves the efficiency, accuracy and versatility of the TLM method. Therefore, an unnecessarily fine TLM mesh does not have to be used to model structures with small details. In addition to that, new algorithms based on parallel and distributed computing as well as signal processing techniques have also been developed to improve the computational efficiency of TLM. To be attractive to the design engineer, field simulation tools must include these advanced techniques automatically in the models. Many of these features have been incorporated in a state-of-the-art multipurpose electromagnetic field simulator which I have developed recently. The simulator also has automatic mesh generation and engineering parameter extraction capabilities; it can also be integrated with external programs so that TLM can be used strategically in conjunction with traditional linear circuit simulators to perform

analysis and optimization. These contribution will make the design of high frequency complex electromagnetic structures with TLM a big step closer to reality.

The research for this thesis started in January 1989. At that time, I was a research engineer at the University of Ottawa, Ottawa, Ontario, Canada. Some original contributions such as recursive generation of Johns Matrix and corner node were published before I started my Ph.D. studies. I also implemented a number of simulators based on 2D-TLM and 3D-TLM for DOS and the X Window system. One of these simulators was published as a book (W.J.R. Hoefer and P.P.M. So, *The Electromagnetic Wave Simulator — A Dynamic Visual Electromagnetics Laboratory based on the Two-Dimensional TLM Method*, John Wiley and Sons Inc., 1991) and a 3D-TLM simulator for the X Window system was distributed to various research institutes and high technology companies for evaluation. I have thus gained invaluable insight into TLM modelling as well as experience in interfacing TLM with various existing microwave engineering CAD tools. I joined the University of Victoria, Victoria, British Columbia, Canada, in June 1992 and started my Ph.D. studies in the September of the same year. In order to verify the new ideas and procedures developed in this thesis, a multipurpose electromagnetic field simulation tool has been implemented using the object oriented design paradigm with the C++ programming language. Most of the simulation results shown in this thesis are obtained using this prototype simulator. It is my desire that the new methods and procedures developed in this thesis can be transferred to the microwave engineering CAD industry as soon as this thesis is published.

Poman Pok-Man So
University of Victoria
June 14, 1996.

Acknowledgments

The research for this thesis was completed with the help of many people. I owe the pioneers and researchers of the TLM method a great debt; in particular to my supervisor professor W.J.R. Hoefer. His special insight about the capability of TLM as well as the need to incorporate the method into a simulation tool inspired me to start my TLM research and development work. In the past seven years, he has been given me many invaluable ideas about the theoretical and practical issues of the method; his unfailing support during the course of my studies is the most important building block of the multi-purpose electromagnetic field simulation tool, *Mefisto*, that has been created to validate the new features developed in this thesis. I would also like to thank Dr. Eswarappa for his help in preparing the material for Prony's method and ARMA modelling.

This work was financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), MPR Teltech Inc., the Science Council of British Columbia, and the University of Victoria.

Last but not least, it is the creator of this universe who said — *Let there be light!* — right at the beginning so that there is electromagnetic wave for us to research and simulate.

Poman Pok-Man So
University of Victoria
June 14, 1996.

Table of Contents

List of Figures

List of Tables

Refereed Publications By the Author

Book

[1] W.J.R. Hoefer and P.P.M. So, *The Electromagnetic Wave Simulator — A Dynamic Visual Electromagnetics Laboratory based on the Two-Dimensional TLM Method*, John Wiley & Sons Inc., 1991.

Journal and Conference Papers

[1] P.P.M. So and W.J.R. Hoefer, *Recent Advances in Transmission Line Matrix Method Electromagnetic Wave Modelling and Visualization*, First International Workshop on Transmission Line Matrix (TLM) Modeling — Theory and Applications, pp.183-186, August 1-3, 1995, Victoria, British Columbia, Canada.

[2] P.P.M. So, C. Eswarappa and W.J.R. Hoefer, *Distributed Parallel TLM Computation and Digital Signal Processing for Electromagnetic Field Modelling*, an invited paper, International Journal of Numerical Modelling — Electronic Networks, Devices and Fields, vol. 8, no. 3/4, May-August 1995, pp 169-185, John Wiley & Sons Inc..

[3] C. Eswarappa, P.P.M. So, J.L. Herring and W.J.R. Hoefer, *Progress and Critical Issues in Accurate Time Domain Modelling of Planar Wave Guiding and Radiating Structures with TLM*, Progress in Electromagnetic Research Proceedings, pp 732, July 1995, Seattle, Washington.

[4] P.P.M. So, C. Eswarappa and W.J.R. Hoefer, *Massively Parallel and Distributed Computing and Digital Signal Processing for TLM Electromagnetic Field Modeling*, a workshop paper, IEEE APS, June 1994, Seattle, Washington.

[5] P.P.M. So and W.J.R. Hoefer, *3D-TLM Techniques for Field-Based Optimization of Electromagnetic Structures*, IEEE APS, pp.1116-1119, June 1994, Seattle, Washington.

[6] C. Eswarappa, P.P.M. So and W.J.R. Hoefer, *Efficient Field-Based CAD of Microwave Circuits on Massively Parallel Processor Computer Using TLM and Prony's Methods*, IEEE MTT-S, pp.1531-1534, May 1994, San Diego, California.

[7] P.P.M. So and W.J.R. Hoefer, *Optimization of Microwave Structures using a Parallel TLM Module*, 10th Annual Review of Progress in Applied Computational Electromagnetics Digest, pp.546-553, March 1994, Monterey, California.

[8] P.P.M. So and W.J.R. Hoefer, *Distributed Computing for Transmission Line Matrix Method*, Second International Workshop on Discrete Time Domain Modelling of Electromagnetic Fields and Networks, October 1993, Berlin Germany.

[9] P.P.M. So, W.J.R. Hoefer, J.W. Bandler, R.M. Biernacki and S.H. Chen, *Hybrid Frequency/Time Domain Field Theory Based CAD of Microwave Circuits*, 23rd European Microwave Conference Digest, pp.218-219, September 1993, Madrid, Spain.

[10] P.P.M. So and W.J.R. Hoefer, *A New Look at the 3D Condensed Node TLM Scattering*, IEEE MTT-S, pp. 1443-1446, June 1993, Atlanta, Georgia.

[11] P.P.M. So, C. Eswarappa and W.J.R. Hoefer, *Transmission Line Matrix Method on Massively Parallel Processor Computers*, 9th Annual Review of Progress in Applied Computational Electromagnetics Digest, pp.467-474, March 1993, Monterey, California.

[12] W. Menzel, F. Alessandri, M. Mongiardo, R. Sorrentino, C. Eswarappa, P.P.M. So and W.J.R. Hoefer, *Analysis of a Millimeter-Wave Filter Using Transmission Lines Matrix and Mode Matching Methods and Comparison with the Measurements*, 9th Annual Review of Progress in Applied Computational Electromagnetics, pp.289-296, March 1993, Monterey, California.

[13] U. Müller, P.P.M. So and W.J.R. Hoefer, *The Compensation of Coarseness Error in 2D TLM Modeling of Microwave Structures*, IEEE MTT-S, vol. 1, pp.373-376, June 1992, Albuquerque, New Mexico.

[14] R. Sorrentino, P.P.M. So and W.J.R. Hoefer, *Numerical Microwave Synthesis by Inversion of the TLM Process*, 21st European Microwave Conference Digest, pp. 1273-1277, September 1991, Stuttgart, Germany.

[15] P.P.M. So and W.J.R. Hoefer, *3D-TLM Time Domain Electromagnetic Wave Simulator for Microwave Circuit Modelling*, IEEE MTT-S, vol. 2, pp. 631-634, June 1991, Boston, Massachusetts.

[16] P. Russer, P.P.M. So and W.J.R. Hoefer, *Modelling of Nonlinear Active Region in TLM*, IEEE Microwave and Guided Wave Letters, January 1991.

[17] W.J.R. Hoefer, P.P.M. So, D. Pompei, and A. Papiernik, *Numerical Analysis of Guiding and Radiating Microwave Structures based on Huygens' Principle*, Huygens' Symposium, November 1990, Scheveningen, The Netherlands.

[18] Eswarappa, P.P.M. So and W.J.R. Hoefer, *New Procedure for 2D and 3D Microwave Circuit Analysis with the TLM Method*, IEEE MTT-S, vol 2, pp. 661-664, May 1990, Dallas, Texas.

[19] P.P.M. So, Eswarappa and W.J.R. Hoefer, *A Two-dimensional Transmission Line Matrix Microwave Field Simulator Using New Concepts and Procedures*, IEEE MTT, vol. 37, no 12, pp. 1877-1884, December 1989.

[20] P.P.M. So and W.J.R. Hoefer, *A General Planar Circuit Simulator Based on Two-dimensional TLM Method*, IEEE MTT-S, vol. 1, pp. 343-346, June 1989, Long Beach, California.

[21] P.P.M. So and W.J.R. Hoefer, *CAD of E-Plane Circuits with Field-Theory Based Lookup Tables and Discontinuity Models*, IEEE MTT-S, vol 1, pp. 335-338, May 1988, New York, New York.

CHAPTER 1 Objectives and Contributions

1.1 Introduction

The design of complex high frequency electromagnetic structures is a very challenging task. The traditional design method is an iterative process based on transmission line and lumped element equivalent circuit models for some pre-defined electromagnetic components as well as educated trial-and-error experiments. Since the accuracy of these models depends on the frequency and circuit density, such an approach may not always yield the desired specifications with a minimum cost.

The microwave engineering software industry has provided some new products (see Table 1.1 for a list of the well known products) to tackle the above problems. These products are based on Finite Elements (FE), Method of Moment (MoM), Finite Difference Time Domain (FDTD) and the Transmission Line Matrix (TLM) method. All of the above software tools have their weaknesses and strengths. Since TLM is one of the more recent methods in the electromagnetic design area, many issues must still be resolved before TLM can be used routinely to design complex electromagnetic structures. The main objective of this thesis is to identify these issues and to provide innovative solutions to them, if possible, in a form that is readily transferrable to the research and educational communities as well as to the software industry for microwave and/or millimeter wave engineering.

Simulator	Numerical Method	
HFSS	Finite Elements	(FEM)
Maxwell Eminence	Finite Elements	(FEM)
EMAS/MicroWaveLab	Finite Elements	(FEM)
Microwave Explorer	Method of Moment	(MoM)
MAFIA	Finite Difference Time Domain	(FDTD)
XFDTD	Finite Difference Time Domain	(FDTD)
Stripes	Transmission Line Matrix	(TLM)

Table 1.1: Some well known electromagnetic wave simulation tools and the numerical methods that they are based upon.

1.2 The Difficulty of Using TLM to Design Electromagnetic Structures

To design complex electromagnetic structures with TLM is not trivial in spite of the simplicity of the fundamental TLM algorithm. Some of the problems are:

- TLM is a time and space discretization method; the algorithm has a per-time-step computational requirement which is proportional to the number of cells in the discretized space. Therefore, by simply applying TLM in a brute force manner one cannot tackle any realistic design problems.
- Traditional TLM programs provided by many research and educational institutions are very difficult to use. The users must have an in-depth knowledge of the method in order to use the programs effectively. Because all of these programs require the users to manually discretize the structures, it is quite difficult to analyze complex electromagnetic structures with these programs.
- TLM is a relatively recent method for computed-aided design (CAD) of microwave and millimeter wave circuits; the CAD software industry has been very slow in creating a well designed tool based on TLM*. The lack of such a tool makes it very difficult for engineers to use TLM in their design work.

* Stripes is the first commercially available CAD product based on TLM, but it does not have the advanced features mentioned in this thesis.

1.3 Research Objectives

The principal objective of this thesis is to provide solutions to the above problems. The goal is to develop new theoretical and computational features, and to implement them as object modules that can be linked with a simulation tool. These modules should be readily transferrable to researchers and educators as well as to industry practitioners for immediate use and further development. In order to develop and implement these modules, a simulation tool must be created concurrently because commercially available field simulators do not provide linkable binary or source code.

The simulation tool to be created should empower practicing microwave engineers to solve design problems that they cannot tackle with the presently available CAD software. The tool should have features that are not found in presently available commercial simulators. These features should be identifiable as being closely related to state-of-the-art laboratory equipment and should emulate a well equipped research laboratory. The following sub-objectives are set forth to guide the research towards this goal:

- Develop new special nodes and meshes for representing fine geometrical features through local modification. This would improve the efficiency, accuracy and versatility of the TLM method.
- Develop new algorithms based on parallel and distributed computing paradigms to improve the computational efficiency of TLM.
- Employ signal processing techniques to predict the future response from the known TLM response so that unnecessary computing can be eliminated.
- Develop automatic mesh generation and engineering parameter extraction procedures. These procedures should be easily integrated with external programs so that TLM can be used strategically in conjunction with traditional linear circuit simulators to perform analysis and optimization.

To reach these objectives, this thesis must straddle a number of disciplines — electromagnetic theory, TLM modelling, algorithms development, design and optimization, parameter extraction and signal processing in microwave and millimeter-wave engineering, as well as various issues in simulation tool design and implementation. These disciplines and their relationships to each other in the framework of this thesis are depicted in Figure 1.1.

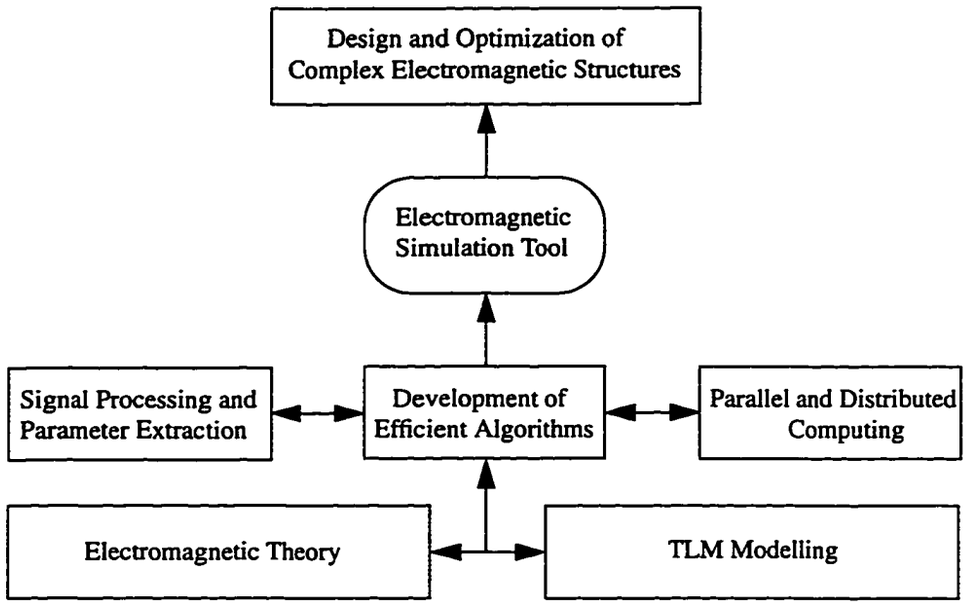


Figure 1.1: This thesis straddle a number of disciplines; their relationship is depicted in this diagram.

1.4 Original Contributions

The above research objectives have been met by virtue of the following original contributions to the TLM method, most of which have been published in leading journals and conference proceedings over the past seven years.

- Recursive Generation of Johns Matrix — [26] and [48].
- Modelling of Non-Linear Devices — [24], [28] and [48].
- Corner Node — [15].
- New 3D-TLM Impulse Numbering Scheme — [9].
- 3D-TLM Impulse Splitting Procedure — [9].
- Time Reversal TLM Process — [20] and [48].
- Parallel Computing Algorithms — [11], and [41].
- Distributed Computing Algorithms — [3], [6] and [41].
- Optimization Procedures — [8].
- Automatic Discretization of Elements.
- Adaptive Cell.
- Adaptive Mesh.
- Automatic Discretization of Elements using Adaptive Mesh.
- Prototype Simulators based on the TLM Method — [2], [22], [28], [29] and [48].

1.5 History of this Research

The research for this thesis started in January 1989. At that time, the author was a research engineer at the University of Ottawa, Ottawa, Ontario, Canada. Some original contributions such as the recursive generation of Johns Matrix [29] and corner node [15] were published before he started his Ph.D. studies. He also implemented a number of simulators based on 2D-TLM and 3D-TLM for DOS and the X Window system. One of these simulator [48] has been marketed by John Wiley and Sons Inc., and a 3D-TLM simulator [22] for the X Window system was distributed to various research and educational institutions as well as to high technology companies for evaluation. The author has thus gained invaluable insight into TLM modelling as well as experience in interfacing TLM with various existing microwave engineering CAD tools. He joined the University of Victoria, Victoria, British Columbia, Canada, in June 1992 and started his Ph.D. studies in the September of the same year. In order to verify the new ideas and procedures developed in this thesis, he has implemented a new multi-purpose electromagnetic field simulation tool based on the TLM method using the object-oriented design paradigm with the C++ programming language. Most of the simulation results shown in this thesis were obtained using this prototype simulator.

1.6 Conclusion

The research objectives and original contributions of this thesis are presented in this chapter. Most of these contributions have been published in leading journals and conference proceedings over the past seven years. However, there are some new contributions which are yet to be published in research journals. This thesis thus provides a first release of these theory and techniques. A prototype multi-purpose electromagnetic field simulation tool has also been created in order to verify the new theory and techniques developed in this thesis.

The above mentioned contributions are described in a coherence fashion in the remaining chapters of this thesis. The prototype simulator, which is created to verify the new theory and techniques developed in this thesis, is still at the experimental stage and no general release will be made until all the advanced features are completely implemented. However, the design framework which its implementation is based upon is described in the last chapter.

CHAPTER 2 Electromagnetic Field Modelling and Applications

2.1 Introduction

Electromagnetic field models are important if not essential for understanding the properties of electromagnetic fields. These models also form the foundation for electromagnetic field modelling with modern computers. This chapter begins with a discussions of two electromagnetic field models and the associated computer algorithms. Once the basic features of these models are given, the importance and applications of electromagnetic field modelling as well as their impact on microwave engineering CAD tools are discussed. Finally, the reasons for choosing TLM as the core method in this thesis are given.

2.2 Electromagnetic Field Models and Their Computer Algorithms

Electromagnetic field models assist one to understand the properties of electromagnetic fields; the number of models developed in the past to describe the propagation of light underlines their importance. The following are just some of these models [44]:

- Pythagoras, 6^{th} century B.C. — Light is a stream of particles emitted from a source of light.

- Demokritos, 460-370 B.C — All objects consist of a large number of particles: the atoms. Images are transmitted by atoms detaching themselves from the surface of objects and drifting through space.
- Aristotle, 384-322 B.C. — Light moves as a wave, like ripples on the water.
- Newton, 1662-1727 — Light is made up of tiny particles that fly through space.
- Huygens, 1629-1695 — Light is transmitted in the form of shock waves due to the superposition and succession of elementary spherical disturbance of the *Ether*; this is known as Huygens' Principle.
- Maxwell, 1831-1879 — Light is an electromagnetic wave governed by the interaction of electric and magnetic fields; this theory was experimentally confirmed by Hertz.
- Einstein / de Broglie, 20^{th} century — Light has both particle and wave properties. These aspects are complementary; one or the other dominates, depending on the phenomenon under study.

Some of the above models are known to be incorrect and/or incomplete descriptions of the propagation of light. However, Einstein / de Broglie have clearly seen the merits of both the corpuscular and wave models of light and, by extensions, of all electromagnetic waves. The next two sections give an overview of Huygens' principle, Maxwell's theory and computer algorithms based on them.

2.2.1 Huygens' Principle and TLM

TLM is a numerical method for modelling the behaviour of electromagnetic waves in the time domain. This method is based on Huygens' principle which is a localized recursive definition of electromagnetic wave propagation in the time domain. According to Huygens, a wavefront consists of a number of secondary radiators which give rise to spherical wavelets. The envelope of these wavelets forms a new wavefront which, in turn, gives rise to a new generation of spherical wavelets, and so on. This idea is depicted in two drawings from the Treatise of Light, [50], which are reproduced in Figure 2.1. The systematic application of this principle leads to an accurate description of wave propagation and scattering.

In order to implement Huygens' wave model on a digital computer, one must formulate it in discretized form; i.e. both space and time must be represented in terms of finite elementary units, Δl and Δt, which are related by the velocity of light, c, such that

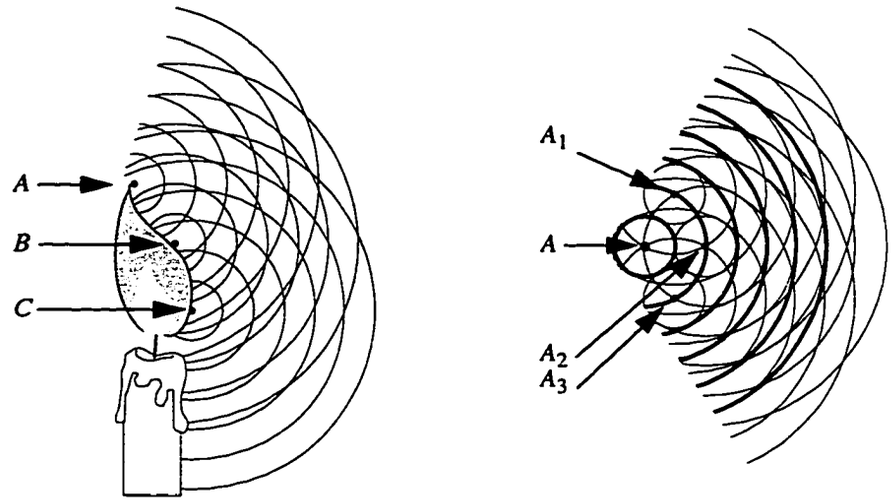


Figure 2.1: A drawing describing Huygens' principle, after the *Treatise of Light* [50].
(a) A candle light with three point sources.
(b) Huygens' principle applied to point A; A_1, A_2 and A_3 are three points on the secondary wavefront emitted from A.

$$\Delta l = c \cdot \Delta t \qquad \text{Eq 2.1}$$

Two-dimensional space is then modelled by a Cartesian matrix of points or nodes, separated by the mesh parameter Δl, see Figure 3.1 in *Chapter 3*. The unit time Δt is the time required for an electromagnetic signal to travel from one node to the next. Three-dimensional space can be modelled similarly. Detailed descriptions of the two-dimensional and three-dimensional TLM method are given in *Chapter 3* and *Chapter 4* of this thesis.

2.2.2 Maxwell's Equations and FDTD

Maxwell's equations in the time domain are:

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \qquad \nabla \times \vec{H} = \vec{J} + \frac{\partial \vec{D}}{\partial t}$$
$$\nabla \cdot \vec{D} = \rho \qquad \nabla \cdot \vec{B} = 0 \qquad \text{Eq 2.2}$$

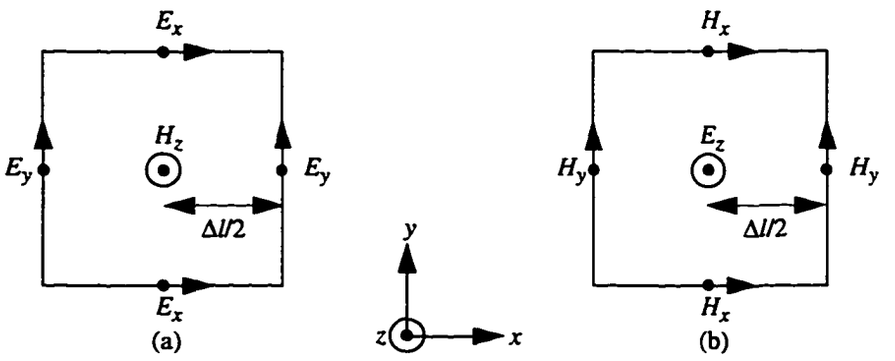


Figure 2.2: The fundamental building block of two-dimensional FDTD. Note that the electric and magnetic field are separated by $\Delta l/2$.
(a) Typical cell for TE wave.
(b) Typical cell for TM wave.

where $\vec{D} = \varepsilon\vec{E}$ and $\vec{B} = \mu\vec{H}$ for isotropic materials. These equations can be expressed using finite difference operators for the spatial and time derivatives. In two-dimensional space, the finite difference forms of the above two curl equations for $\vec{J} = 0$ are [43]:

$$\frac{E(k+1,n)-E(k,n)}{\Delta l} = -\mu\frac{H(k+0.5,n+0.5)-H(k+0.5,n-0.5)}{\Delta t}$$
$$-\frac{H(k+0.5,n+0.5)-H(k-0.5,n+0.5)}{\Delta l} = \varepsilon\frac{E(k,n+1)-E(k,n)}{\Delta t} \qquad \text{Eq 2.3}$$

The modelling method based on Eq 2.3 is known as the Finite Difference Time Domain (FDTD) method; Figure 2.2 depicts the fundamental building block of FDTD.

2.3 Importance and Applications of Electromagnetic Field Modelling

Lumped element models have been successfully used in many areas of circuit design; this approach, however, cannot handle the design of modern complex high speed microwave circuits. TLM and FDTD are good alternatives for analyzing electromagnetic structures with highly irregular and complex geometry; these are the areas where the traditional analytical methods fail to yield accurate results. Two major areas for electromagnetic field modelling are:

Computer Aided Design

- The key to economical success of a product is a short development time. Field theory based modelling methods can be used to create high quality engineering

CAD tools for designing products with complex electromagnetic structures in a short time. High quality CAD tools are also the only means to perform analysis and optimization of complex electromagnetic structures. They can also be used to validate lumped element models in conjunction with measurements.

Education and Field Visualization

- Field visualization tools can be built by combining TLM and/or FDTD with interactive computer graphics. These tools can produce high quality, realistic images and/or movies of electromagnetic waves, thus allowing students and researchers to observe electromagnetic processes which can only be imagined if such a tool is not available. Through interactive graphic representation, most electromagnetic processes can be comprehended in their full complexity by intuition. New ideas and associations can be stimulated. Engineers and scientists have always thought in pictures; with such a tool, they can create and observe these pictures on the computer screen.

2.4 Features of an Advanced Electromagnetic Field Simulation Tool

In order for FDTD or TLM to be used routinely by researchers and engineers, a well designed simulation tool based on these methods must be available. Traditional microwave engineering CAD tools use empirical closed-form expressions and parameter tables to represent the behaviour of complex microwave structures by assuming that interactions among circuit elements only take place at their input and output ports. No leakage effect or crosstalk is taken into account directly. Therefore, these tools can only be used to design circuits of low complexity and density. A new generation of microwave engineering CAD tools based on electromagnetic field modelling methods such as TLM would have the capability to account for the parasitic interactions among adjacent circuit elements. Some of the important features that such a new generation of microwave engineering CAD tool must have are:

User Friendly Interface

- Automatic circuit layout and mesh generation.
- Direct interface with measurement hardware for verification and testing.
- Signal processing capability.
- Integrated documentation facility.

Powerful Modelling Capabilities

- Impulsive, steady state and transient excitation waveforms.
- Nonlinear, thermal and time effects.
- Circuits of high complexity and density.
- Partitioning of structures.
- EMI/EMC and crosstalk effects.
- Packaging effects.

Versatile Simulation Features

- Multiple time and frequency domain methods.
- Analysis.
- Optimization.
- Synthesis.
- Engineering parameter extraction.
- Field visualization and animation.

Compatibility with Existing Microwave Engineering CAD Tools

- Support Touchstone and Supercompact *S*-parameter file syntax.
- Support Microsoft Windows Object Linking and Embedding (OLE).
- Support UNIX piping and other X Windows based communication features.

Obviously, this thesis cannot cover all of the above areas. In fact, many of them are outside the area of electromagnetic wave engineering. Therefore, only selected areas are covered in the remaining chapters.

2.5 A Practical Limitation of TLM and FDTD

The per-time-step computational requirements for TLM and FDTD* are proportional to the number of cells in the discretized space. For example, if the size of a three-dimensional problem is one wavelength in each spatial direction, and the discretization resolution Δl is $\lambda/10$ then 1000 cells must be stored in the computer memory. Doubling the discretization resolution of this problem to

* This is true in general for all electromagnetic field modelling methods that depend on the discretization of space and time.

$\Delta l = \lambda/20$ in each direction would require $2^3 \times 1000 = 8000$ cells to be stored. Since the speed of electromagnetic waves, c, is the same as before, according to Eq 2.1 the new elapsed time for the electromagnetic signal to travel from one node to the next is $\Delta t_{new} = \Delta t_{old}/2$. In the other words, the number of time-steps required for a wave-front to cross the same problem space is doubled*. Hence, the increase in execution time for this particular situation is $2^4 = 16$ times.

In many practical situations, especially when square cells are used, increasing the discretization resolution implies doubling† the resolution in all directions. This is because, the initial resolution is usually chosen so that all the elements in the structure can nicely aligned with the cell boundaries. In order to preserve the alignment condition, the resolution must be increased by an integer value.

Therefore, a straightforward application of TLM or FDTD to complex design problems would rapidly exceed the capacity of the most powerful machine. Hence, advanced techniques must be developed to avoid unnecessary fine discretization resolution and to increase the computation speed as well as to reduce the number of computation steps required for convergence of the results. These topics are covered in *Chapter 6* and *Chapter 7* of this thesis.

2.6 Conclusion

The fundamental principle of TLM and FDTD as well as their importance in electromagnetic wave engineering are presented. Neither TLM nor FDTD is best for all kinds of applications. These methods are powerful time domain techniques which have their advantages and disadvantages depending on the areas of applications; in fact these two methods are equivalent in the sense that both can be derived from Maxwell's equations, and in addition, TLM can be derived from FDTD and vice versa — [4], [16], [19], [23] and [33]. It is beyond the scope of this thesis to include both TLM and FDTD. Hence TLM is chosen for the work of this thesis. However, most of the theory and techniques presented in the following chapters can be applied to FDTD as well.

---

* The total amount of time is still the same; only the number of time steps is doubled because the new elapsed time for the electromagnetic signal to travel from one node to the next is half.

† Or tripling, quadrupling and so on.

CHAPTER 3 The 2D-TLM Method

3.1 Introduction

The 2D-TLM method is of considerable interest and practical importance because many wave problems can be formulated as two-dimensional problems. For instance, the propagation of TE_{n0} modes in homogenous parallel plate and rectangular waveguides, including scattering at discontinuties of constant dimensions in the direction of the E-field, involve only two spatial directions. Inductive strips, posts, irises, and filters made thereof, as well as T-junctions, bends and n-furcations in the H-plane are some of the numerous examples. Furthermore, the cutoff frequencies of all modes in inhomogeneously filled cylindrical waveguides of arbitrary cross-sectional geometry can be found using two-dimensional analysis.

This chapter gives an overview of the theoretical foundations of the 2D-TLM method, and shows how it can be used to model the propagation, reflection, refraction and attenuation of fields in two-dimensional space. Because of the simplicity of the 2D-TLM networks, it is not difficult to understand the 2D-TLM method, its algorithm as well as it limitations. The two-dimensional TLM method also serves as the basis for understanding the more complicated three-dimensional TLM method described in the following chapter.

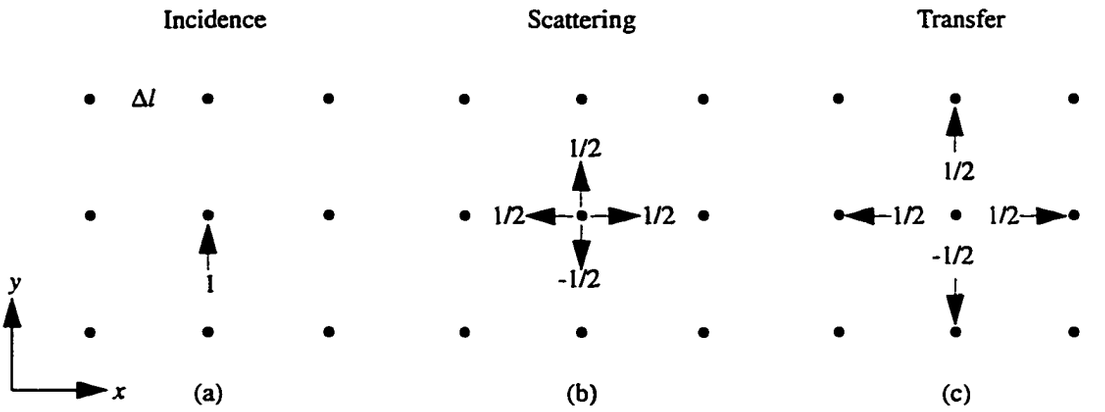


Figure 3.1: Huygens' wave model in the discretized two-dimensional space.
(a) Incidence of a short voltage impulse at a space point (scattering center).
(b) Scattering of the impulse.
(c) Transfer of the scattered impulses to neighbouring nodes.

3.2 Discretization of Space and Time

In order to implement Huygens' wave model on a digital computer, one must formulate it in discretized form; i.e. both space and time must be represented in terms of finite elementary units, Δl and Δt, which are related by the velocity of light, c, such that:

$$\Delta l = c \cdot \Delta t \qquad \text{Eq 3.1}$$

Two-dimensional space is then modelled by a Cartesian matrix of points or nodes, separated by a unit distance Δl, Figure 3.1a. The unit time Δt is the time required for an electromagnetic signal to travel from one node to the next.

Assume that a very short electromagnetic impulse is incident upon one of these nodes from the negative y-direction. The field amplitude of the impulse is unity. According to Huygens' principle, its energy is scattered isotropically in all four directions, each radiated impulse carrying one fourth of the incident energy. The scattered field quantities must then be $1/2$ in magnitude. Furthermore, the reflection coefficient seen by the incident impulse must be $-1/2$ to ensure field continuity at the node, Figure 3.1b. This event can be described by the following simple matrix equation [48]:

$${}_{k+1}V^r = S \cdot {}_kV^i \qquad \text{Eq 3.2}$$

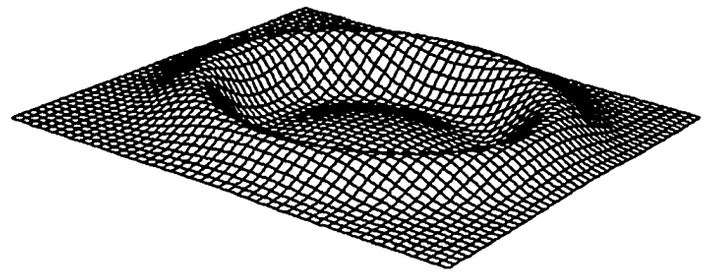


Figure 3.2: Propagation of wave simulated by executing equations 3.2 and 3.3 repeatedly.

where $V = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}$, $S = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}$, V_1, V_2, V_3 and V_4 represent the voltage impulses incident from the bottom, left, top and right* of a node. The superscripts i and r denote incident and reflected impulses. The subscripts k and $k+1$ represent the time in Δt. Furthermore, any impulse emerging from a node becomes automatically an incident impulse on the neighbouring node (Figure 3.1b and c). This event can be best described by the following set of equations:

$$
\begin{aligned}
{}_{k+1}V_1^{\,i}(x,y) &= {}_{k+1}V_3^{\,r}(x,y-1) \\
{}_{k+1}V_2^{\,i}(x,y) &= {}_{k+1}V_4^{\,r}(x-1,y) \\
{}_{k+1}V_3^{\,i}(x,y) &= {}_{k+1}V_1^{\,r}(x,y+1) \\
{}_{k+1}V_4^{\,i}(x,y) &= {}_{k+1}V_2^{\,r}(x+1,y)
\end{aligned}
\qquad \text{Eq 3.3}
$$

The coordinates in Eq 3.3 are normalized to the space discretization unit, Δl.

Equations 3.2 and 3.3 form the basic algorithm of the 2D-TLM method. Thus, if the magnitudes, positions and directions of all voltage impulses are known at time $k\Delta t$, then the corresponding values of the voltage impulses at time $(k+1)\,\Delta t$ can be obtained by operating equations 3.2 and 3.3 on each node in the mesh. Figure 3.2 depicts a snapshot of a wave propagating away from the center simulated by executing the equations repeatedly.

---

* Bottom, left, top and right represent the negative y, negative x, positive y, and positive x-direction.

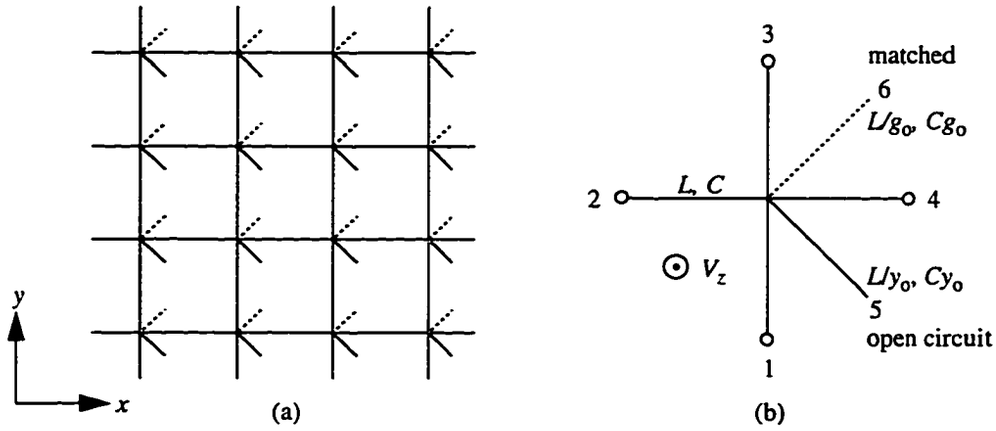


Figure 3.3: (a) A bird's eye view of a 2D shunt-connected transmission line network.
(b) A building block of the mesh in (a). For synchronization purposes, transmission lines 1 to 5 are $\Delta l/2$ long. L and C are the inductance and capacitance per unit length for lines 1 to 4. L/y_o and Cy_o are the inductance and capacitance per unit length for line 5, which is called the *permittivity stub*. L/g_o and Cg_o are the inductance and capacitance per unit length for line 6, which is called the *loss stub*.

3.3 Shunt-Connected 2D-TLM Network

The previous section introduced a discretized version of Huygens' principle in the two-dimensional space. In fact, such a system can be modelled by an equivalent network of transmission lines, or a transmission line matrix, [38], [39] and [40]. This section gives an overview of the formulation of the shunt-connected transmission line matrix method; an in-depth discussion of it can be found in [48] and [51].

Figure 3.3a depicts a bird's eye view of a 2D shunt-connected transmission line network, or 2D-TLM shunt mesh. The building block of the mesh is called the 2D-TLM shunt node, Figure 3.3b, which consists of the following elements:

- Four transmission lines of length $\Delta l/2$; their inductance and capacitance per unit length are L and C, respectively.
- An open-circuited transmission line of length $\Delta l/2$, its inductance and capacitance per unit length are L/y_o and Cy_o, respectively; y_o is called the normalized characteristic admittance. This is called the *permittivity stub*.
- A matched transmission line, its inductance and capacitance per unit length are L/g_o and Cg_o, respectively. This is called the *loss stub* and is used to model a lumped normalized shunt conductance g_o.

3.3.1 2D-TLM Shunt Node Scattering Matrix

A voltage impulse incident upon the 2D-TLM shunt node is scattered into all six lines, but only five of these impulses will be returned to the node. The sixth is absorbed by the loss stub and disappears. The reflection coefficient seen by an impulse incident on one of the four mesh line branches is:

$$S_{ii} = \frac{2-y}{y}, \qquad i = 1...4 \quad \text{and} \quad y = 4+y_o+g_o. \qquad \text{Eq 3.4}$$

The transmission coefficient for each outgoing line is

$$S_{ij} = 1+S_{jj} = \frac{2}{y}, \qquad i \neq j. \qquad \text{Eq 3.5}$$

Impulses entering the permittivity stubs are reflected at the open end and become incident impulse on the node:

$${}_{k+1}V_5^{\,i}(x,y) = {}_{k+1}V_5^{\,r}(x,y), \qquad \text{Eq 3.6}$$

where they are subject to the following reflection and transmission coefficients:

$$S_{55} = \frac{2y_o-y}{y}, \qquad S_{i5} = 1+S_{55} = \frac{2y_o}{y} \quad \text{for} \quad i \neq 5 \qquad \text{Eq 3.7}$$

The impulse scattering equation for the 2D-TLM shunt node thus becomes:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix}_{k+1}^{r} = \frac{1}{y}\begin{bmatrix} 2-y & 2 & 2 & 2 & 2y_o \\ 2 & 2-y & 2 & 2 & 2y_o \\ 2 & 2 & 2-y & 2 & 2y_o \\ 2 & 2 & 2 & 2-y & 2y_o \\ 2 & 2 & 2 & 2 & 2y_o-y \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix}_{k}^{i}, \qquad \text{Eq 3.8}$$

where $y = 4+y_o+g_o$; and the subscripts correspond to the impulse numbering scheme depicted in Figure 3.3. Note that, Eq 3.8 has a 5×5 scattering matrix and the corresponding scattering algorithm does not contains V_6. This is correct because the loss stub is terminated by a matched load and it absorbs all impulses scattered into the stub.

3.3.2 2D-TLM Shunt Mesh Wave Properties

As long as the mesh parameters Δl is small compared with the wavelength of interest, the voltage and current change in the x and y direction of the 2D-TLM shunt mesh can be found by comparing the differential equations that govern the propagation of voltage and currents in the mesh to the Maxwell's equations in the two-dimensional space.

The differential equations that govern the propagation of voltage and currents in the mesh are:

$$\begin{aligned} \frac{\partial V_z}{\partial x} &= -L\frac{\partial I_x}{\partial t} \\ \frac{\partial V_z}{\partial y} &= -L\frac{\partial I_y}{\partial t} \\ \frac{\partial I_x}{\partial x} + \frac{\partial I_y}{\partial y} &= -2C\left(1+\frac{y_o}{4}\right)\frac{\partial V_z}{\partial t} - \frac{g_o\sqrt{C/L}}{\Delta l}V_z \end{aligned} \qquad \text{Eq 3.9}$$

The corresponding Maxwell's equations in a lossy medium obtained by setting $\partial/\partial z = 0$ and $E_x = E_y = H_z = 0$ are:

$$\begin{aligned} \frac{\partial E_z}{\partial x} &= \mu\frac{\partial H_y}{\partial t} \\ \frac{\partial E_z}{\partial y} &= -\mu\frac{\partial H_x}{\partial t} \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} &= \varepsilon\frac{\partial E_z}{\partial t} + \sigma E_z \end{aligned} \qquad \text{Eq 3.10}$$

The following equivalences can be established by comparing Eq 3.9 and Eq 3.10:

$$\begin{aligned} E_z &\equiv V_z & H_x &\equiv I_y & H_y &\equiv -I_x \\ \mu &\equiv L & \varepsilon &\equiv 2C\left(1+\frac{y_o}{4}\right) & \sigma &\equiv \frac{g_o\sqrt{C/L}}{\Delta l} \end{aligned} \qquad \text{Eq 3.11}$$

3.4 Series-Connected 2D-TLM Network

Another way to model Huygens' principle in the two-dimensional space is via the series-connected transmission line matrix. This section gives an overview of the formulation of this method; an in-depth discussion of it can be found in [48] and [51].

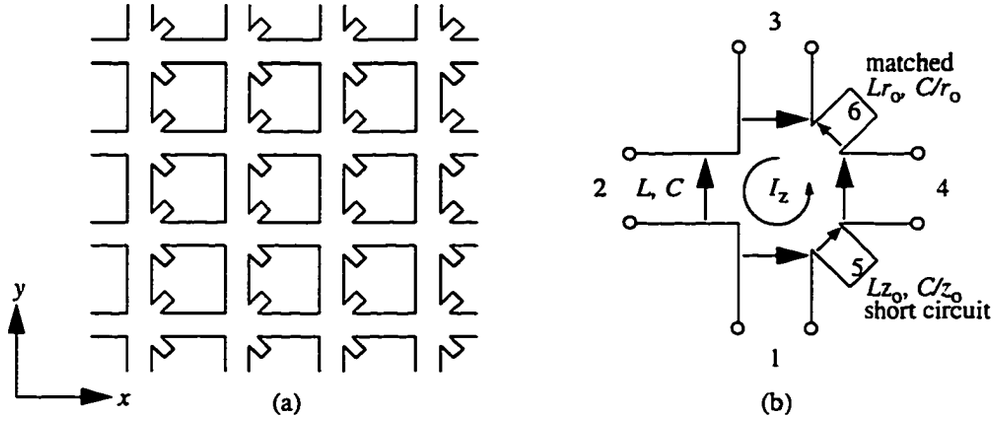


Figure 3.4: (a) A bird's eye view of a 2D series-connected transmission line network.
(b) A building block of the mesh in (a). For synchronization purposes, transmission lines 1 to 5 are $\Delta l/2$ long. L and C are the inductance and capacitance per unit length for lines 1 to 4. Lz_o and C/z_o are the inductance and capacitance per unit length for line 5, which is called the *permeability stub*. Lr_o and C/r_o are the inductance and capacitance per unit length for line 6, which is called the *loss stub*.

Figure 3.4a depicts a bird's eye view of a 2D series-connected transmission line network, or 2D-TLM series mesh. The building block of the mesh is called the 2D-TLM series node, Figure 3.4b, which consists of the following elements:

- Four transmission lines of length $\Delta l/2$; their inductance and capacitance per unit length are L and C, respectively.
- A short-circuited transmission line of length $\Delta l/2$, its inductance and capacitance per unit length are Lz_o and C/z_o, respectively; z_o is called the normalized characteristic impedance. This is called the *permeability stub.*
- A matched transmission line, its inductance and capacitance per unit length are Lr_o and C/r_o, respectively. This is called the *loss stub* and is used to model a lumped normalized series magnetic conductance r_o. The inclusion of such a parameter makes the series node a dual of the shunt node.

3.4.1 2D-TLM Series Node Scattering Matrix

Following a similar procedure as for the shunt node in *Section 3.3.1*, and taking into account the signs of voltages and current defined in Figure 3.4, the impulse scattering equation for the 2D-TLM series node thus becomes:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix}_{k+1}^{r} = \frac{1}{z} \begin{bmatrix} z-2 & 2 & 2 & -2 & -2 \\ 2 & z-2 & -2 & 2 & 2 \\ 2 & -2 & z-2 & 2 & 2 \\ -2 & 2 & 2 & z-2 & -2 \\ -2z_o & 2z_o & 2z_o & -2z_o & z-2z_o \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix}_{k}^{i}, \qquad \text{Eq 3.12}$$

where $z = 4 + z_o + r_o$; and the subscripts correspond to the impulse numbering scheme depicted in Figure 3.4. Note that, Eq 3.12 has a 5×5 scattering matrix and the corresponding scattering algorithm does not contain V_6. This is correct because the loss stub is terminated by a matched load and absorbs all impulses scattered into the stub.

3.4.2 2D-TLM Series Mesh Wave Properties

Following a similar procedure as for the shunt mesh in *Section 3.3.2*, the equivalence between the field and the network quantities in the series mesh are:

$$H_z \equiv I_z \qquad E_x \equiv -V_x \qquad E_y \equiv V_y$$

$$\varepsilon \equiv C \qquad \mu \equiv 2L\left(1 + \frac{z_o}{4}\right) \qquad \tan\delta = \frac{\mu''_r}{\mu'_r} = \frac{r_o c}{2\omega\Delta l\,(1 + z_o/4)} \qquad \text{Eq 3.13}$$

3.5 Definition of Time Step

As mentioned in *Discretization of Space and Time* (on page 14), both space and time must be represented in terms of finite elementary units, Δl and Δt, which are related by the velocity of light, c, such that

$$\Delta l = c \cdot \Delta t \qquad \text{Eq 3.14}$$

However, the wave velocity in the 2D-TLM mesh that model free space is:

$$v_{wave} = \frac{1}{\sqrt{\varepsilon_o \mu_o}} = \frac{1}{\sqrt{2LC}} = \frac{c}{\sqrt{2}} \qquad \text{Eq 3.15}$$

That means, the 2D-TLM mesh is a slow wave structure and the relationship between the wave impedance and the link line impedance is:

$$Z_{wave} = \frac{Z_{lo}}{\sqrt{2}} \qquad \text{Eq 3.16}$$

where $Z_{lo} = \sqrt{L/C}$. Therefore, for a given wavelength, a wave on the mesh has a lower frequency than a wave in the free space, or:

$$f_{wave} = \frac{f}{\sqrt{2}} \quad \text{or} \quad \omega_{wave} = \frac{\omega}{\sqrt{2}} \qquad \text{Eq 3.17}$$

Hence, in order to model the free space with the 2D-TLM mesh, one must scale frequencies and impedances according to Equations 3.16 and 3.17. In time harmonic applications, one may chose to scale the time step:

$$\Delta t_{wave} = \frac{\Delta t}{\sqrt{2}} = \frac{\Delta l}{c\sqrt{2}} \qquad \text{Eq 3.18}$$

so that, $(n\Delta t_{wave})\,\omega$ can be used in place of $(n\Delta t)\,\omega_{wave}$.

This is the definition of the elementary time unit for the wave, Δt_{wave}, in the 2D-TLM mesh that will be used throughout this thesis to compute Fourier transform as well as to generate discretized time domain functions for the 2D-TLM method.

3.6 Modelling of Boundaries

The shape and size of real structures such as waveguides, circuit components and scatters must be defined by boundaries. Furthermore, the size of the 2D-TLM computation domain is limited by the amount of available computer memory*; therefore, an absorbing boundary is required to model open space. These boundaries can be modelled by introducing appropriate impulse reflection coefficients, $\Gamma_{impulse}$, in the 2D-TLM mesh. These reflection coefficients must be:

- Real numbers because the impulse are real numbers.
- Implemented at node or halfway between two nodes because the movement of the impulses in the mesh must be synchronized.
- Chosen such that the interaction of the electromagnetic wave, represented by the superposition of impulses in the mesh, with the boundary is properly modelled.

* This includes the available amount of virtual memory which is supplied to the running process by means of swap space on the hard disk. However, the performance of the simulation tool would depend heavily on the efficiency of the swapping mechanism provided by the computer.

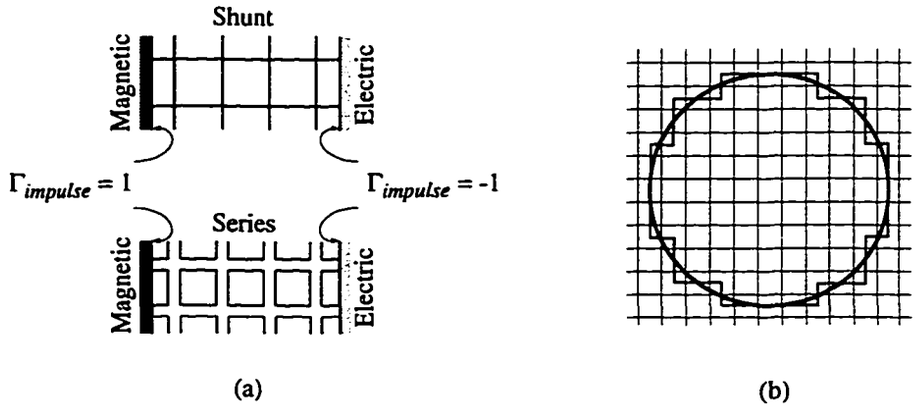


Figure 3.5: Representation of electric and magnetic boundaries in the 2D-TLM mesh.
(a) Electric and magnetic boundaries in the 2D-TLM shunt and series mesh.
(b) Approximation of a circular boundary by piecewise straight sub-boundaries.

The second condition implies that curved boundaries must be modelled by piecewise straight sub-boundaries. The following sub-sections describe the modelling of perfect electric and magnetic boundaries as well as simple absorbing boundaries. A thorough discussion of the various types of boundaries for the 2D-TLM method is given in [48].

3.6.1 Perfect Electric and Magnetic Boundaries

Electric and magnetic boundaries are the most common boundaries for electromagnetics problems. They can be used to define electromagnetic structures as well as to define the planes of symmetry in many structures so that the size of the computation domain can be reduced drastically. However, by solving the field problem in a single subregion, only modes with the corresponding symmetry will be found.

Figure 3.5a shows how electric and magnetic boundaries are modelled. These boundaries are perfectly reflecting walls of either zero or infinite impedance:

Electric Boundary	$Z_{impulse} = 0$	$\Gamma_{impulse} = -1$	
Magnetic Boundary	$Z_{impulse} = \infty$	$\Gamma_{impulse} = +1$	Eq 3.19

The reflection operations for electric and magnetic boundaries shown in Figure 3.5a can be achieved by executing the following equations:

$$ {}_kV_4^i(x,y) = -{}_kV_4^r(x,y) \qquad {}_kV_2^i(x,y) = {}_kV_2^r(x,y)\,, \qquad \text{Eq 3.20} $$

Since the impulse reflection coefficients can only be placed halfway between two nodes, curved boundaries must be modelled by piecewise straight sub-boundaries. Figure 3.5b shows a circular boundary modelled by piecewise straight sub-boundaries.

3.6.2 Simple Absorbing Boundary

An absorbing boundary can be simulated by a wall of impedance equal to the intrinsic impedance of the mesh. Eq 3.16 relates the wave impedance with the impulse impedance. In the stub-loaded mesh, their relationship is:

$$\text{Shunt} \qquad Z_{wave} = \frac{Z_{lo}}{\sqrt{2\,(1+y_0/4)}} = \frac{Z_{lo}}{\sqrt{2\varepsilon_r}}$$

$$\text{Series} \qquad Z_{wave} = \frac{Z_{lo}}{\sqrt{2\,(1+z_0/4)}} = \frac{Z_{lo}}{\sqrt{2\mu_r}} \qquad \text{Eq 3.21}$$

The impulse reflection coefficient, $\Gamma_{impulse}$, is $(Z_{wave}-Z_{lo})/(Z_{wave}+Z_{lo})$; that means:

$$\text{Shunt} \qquad \Gamma_{impulse} = \frac{1-\sqrt{2\varepsilon_r}}{1+\sqrt{2\varepsilon_r}}$$

$$\text{Series} \qquad \Gamma_{impulse} = \frac{1-\sqrt{2\mu_r}}{1+\sqrt{2\mu_r}} \qquad \text{Eq 3.22}$$

For free space, $\varepsilon_r = \mu_r = 1$:

$$\Gamma_{impulse} = \frac{1-\sqrt{2}}{1+\sqrt{2}} = -0.17157 \qquad \text{Eq 3.23}$$

Equations 3.22 and 3.23 are accurate only for $\Delta l/\lambda \ll 1$ because of the dispersive behaviour of the 2D-TLM mesh; see [48] and [51] for a general discussion of this behaviour. Furthermore, these two equations apply only in the case of normal incidence of a TEM wave upon the boundary. In general, the impulse reflection coefficient for a boundary is [48]:

$$\text{Shunt} \qquad \Gamma_{impulse}(\theta) = \frac{1-\sqrt{2\varepsilon_r}\cos\theta}{1+\sqrt{2\varepsilon_r}\cos\theta}$$

$$\text{Series} \qquad \Gamma_{impulse}(\theta) = \frac{1-\sqrt{2\mu_r}\cos\theta}{1+\sqrt{2\mu_r}\cos\theta} \qquad \text{Eq 3.24}$$

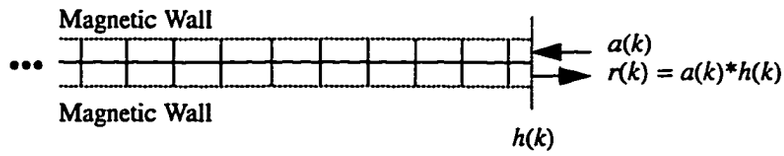


Figure 3.6: $h(k)$ is the impulse response of the TLM structure; therefore, $r(k) = a(k) * h(k)$.

where θ is the angle of incidence. A thorough discussion of the absorbing boundary for 2D-TLM is given in [48].

3.6.3 Advanced Absorbing Boundary

A more elegant way to implement an absorbing boundary is to make use of the intrinsic property of the TLM model. Since TLM describes field propagation by the incident and reflected voltage impulses on the link lines rather than the total voltages on the nodes, it is appropriate to formulate an absorbing boundary algorithm via the discrete Green's function concept — Let $h(k)$ be the one-dimensional Green's function of a structure, then the response, $r(k)$, of the structure due to an arbitrary excitation, $a(k)$, is:

$$r(k) = a(k) * h(k) = \sum_{n=0}^{k} a(n) \cdot h(k-n) \qquad \text{Eq 3.25}$$

If, $a(k)$ and $r(k)$ are used to represent the incident and reflected voltage impulses in the TLM world, and $h(k)$ is the impulse response of a semi-infinite TLM structure as shown in Figure 3.6, then Eq 3.25 is an absorbing boundary formulation for the TLM structure. This formulation can be generalized to two- and three-dimensional cases, [48]. This type of discrete impulse response is called the Johns Matrix in honor of the late P.B Johns who pioneered the TLM method and time domain diakoptics.

The Johns Matrix for the absorbing boundary can be generated recursively by using a single slice of TLM mesh terminated with the Johns Matrix which is yet to be generated, [26]. This is possible because:

- There is a $1\Delta t$ delay for the impulse to travel from the input plane to the Johns Matrix termination.
- The impulse response of a TLM mesh terminated with a Johns Matrix is itself the same Johns Matrix.

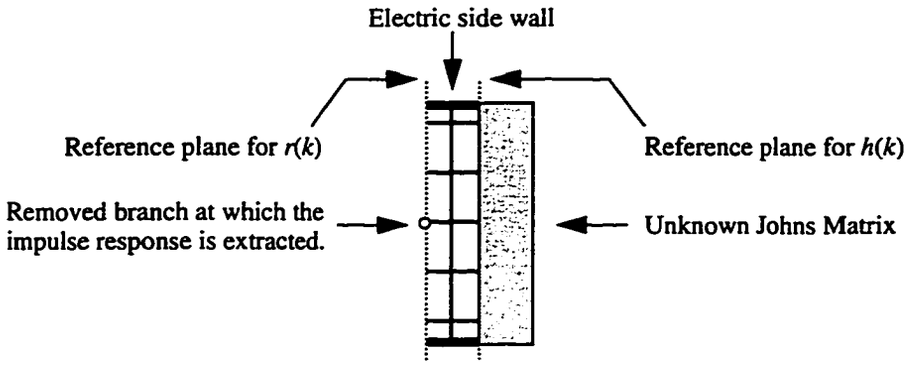


Figure 3.7: An undefined Johns Matrix is used to terminate a waveguide section. The Johns Matrix which would model an infinitely long waveguide section, is then built during a TLM iteration process by simply transferring each impulse from the removed branch to the load side to be convolved with the incident impulses after a delay of $1\Delta t$. For this example, impulses are injected into all removed branches with a half-sine weighted amplitude with the maximum at the center. Convolution on the Johns Matrix side is also done at the center link line only, the values of the other reflected impulses are computed using a half-sine spatial interpolation scheme.

This self-generating recursive algorithm can be explained easily by using a familiar waveguide section terminated by a Johns Matrix which is yet to be generated, Figure 3.7. An impulsive excitation, i.e. $a(k) = \delta(k)$, is launched into the removed branches. The Johns Matrix which would model an infinitely long waveguide section is then built during a TLM iteration process by simply transferring each impulse coming out of the removed branch to the load side for convolving with the incident impulses after a delay of $1\Delta t$. This is accomplished by assigning the impulse response $r(k)$ to the Johns Matrix $h(k)$. The evolution of the Johns Matrix generated by this recursive algorithm is shown in Figure 3.8.

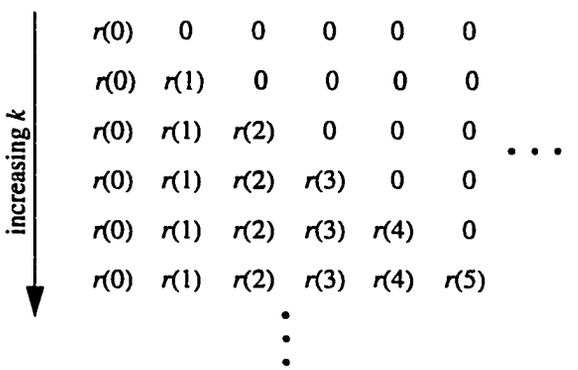


Figure 3.8: The evolution process of the Johns Matrix generated by the recursive algorithm described in Figure 3.7. Note that, $r(k)$ is the impulse response at $t = k\Delta t$, which is equal to $h(k)$.

3.7 Computation of Fields

Since field quantities in the two-dimensional space are directly proportional to the node voltages and currents in the 2D-TLM mesh, they can be obtained by superposition of voltages and currents in the mesh.

For the 2D-TLM shunt mesh; by using the voltage definition in Figure 3.3 and equivalence in Eq 3.11, the field quantities are:

$$
\begin{aligned}
{}_kE_z \equiv {}_kV_z &= \frac{2}{y}\left(\sum_{m=1}^{4} {}_kV_m^{\,i} + y_o \cdot {}_kV_5^{\,i}\right) \\
{}_kH_x \equiv {}_kI_y &= ({}_kV_1^{\,i} - {}_kV_3^{\,i})/Z_{lo} \\
{}_kH_y \equiv -{}_kI_x &= ({}_kV_4^{\,i} - {}_kV_2^{\,i})/Z_{lo}
\end{aligned}
\qquad \text{Eq 3.26}
$$

where $y = 4 + y_o + g_o$ and $Z_{lo} = \sqrt{L/C} = \sqrt{\mu_o/\varepsilon_o} = \eta_o$ is the characteristic impedance of the mesh lines.

Similarly, for the 2D-TLM series mesh; according to the voltage definition in Figure 3.4 and equivalence in Eq 3.13, the field quantities are:

$$
\begin{aligned}
{}_kH_z \equiv {}_kI_z &= \frac{2}{z}({}_kE_y + {}_kE_x + z_o \cdot {}_kV_5^{\,i})/Z_{lo} \\
{}_kE_x \equiv -{}_kV_x &= {}_kV_3^{\,i} - {}_kV_1^{\,i} \\
{}_kE_y \equiv {}_kV_y &= {}_kV_2^{\,i} - {}_kV_4^{\,i}
\end{aligned}
\qquad \text{Eq 3.27}
$$

where $z = 4 + z_o + r_o$ and $Z_{lo} = \sqrt{L/C} = \sqrt{\mu_o/\varepsilon_o} = \eta_o$ is the characteristic impedance of the mesh lines.

3.8 Excitation and Response

The TLM method allows maximum flexibility in the excitation of a structure as well as in the extraction of its response to that excitation because both the time behaviour and spatial distribution of the field quantities in the mesh can be controlled. The following sub-sections give a general discussion of the various excitation mechanisms as well as how to sample and process the response to yield useful engineering parameters.

3.8.1 Impulsive Excitation and Response

In many applications, the response of a structure for a wide frequency band is needed; such a frequency response can be obtained by an impulsive excitation. Once an impulsive signal is injected into the mesh at the input region, the injected energy spreads out from one node to the next along the link lines and are scattered at each node. Each iteration in the computer corresponds to the time interval Δt_{wave} in the real world (see *Section 3.5* for the definition of Δt_{wave}).

The impulse response at a particular output node in the mesh is simply obtained by observing the stream of impulses as they pass through that point. It can be written as:

$$f(t_{wave}) = \sum_{k=0}^{K} {}_kA \cdot \delta(t_{wave} - k\Delta t_{wave}) \qquad \text{Eq 3.28}$$

where K is the number of iterations that has been performed. The transfer function for the above response is:

$$F(\omega) = \sum_{k=0}^{K} {}_kA \cdot e^{-j\omega k\Delta t_{wave}} = F_{re}(\omega) - jF_{im}(\omega)$$

$$F_{re}(\omega) = \sum_{k=0}^{K} {}_kA \cdot \cos(\omega k\Delta t_{wave}) \qquad \text{Eq 3.29}$$

$$F_{im}(\omega) = \sum_{k=0}^{K} {}_kA \cdot \sin(\omega k\Delta t_{wave})$$

As the 2D-TLM mesh gives accurate results for $\Delta l/\lambda \ll 1$, Eq 3.29 gives accurate result for $\omega\Delta t_{wave} \ll 2\pi$. Eq 3.29 has an additional constraint related to the Nyquist rate which requires $\Delta t_{wave} \leq \pi/\Omega$, where Ω is the maximum allowable value for ω in the transfer function.

Note that, like in a real field measurement, the position of input and output points as well as the nature of the field component will affect the magnitudes of the spectral lines. By carefully selecting the input and output position, the response for certain modes can be either suppressed or enhanced. Figure 3.9 depicts such an effect for the response of a circular waveguide.

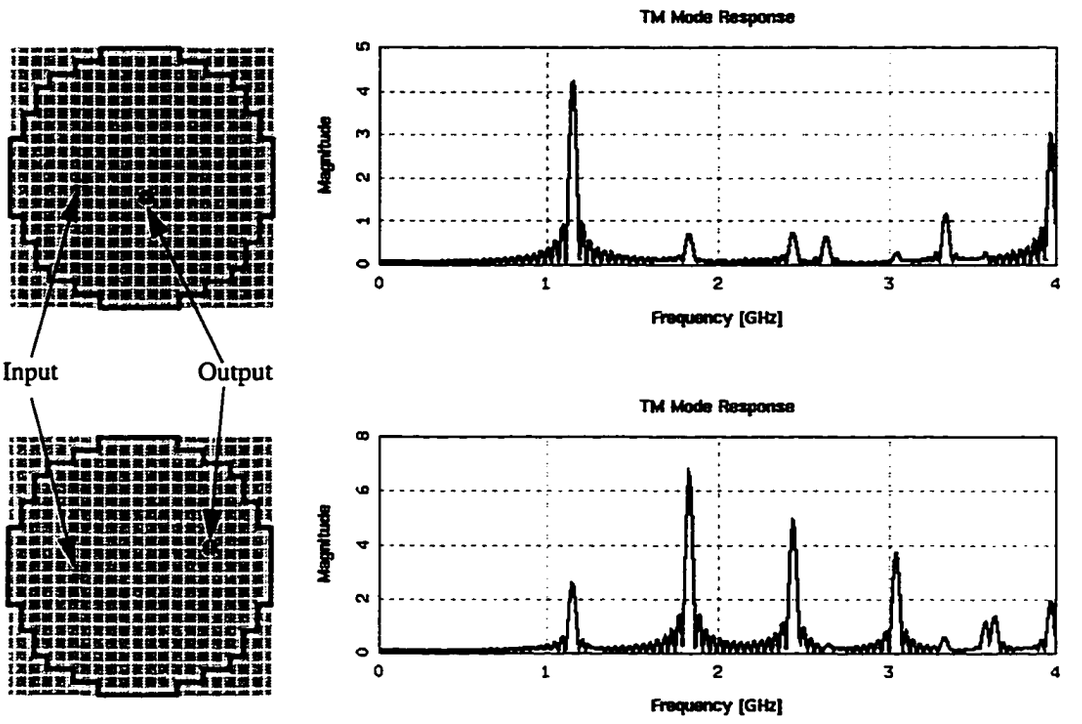


Figure 3.9: The effect of sampling at different positions. The input positions in both cases are the same, but the responses are different because the output positions are different. This feature can be used to enhance or suppress the response of certain modes in the structure.

3.8.2 Continuous Excitation and Response

By superposition of single impulse excitations in space and time, it is possible to simulate any source configuration and any waveform provided that the shortest wavelength of interest is much longer than the mesh discretization parameter, Δl. A wave with a particular profile can be generated by injecting impulses with a corresponding spatial amplitude distribution.

To launch a particular waveform, a stream of impulse functions with the exciting waveform as an envelope is injected into the mesh. These impulses are added at each iteration to the impulses already present in the mesh, thus continuously injecting energy into the system. The response of the mesh to the digitized waveform can be extracted directly at all nodes without further processing. This capability of simulating arbitrary waveforms is extremely useful in analyzing the propagation of impulse waves and transients, as well as in modelling of nonlinear and time dependent phenomena.

3.8.3 Computation of Scattering Parameters

Transmission line and waveguide components with a well defined mode of propagation are usually characterized in the frequency domain by their scattering parameters or *S*-parameters. In the laboratory, these complex parameters are determined with a network analyzer by a number of swept measurements; all ports of the component are terminated in a wideband matched load, and the complex ratios of reflected and transmitted signals are determined over the desired frequency range.

The TLM method allows us to simulate the swept measurement with a single impulsive analysis in the time domain. The structure under study, including its input and output ports, is represented by a TLM mesh. As in a typical *S*-parameters measurement, the reference plane must be chosen far enough from any discontinuity to ensure that higher order modes have sufficiently decayed. All output ports are than terminated with wideband absorbing boundaries. The input port is excited by a matched impulsive source having the transverse field distribution of the operating mode. The impulse response in the input and output reference planes is obtained by Fourier transforming the sampled signal in the center of the ports.

In order to compute the *S*-parameters, one must obtain the incident field. Because the field at the input port is a sum of the incident and the reflected field, a separate reference structure which models only the input section is needed to obtain the incident field. This idea is illustrated in Figure 3.10. In the most general case, the reference impedances of the various ports can be different from each other. It is therefore necessary to compute these impedances and to take them into account when extracting S_{21}. Once these values are obtained, the *S*-parameters can be computed using the following formulae:

$$S_{11} = \frac{V_1 - V_{ref}}{V_{ref}} \qquad S_{21} = \frac{V_2}{V_{ref}} \cdot \sqrt{\frac{Z_1}{Z_2}} \qquad \text{Eq 3.30}$$

Where V_{ref}, V_1 and V_2, are the impulse responses at the reference, input and output ports; Z_1 and Z_2 are the impedance in the input and output ports, respectively. Figure 3.10 depicts a low-pass filter and its S_{11} and S_{21} characteristics computed by this method. Note that, for this particular example, the input and output impedance are the same.

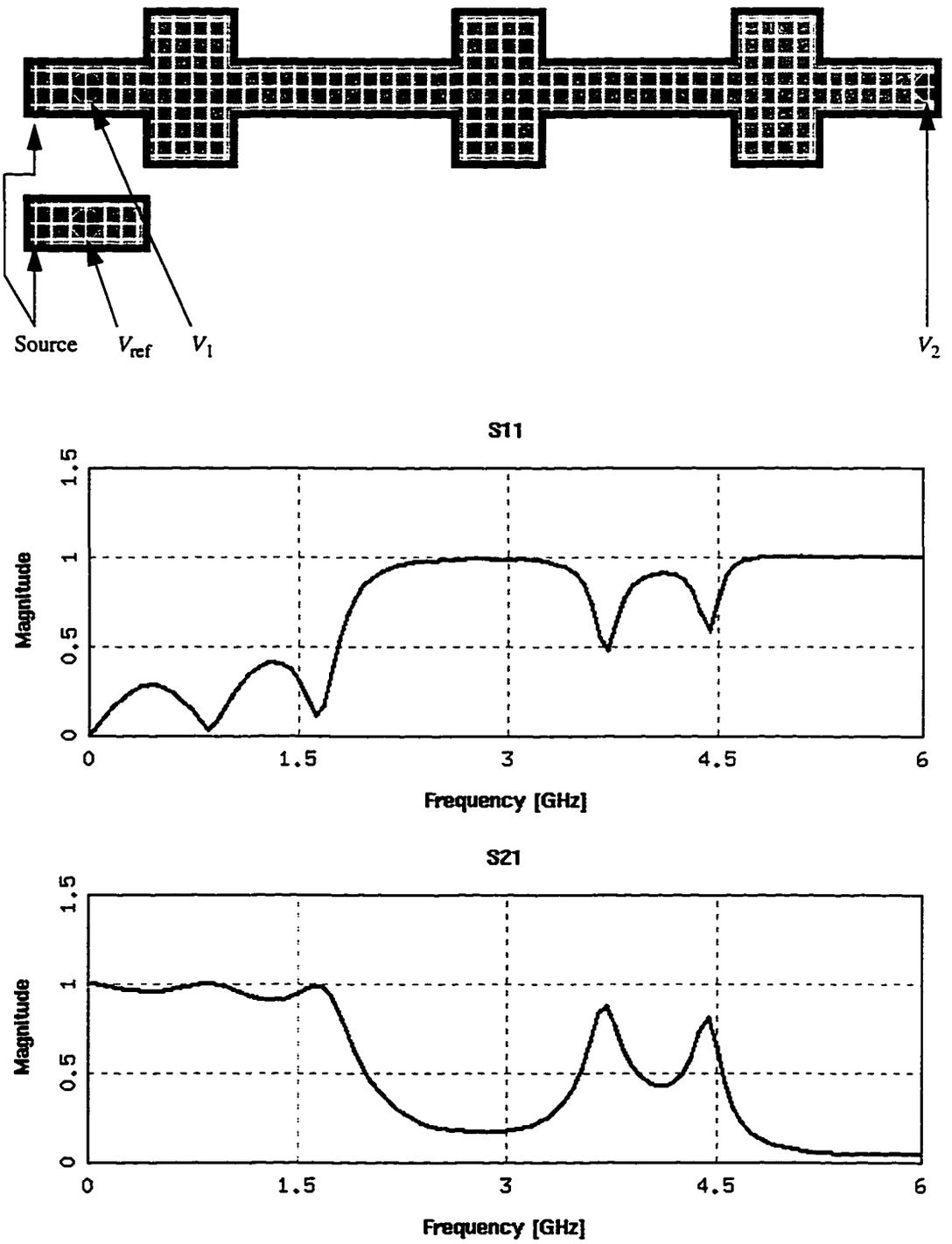


Figure 3.10: A typical arrangement for the computation of S-parameters. The structure is an equivalent two-dimensional TLM model of a microstrip low-pass filter and a separate matched reference section for the computation of the incident field.

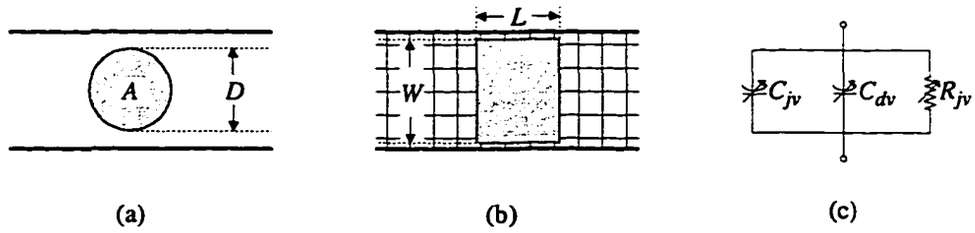


Figure 3.11: 2D-TLM simulation of a microstrip varactor multiplier.
(a) Packaged diode in the equivalent microstrip model.
(b) 2D-TLM mesh with stub loaded region simulating the diode.
(c) The equivalent circuit of the varactor diode.

3.9 Modelling of Non-Linear Devices

The constitutive parameters of the region represented by TLM nodes is related to the impedance values of the loading stubs, which can be recalculated and updated according to the local voltage and current at each time step. Non-linear structures and devices can therefore be modelled easily by updating the impedance values of these loading stubs.

In this section, the characteristic admittance of the shunt stub in a 2D-TLM region is made dependent on the instantaneous local electric field, [28] and [29], so that the region would behave like a simple microstrip varactor multiplier, Figure 3.11. The microstrip is replaced by an equivalent parallel-plate waveguide with magnetic side-walls for two-dimensional modelling (Figure 3.11b). The capacitance and conductance of the varactor diode are modelled as a small subregion embedded into the microstrip. It is essentially a small parallel-plate capacitor with an area A equal to the cross-sectional area of the diode package as shown in Figure 3.11a. The plate spacing equals the dielectric thickness h of the microstrip. Hence, the diode model is entirely two-dimensional.

Electrically, the diode is described by an equivalent circuit consisting of a voltage-dependent capacitance, C_{tot}:

$$C_{tot} = C_{jv} + C_{dv} \qquad \text{Eq 3.31}$$

in parallel with a voltage-dependent resistance R_{jv}, Figure 3.11c. C_{jv} and C_{dv} are the depletion layer capacitance and diffusion capacitance of the diode, respectively.

Under reverse bias, the depletion layer capacitance of a varactor diode can be represented as:

$$C_{jv} = \frac{C_{jo}}{(1 - v/\phi_o)^{\gamma}} \qquad \text{Eq 3.32}$$

where C_{jo} is the zero-bias capacitance, v is the junction voltage, ϕ_o is the built-in potential, and γ is 0.5 for Schottky barriers.

Under forward bias, Eq 3.32 rapidly becomes unsatisfactory because it gives infinite capacitance as v approach ϕ_o. This non-physical behaviour can be avoided by using a straight-line approximation suggested by Getreu [36] in the context of modelling bipolar transistors. For forward bias exceeding $\phi_o/2$, Eq 3.32 is replaced by:

$$C_{jv} = 2^{\gamma} C_{jo}\left[2\gamma\left(\frac{v}{\phi_o}\right) + (1-\gamma)\right] \qquad \text{for} \qquad v \geq \phi_o/2 \qquad \text{Eq 3.33}$$

Thus, for Schottky barriers with $\gamma = 0.5$, the following representation of the depletion layer capacitance is valid:

$$C_{jv} = \begin{cases} \dfrac{C_{jo}}{\sqrt{1 - v/\phi_o}} & \text{for} \quad v < \phi_o/2 \\ \sqrt{2} C_{jo}\left(\dfrac{v}{\phi_o} + \dfrac{1}{2}\right) & \text{for} \quad v \geq \phi_o/2 \end{cases} \qquad \text{Eq 3.34}$$

Note that the slope of C_{jv} is continuous across $v = \phi_o/2$. Under forward bias, the diffusion capacitance C_{dv} becomes important as well, and it is:

$$C_{dv} = C_{jo} \cdot \exp\left(\frac{ev}{kT}\right) \qquad \text{Eq 3.35}$$

where e is electron charge, k is Boltzmann's constant and T is temperature in ^{o}K. The voltage-dependent resistance of the diode can be written as:

$$R_{jv} = \left(\frac{kT}{eI_s}\right) \cdot \exp\left(-\frac{ev}{kT}\right) \qquad \text{Eq 3.36}$$

where I_s is the saturation current.

If the relative dielectric constant and conductivity of the diode region in the TLM mesh are functions of the voltage v, then the total capacitance and resistance of the diode region (ignoring the fringing capacitance) is:

$$C_{tot}(v) = \frac{A\varepsilon_o}{h}\varepsilon_r(v) \qquad R(v) = \frac{h}{A\sigma(v)} \qquad \text{Eq 3.37}$$

where A and h are the diode area and height, respectively. Combining Eq 3.31, Eq 3.34 and Eq 3.36 with Eq 3.37:

$$\varepsilon_r(v) = \begin{cases} \frac{h}{A\varepsilon_o}\left[\frac{C_{jo}}{\sqrt{1-v/\phi_o}} + C_{dv}\right] & \text{for} \quad v < \phi_o/2 \\ \frac{h}{A\varepsilon_o}\left[\sqrt{2}C_{jo}\left(\frac{v}{\phi_o} + \frac{1}{2}\right) + C_{dv}\right] & \text{for} \quad v \geq \phi_o/2 \end{cases} \qquad \text{Eq 3.38}$$

$$\sigma(v) = \frac{h}{A}\frac{eI_s}{kT} \cdot \exp\left(\frac{ev}{kT}\right)$$

Eq 3.11 and Eq 3.16 require the voltage dependent stub parameters in the diode region to be:

$$\begin{aligned} y_o(v) &= 4\left[\varepsilon_r(v) - 1\right] \\ g_o(v) &= \sqrt{2}\eta_o \Delta l \sigma(v) \end{aligned} \qquad \text{Eq 3.39}$$

In the TLM model it is convenient to approximate a normally circular diode region by a rectangular region of equal area A and width w as shown in Figure 3.11b. The length L of the region is then:

$$L = \frac{D^2\pi}{4w} \qquad \text{Eq 3.40}$$

or the closest integer multiple of the mesh parameter Δl. The value of $\varepsilon_r(v)$ and $\sigma(v)$ in Eq 3.38 should be computed by using:

$$A = mn(\Delta l)^2 \qquad \text{Eq 3.41}$$

where m = round($w/\Delta l$) and n = round($L/\Delta l$), respectively. The value of $y_o(v)$ and $g_o(v)$ in the region are computed and updated at each iteration based on the instantaneous voltage at a node in the center of the diode area. In this way, the diode capacitance and loss follow the voltage with an average delay of Δt_{wave} which is negligible since the periods of all frequencies of interest are much greater than that. Furthermore, in order to make $y_o(v)$ and $g_o(v)$ insensitive to voltage

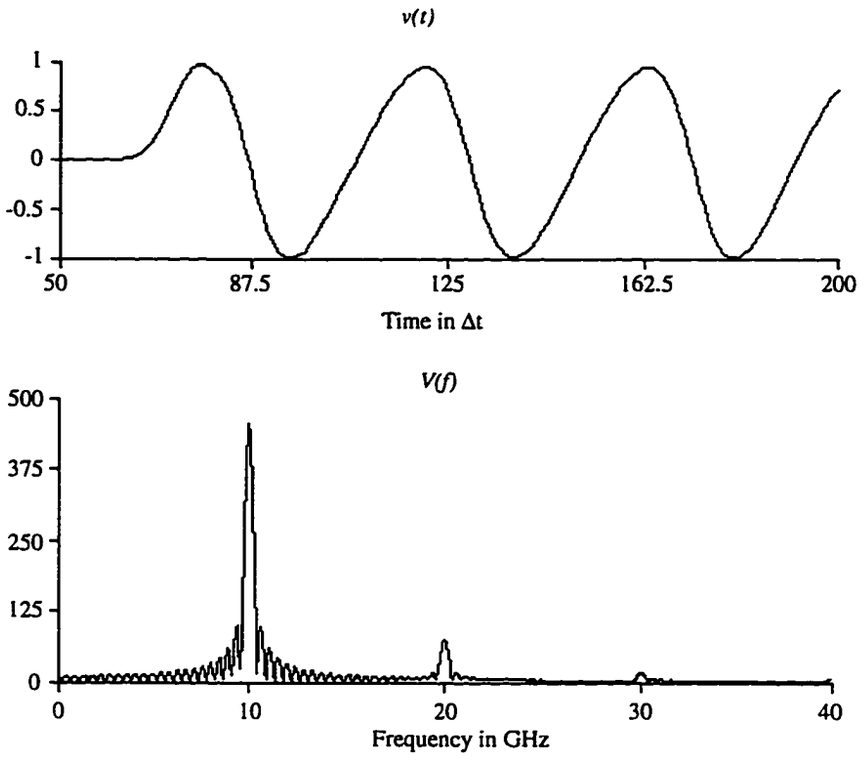


Figure 3.12: Time and frequency response of the frequency multiplier, depicted in Figure 3.11a, due to a 10 GHz sinusoidal excitation.

swings at a frequency much higher than the frequency range of interest, one can compute the average of that voltage over a number of time steps and compute $y_o(v)$ and $g_o(v)$ on the basis of this average value.

Figure 3.12 depicts the time domain response $v(t)$, obtained at the output end of the multiplier in Figure 3.11a, and its Fourier transform $V(f)$. The result is obtained by using a 10 GHz sinusoidal excitation signal. The curves in Figure 3.12 clearly show that higher order harmonics are generated by the nonlinear TLM diode model.

3.10 Conclusion

The fundamental theory of 2D-TLM, both the shunt and series node as well as some advanced features for the methods have been presented. The advanced features such as the recursive generation of Johns Matrix, computation of *S*-Parameter and modelling of non-linear devices are not just applicable to 2D-TLM only; they can be generalized to 3D-TLM as well as FDTD.

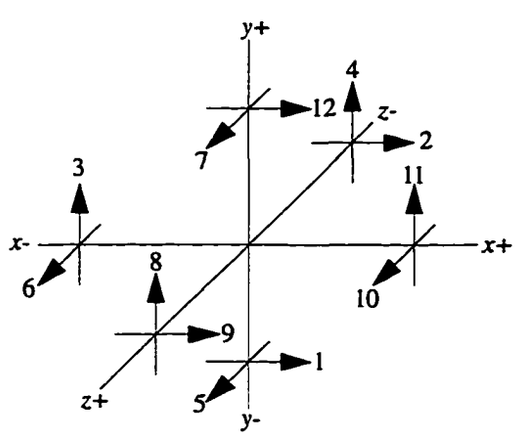
CHAPTER 4 The 3D-TLM Method

4.1 Introduction

The 3D-TLM method is of more practical importance than the 2D-TLM method because most wave problems cannot be formulated as two-dimensional problems. Examples are the rectangular waveguide magic-T and the via hole in a microstrip circuit board.

The equivalence between 3D-TLM and FDTD has been proven from various perspectives by different researchers, [4], [16], [19], [23] and [33]. Unlike FDTD and the expanded node 3D-TLM, in which the positions of the electric and magnetic field components are staggered in space, the symmetrical condensed node 3D-TLM (pioneered by P.B. Johns in the 80's [31]) defines all electric and magnetic field components at the same point in space. This allows a more accurate description of the positions of boundaries and hence directly affects the accuracy of the simulation results. For this reason, most of the recent advancements in 3D-TLM are focused on the symmetrical condensed node 3D-TLM; this is also true for the new contributions presented in this chapter.

The following sections begin with a discussion of P.B Johns original 12×12 scattering matrix for the symmetrical condensed node 3D-TLM. Two new numbering schemes for the 3D-TLM scattering matrix as well as their advantages and applications are presented before the conclusion.



(a)

	1	2	3	4	5	6	7	8	9	10	11	12
1		a	a						a		-a	
2	a					a				-a		a
3	a			a				a				-a
4			a		a		-a				a	
5				a		a		-a		a		
6		a			a		a		-a			
7				-a		a		a		a		
8			a		-a		a				a	
9	a					-a				a		a
10		-a			a		a		a			
11	-a			a				a				a
12		a	-a						a		a	

(b)

Figure 4.1: P.B. Johns' 3D-TLM symmetrical condensed node for free space.
(a) P.B. Johns' impulse numbering scheme.
(b) The associated 12×12 scattering matrix; where $a = 0.5$.

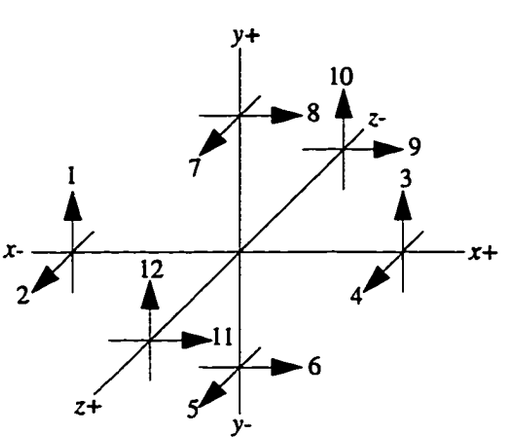
4.2 The Original 3D-TLM Scattering Matrix

In early 80's, P.B. Johns published the first paper on the symmetrical condensed node 3D-TLM [31]. In the case for free space modelling, the node consists of 12 link lines which are organized as shown in Figure 4.1a. The incident and reflected voltage impulses are related to each other via the familiar scattering matrix equation:

$$_{k+1}V^r = S \cdot {}_kV^i \qquad \text{Eq 4.1}$$

where S is the scattering matrix shown in Figure 4.1b. The impulse numbering scheme shown in Figure 4.1a is not quite systematic. As a result the scattering matrix elements do not relate to each other in a systematic manner, the scattering matrix thus fails to give special insight into the physics of the node.

Two alternative numbering schemes are presented in the following section, one was published in [9] by Hoefer & So and the other one was first used by Russer in [18] and was then adopted by his research group. These two new schemes are quite systematic, hence they provide insight into the scattering procedure and facilitate the formulation of efficient scattering algorithms for computer implementation as well as derivation of new scattering matrices for the condensed nodes with special properties.



(a)

	1	2	3	4	5	6	7	8	9	10	11	12
1						a		-a		a		a
2					a		a		a		-a	
3						-a		a		a		a
4					a		a		-a		a	
5		a		a						a		-a
6	a		-a						a		a	
7		a		a						-a		a
8	-a		a						a		a	
9		a		-a		a		a				
10	a		a		a		-a					
11		-a		a		a		a				
12	a		a		-a		a					

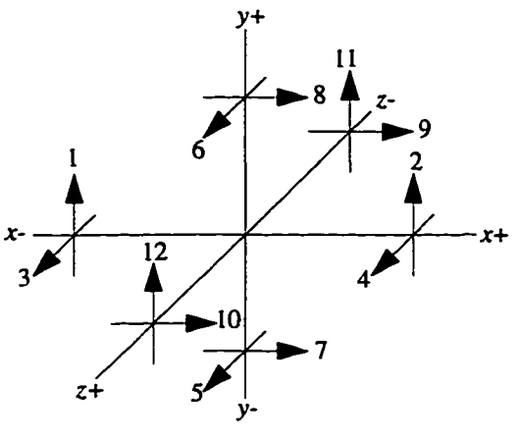
(b)

Figure 4.2: P.B. Johns' 3D-TLM symmetrical condensed node for free space.
(a) Hoefer & So's impulse numbering scheme.
(b) The associated 12×12 scattering matrix; where $a = 0.5$.

4.3 New 3D Symmetrical Condensed Node Impulse Numbering Schemes

Two symmetrical condensed node impulse numbering schemes which were developed independently by two research groups are presented in this section. The schemes for the node without stubs will be presented first and the idea will then be generalized to the fully loaded 3D-TLM lossy symmetrical condensed node.

The numbering scheme shown in Figure 4.2a, developed by Hoefer & So, is based on the right-hand rule used in vector calculus. The impulses travelling along the x-axis are incident from either the x- or x+ direction and their polarizations are parallel to either the y or z-axis (Figure 4.2a). These impulses are named V_y^{x-}, V_z^{x-},V_y^{x+} and V_z^{x+}; V_y^{x-} stands for the voltage impulse incident along the x-axis form the negative side and having a polarization parallel to the y-axis, and so on. Similarly the impulses travelling along the y and z-axes are named V_z^{y-}, V_x^{y-}, V_z^{y+}, V_x^{y+}, V_x^{z-}, V_y^{z-}, V_x^{z+} and V_y^{z+}, respectively. These voltage names correspond to V_3, V_6, V_{11}, V_{10}, V_5, V_1, V_7, V_{12}, V_2, V_4, V_9 and V_8 in Figure 4.1a. The new reorganized scattering matrix S is shown in Figure 4.2b. Another impulse numbering scheme developed by Russer is shown Figure 4.3. The scheme is similar to that of Hoefer & So, the only



(a)

	1	2	3	4	5	6	7	8	9	10	11	12
1							a	$-a$			a	a
2							$-a$	a			a	a
3					a	a			a	$-a$		
4					a	a			$-a$	a		
5			a	a							a	$-a$
6			a	a							$-a$	a
7	a	$-a$							a	a		
8	$-a$	a							a	a		
9			a	$-a$			a	a				
10			$-a$	a			a	a				
11	a	a			a	$-a$						
12	a	a			$-a$	a						

(b)

Figure 4.3: P.B. Johns' 3D-TLM symmetrical condensed node for free space.
(a) Russer's impulse numbering scheme.
(b) The associated 12×12 scattering matrix; where $a = 0.5$.

difference is it groups the impulses on a transmission line according to their polarizations.

The scattering matrices obtained from Hoefer & So and Russer's impulse numbering schemes have some interesting properties. This can be seen easily from the scattering matrix in Figure 4.2b and Figure 4.3b.

(1) The 12×12 scattering matrix can be partitioned in to a 3×3 matrix system; each element in that matrix system is a 4×4 matrix:

$$\boldsymbol{S} = \boldsymbol{S}^{-1} = \boldsymbol{S}^{T} = \frac{1}{2}\begin{bmatrix} & \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B} & & \boldsymbol{A} \\ \boldsymbol{A} & \boldsymbol{B} & \end{bmatrix} \qquad \text{Eq 4.2}$$

(2) The submatrices in the matrix system are $\boldsymbol{A}$ and $\boldsymbol{B}$, and $\boldsymbol{B} = \boldsymbol{A}^{T}$. This is because the node is symmetrical, and the voltage impulses are ordered in a systematic arrangement.

(3) The 4×4 diagonal matrices of this 3×3 matrix system are zero matrices. This is so because impulses travelling along one axis are scattered into the other two axial directions only.

(4) The elements of the second and third row of the matrix system are the cyclic permutation of the elements of the first row; this is a direct benefit of ordering the voltage impulses in a systematic arrangement.

Impulse Names	V_y^{x-}	V_y^{x+}	V_z^{x-}	V_z^{x+}	V_z^{y-}	V_z^{y+}	V_x^{y-}	V_x^{y+}	V_x^{z-}	V_x^{z+}	V_y^{z-}	V_y^{z+}
Johns	V_3	V_{11}	V_6	V_{10}	V_5	V_7	V_1	V_{12}	V_2	V_9	V_4	V_8
Hoefer & So	V_1	V_3	V_2	V_4	V_5	V_7	V_6	V_8	V_9	V_{11}	V_{10}	V_{12}
Russer	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{11}	V_{12}

Table 4.1: The relationships among Johns, Hoefer & So and Russer's node numbering schemes for the 3D-TLM symmetrical condensed node.

In addition, the submatrices resulting from Russer's impulse numbering scheme are also block matrices. This property may be useful in some special programming environment and in describing the theory of 3D-TLM using matrix properties.

The relationship among Johns, Hoefer & So and Russer's impulse numbering schemes are shown in Table 4.1. Since Russer's scheme has more special property, his numbering scheme is adopted in this thesis as well.

Russer's numbering scheme can be extended to the fully loaded 3D-TLM lossy symmetrical condensed node. The scattering matrix for such a node using Johns' numbering scheme is given in [4] and is reproduced in Figure 4.4.

#			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	o		x	x	y	y	z	z	z	y	x	z	y	x	x	y	z			
		s	z	y	z	x	x	y	x	x	y	y	z	z				x	y	z
1	x	z	a	b	d						b		-d	c	e					f
2	x	y	b	a				d			c	-d		b	e				-f	
3	y	z	d		a	b				b			c	-d		e				-f
4	y	x			b	a	d		-d	c			b			e		f		
5	z	x				d	a	b	c	-d		b					e	-f		
6	z	y		d			b	a	b		-d	c					e		f	
7	z	x				-d	c	b	a	d		b					e	f		
8	y	x			b	c	-d		d	a			b			e		-f		
9	x	y	b	c				-d			a	d		b	e				f	
10	z	y		-d			b	c	b		d	a					e		-f	
11	y	z	-d		c	b				b			a	d		e				f
12	x	z	c	b	-d						b		d	a	e					-f
13	x		e	e							e			e	g					
14	y				e	e				e			e			g				
15	z						e	e	e			e					g			
16		x				f	-f		f	-f								h		
17		y		-f				f			f	-f							h	
18		z	f		-f								f	-f						h

Figure 4.4: The 18 × 18 scattering matrix for the fully loaded 3D-TLM lossy symmetrical condensed node using Johns numbering scheme.

Impulse Names	V_y^{x-}	V_y^{x+}	V_x^e	V_x^h	V_z^{x-}	V_z^{x+}	V_z^{y-}	V_z^{y+}	V_y^e	V_y^h	V_x^{y-}	V_x^{y+}	V_x^{z-}	V_x^{z+}	V_z^e	V_z^h	V_y^{z-}	V_y^{z+}
Johns	V_3	V_{11}	V_{13}	V_{16}	V_6	V_{10}	V_5	V_7	V_{14}	V_{17}	V_1	V_{12}	V_2	V_9	V_{15}	V_{18}	V_4	V_8
RHS	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{11}	V_{12}	V_{13}	V_{14}	V_{15}	V_{16}	V_{17}	V_{18}

Table 4.2: The relationship between Johns and RHS impulse numbering schemes for the 3D-TLM lossy symmetrical condensed node.

Similar to the scattering matrix shown in Figure 4.1b, the scattering matrix shown in Figure 4.4 does not have any obvious property. By extending Hoefer & So and Russer's numbering schemes, scattering matrices with the special properties as described previously can be obtained. Figure 4.5 depicts the scattering matrix using an impulse numbering scheme based on Russer and Hoefer & So's schemes — from now on, it is called the RHS scheme. The relationship between Johns and RHS scheme is shown in Table 4.2.

#			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	o		*y*	*y*	*x*		*z*	*z*	*z*	*z*	*y*		*x*	*x*	*x*	*x*	*z*		*y*	*y*
		s	*z*	*z*		*x*	*y*	*y*	*x*	*x*		*y*	*z*	*z*	*y*	*y*		*z*	*x*	*x*
1	*y*	*z*	*a*	*c*							*e*		*d*	*-d*				*-f*	*b*	*b*
2	*y*	*z*	*c*	*a*							*e*		*-d*	*d*				*f*	*b*	*b*
3	*x*				*g*								*e*	*e*	*e*	*e*				
4		*x*				*h*			*-f*	*f*									*f*	*-f*
5	*z*	*y*					*a*	*c*	*b*	*b*		*f*			*d*	*-d*	*e*			
6	*z*	*y*					*c*	*a*	*b*	*b*		*-f*			*-d*	*d*	*e*			
7	*z*	*x*				*-f*	*b*	*b*	*a*	*c*							*e*		*d*	*-d*
8	*z*	*x*				*f*	*b*	*b*	*c*	*a*							*e*		*-d*	*d*
9	*y*		*e*	*e*							*g*								*e*	*e*
10		*y*					*f*	*-f*				*h*			*-f*	*f*				
11	*x*	*z*	*d*	*-d*	*e*								*a*	*c*	*b*	*b*		*f*		
12	*x*	*z*	*-d*	*d*	*e*								*c*	*a*	*b*	*b*		*-f*		
13	*x*	*y*			*e*		*d*	*-d*				*-f*	*b*	*b*	*a*	*c*				
14	*x*	*y*			*e*		*-d*	*d*				*f*	*b*	*b*	*c*	*a*				
15	*z*						*e*	*e*	*e*	*e*							*g*			
16		*z*	*-f*	*f*									*f*	*-f*				*h*		
17	*y*	*x*	*b*	*b*		*f*			*d*	*-d*	*e*								*a*	*c*
18	*y*	*x*	*b*	*b*		*-f*			*-d*	*d*	*e*								*c*	*a*

Figure 4.5: The 18 × 18 scattering matrix for the fully loaded 3D-TLM lossy symmetrical condensed node using RHS numbering scheme.

The definitions of the variables used in Figure 4.4 and 4.5 are given in [4]; they are reproduced in below for convenience:

$$
\begin{aligned}
a &= -\frac{Y+g_e}{2(Y+g_e+4)} + \frac{Z+g_m}{2(Z+g_m+4)} \\
b &= \frac{2}{Y+g_e+4} \\
c &= -\frac{Y+g_e}{2(Y+g_e+4)} - \frac{Z+g_m}{2(Z+g_m+4)} \\
d &= \frac{2}{Z+g_m+4} \\
e &= \sqrt{Y}\cdot b \\
f &= \sqrt{Z}\cdot d \\
g &= \frac{Y-g_e-4}{Y+g_e+4} \\
h &= -\frac{Z-g_m-4}{Z+g_m+4}
\end{aligned}
\qquad \text{Eq 4.3}
$$

and

$$
\begin{aligned}
Y_x &= 4\left(\varepsilon_x \frac{vw}{u\Delta l} - 1\right) & Y_y &= 4\left(\varepsilon_y \frac{uw}{v\Delta l} - 1\right) & Y_z &= 4\left(\varepsilon_z \frac{uv}{w\Delta l} - 1\right) \\
Z_x &= 4\left(\mu_x \frac{vw}{u\Delta l} - 1\right) & Z_y &= 4\left(\mu_y \frac{uw}{v\Delta l} - 1\right) & Z_z &= 4\left(\mu_z \frac{uv}{w\Delta l} - 1\right) \\
g_{ex} &= \sigma_{ex} \frac{vw}{u} Z_o & g_{ey} &= \sigma_{ey} \frac{uw}{v} Z_o & g_{ez} &= \sigma_{ez} \frac{uv}{w} Z_o \\
g_{mx} &= \sigma_{mx} \frac{vw}{uZ_o} & g_{my} &= \sigma_{my} \frac{uw}{vZ_o} & g_{mz} &= \sigma_{mz} \frac{uv}{wZ_o}
\end{aligned}
\qquad \text{Eq 4.4}
$$

Where

- u, v and w are the size of the unit cell in the x, y and z-direction, respectively.
- Y, g_e, Z and g_m take a subscript corresponding to their associated open or short circuited stubs.

For instance:

$$
S_{11} = a_{yz} = -\frac{Y_y+g_{ey}}{2(Y_y+g_{ey}+4)} + \frac{Z_z+g_{mz}}{2(Z_z+g_{mz}+4)}
$$

$$
S_{55} = a_{zy} = -\frac{Y_z+g_{ez}}{2(Y_z+g_{ez}+4)} + \frac{Z_y+g_{my}}{2(Z_y+g_{my}+4)}
$$

The field components can be calculated from the voltage impulses; using the RHS numbering scheme, the field are:

$$E_x = \frac{2u\,(V_{11}^i + V_{12}^i + V_{13}^i + V_{14}^i + \sqrt{Y_x}\cdot V_3^i)}{(Y_x + g_{ex} + 4)}$$

$$E_y = \frac{2v\,(V_{17}^i + V_{18}^i + V_1^i + V_2^i + \sqrt{Y_y}\cdot V_9^i)}{(Y_y + g_{ey} + 4)}$$

$$E_z = \frac{2w\,(V_5^i + V_6^i + V_7^i + V_8^i + \sqrt{Y_z}\cdot V_{15}^i)}{(Y_z + g_{ez} + 4)}$$

$$H_x = \frac{2u\,(-V_{17}^i + V_{18}^i + V_7^i - V_8^i + \sqrt{Z_x}\cdot V_4^i)}{Z_o\,(Z_x + g_{mx} + 4)}$$

$$H_y = \frac{2v\,(-V_5^i + V_6^i + V_{13}^i - V_{14}^i + \sqrt{Z_y}\cdot V_{10}^i)}{Z_o\,(Z_y + g_{my} + 4)}$$

$$H_z = \frac{2w\,(-V_{11}^i + V_{12}^i + V_1^i - V_2^i + \sqrt{Z_z}\cdot V_{16}^i)}{Z_o\,(Z_z + g_{mz} + 4)}$$

Eq 4.5

4.4 Derivation of an Efficient Scattering Algorithm for the Non Stub Loaded Symmetrical Condensed Node

The scattering matrix in Figure 4.3b shown clearly that, the sum and difference of a number of voltage impulses can be used to perform the impulse scattering. Let's use the first two voltage impulses to illustrate this idea.

Applying the scattering matrix in Figure 4.3b to Eq 4.1, the first two voltage impulses at time $k+1$ are:

$$\begin{aligned} V_1^r &= 0.5 \times (V_7^i - V_8^i + V_{11}^i + V_{12}^i) \\ V_2^r &= 0.5 \times (V_8^i - V_7^i + V_{11}^i + V_{12}^i) \end{aligned} \qquad \text{Eq 4.6}$$

Since the difference and sum of two pairs of impulses are used more than once, it would be beneficial to compute and store them in temporary variables. Then the above equations can be rewritten as:

$$\begin{aligned} V_d &= V_7^i - V_8^i \\ V_s &= V_{11}^i + V_{12}^i \\ V_1^r &= 0.5 \times (V_d + V_s) \\ V_2^r &= 0.5 \times (V_s - V_d) = V_1^r - V_d \end{aligned} \qquad \text{Eq 4.7}$$

The above algorithm takes only 1 multiplication, 2 additions and 2 subtractions to compute the first two reflected voltages. The same algorithm can be applied to calculate the remaining voltages. The total number of floating point operations are then: 6 multiplications, 12 additions and 12 subtractions. If the scattering matrix in Figure 4.1b is applied to Eq 4.1 directly, then:

$$V_1 = 0.5 \times (V_2 + V_3 + V_9 - V_{11}) \qquad \text{Eq 4.8}$$

The remaining voltages can be computed similarly. The total number of floating point operations are: 12 multiplications, 24 additions and 12 subtractions.

The improved algorithm reduced the total number of multiplications and additions by 50%, which is a substantial saving of CPU time.

4.5 Impulse Splitting Procedure

The scattering matrix for 3D-TLM can be expanded by logically splitting each impulse travelling between nodes into two parts, say a and b. By using the node shown in Figure 4.3a as an example, the result of such a splitting procedure gives rise to the impulse distribution shown in Figure 4.6. Replacing the elements in $\boldsymbol{A}$ and $\boldsymbol{B}$ of Eq 4.2 with their equivalent 2×2 zero and identity matrices, the 4×4 matrices of the 3×3 matrix system in Eq 4.2 become 8×8 matrices.

This expanded scattering matrix can be used to derive new scattering matrices for condensed nodes with special properties. For example, it can be used to find, almost by inspection, the scattering matrix of a condensed node with a perfect conducting metallic plate of zero thickness in its xz-plane [45] (Figure 4.7). These special nodes can be used in conjunction with the boundaries placed halfway between nodes to model the sizes and positions of metallic structures with good accuracy without using an unnecessarily refined mesh. The impulse splitting operation can be generalized to model some special boundary properties, such as sharp edges and corners, finite conductivity, skin effect and implementation of discrete devices into a TLM network, which can not be modelled easily and accurately by placing the boundary half-way between nodes.

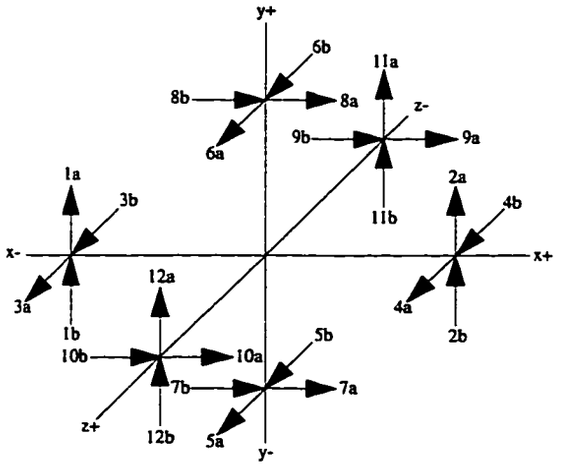


Figure 4.6: Logically splitting impulses in Figure 4.3a into two parts — a and b. The total voltage of each individual impulse is the sum of part a and part b; $V_1=V_{1a}+V_{1b}$, and so on.

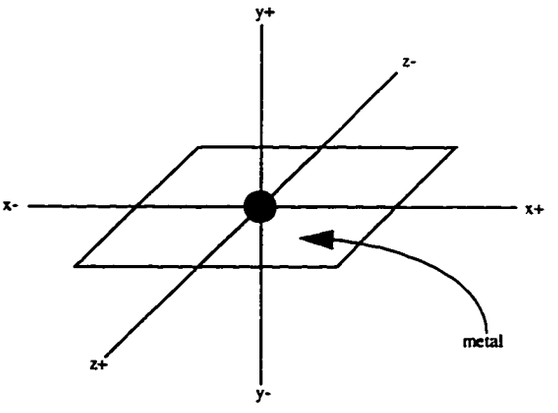


Figure 4.7: Nielsen's half-node for S_{hy} [45].

4.6 The Scattering Matrix for a Half-Node

The perfectly conducting metallic plate in the xz-plane of the condensed node in Figure 4.7 causes V_3, V_4, V_9 and V_{10} to be zero. To make sure that these voltages are zero, all the entries in the scattering matrix that will cause energy to be scattered into them as well as all the entries that will cause energy to be scattered out of them must be set to zero. The entries in the scattering matrix that will cause top-down interaction must also be set to zero. Finally, because of the presence of the conducting plate, energy which would normally be scattered to V_{8b} from V_{1b} must now go to V_{7b}, and so on. The resulting scattering matrix S is:

$$S = \frac{1}{2}\begin{bmatrix} & A & B \\ C & & D \\ B^T & E & \end{bmatrix} \qquad \text{Eq 4.9}$$

where A, B, C, D and E are:

$$A = \begin{bmatrix} [\,] & [\,] & [\,] & [-1] \\ [\,] & [\,] & [-1] & [1] \\ [\,] & [\,] & [\,] & [\,] \\ [\,] & [\,] & [\,] & [\,] \end{bmatrix} \qquad B = \begin{bmatrix} [\,] & [\,] & [1] & [1] \\ [\,] & [\,] & [1] & [1] \\ [\,] & [\,] & [\,] & [\,] \\ [\,] & [\,] & [\,] & [\,] \end{bmatrix}$$

$$C = \begin{bmatrix} [\,] & [\,] & [\,] & [\,] \\ [\,] & [\,] & [\,] & [\,] \\ [2] & [-2] & [\,] & [\,] \\ [-2] & [2] & [\,] & [\,] \end{bmatrix} \qquad D = \begin{bmatrix} [\,] & [\,] & [2] & [-2] \\ [\,] & [\,] & [-2] & [2] \\ [\,] & [\,] & [\,] & [\,] \\ [\,] & [\,] & [\,] & [\,] \end{bmatrix}$$

$$E = \begin{bmatrix} [\,] & [\,] & [\,] & [\,] \\ [\,] & [\,] & [\,] & [\,] \\ [\,1] & [-1] & [\,] & [\,] \\ [-1] & [1] & [\,] & [\,] \end{bmatrix}$$

This scattering matrix is equivalent to the S_{hy} matrix in [45], but is derived with considerably less effort and is easily permuted to find S_{hx} and S_{hz}.

4.7 Conclusion

This chapter covered some basic theory and algorithms of the symmetrical condensed node 3D-TLM. A number of special impulse numbering schemes for the node are also presented. The free space scattering matrices obtained via these numbering schemes shown that the sum and difference of a number of voltages can be used to derive an efficient scattering algorithm; it has been done in *Section 4.4*. An impulse splitting procedure is also presented; the procedure is applied to derive the scattering matrix for a 3D-TLM node cut in half in the *xz*-plane by a perfect electric wall.

CHAPTER 5 Characteristics of the TLM Method

5.1 Introduction

So far, only the modelling capabilities and properties of 2D-TLM and 3D-TLM have been discussed. The sources of error and the unique characteristics of the methods have been purposely left out in the pervious chapters. It is time to give a general discussion of these topics; the material to be discussed is applicable to both the two-dimensional and three-dimensional cases.

Similar to FDTD, or other numerical methods which requires discretization of space and time, TLM is subject to various sources of error. However, the errors introduced by these sources can be significantly reduced as long as the method is applied properly. On the other hand, TLM does have some superb characteristics that the other numerical methods do not have. These superb characteristics make TLM a more attractive method in certain types of modelling applications.

The first part of this chapter gives an overview of the sources of errors of TLM and ways to reduce their effects on the accuracy of the simulation results. In certain situations, some of their effects can actually be compensated by using the appropriated dispersion formulas. The second part of this chapter outlines the superb characteristics of TLM together with the corresponding areas of application.

5.2 Source of Errors

TLM has three main sources of error which must be observed in order to yield reliable and accurate results. They are truncation error, velocity error and coarseness error. These sources of error are not unique to TLM; truncation error is an inherit characteristic of all time domain methods, velocity and coarseness errors are present in FDTD as well.

5.2.1 Truncation Error

Since TLM is a time domain method, and due to practical reason one must stop the computation in time; this lead to the so-called truncation error. Figure 5.1 depicts the typical effects of this error on the frequency domain response of a resonator. Due to the finite duration of the time domain response, its Fourier transform is not a line

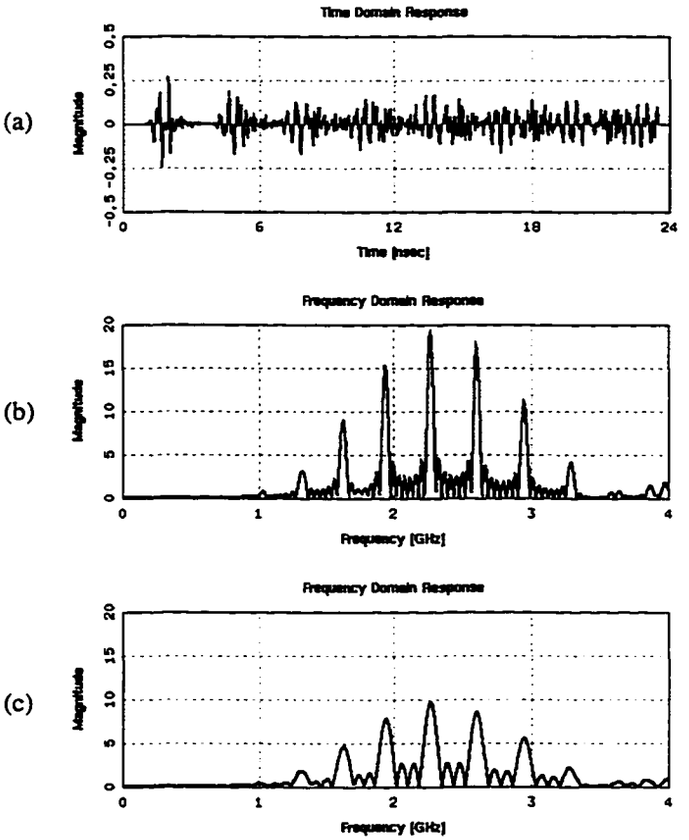


Figure 5.1: The effect of truncating a time domain response on its Fourier transform.
(a) A typical time domain response in a resonator.
(b) DFT of the time domain response in (a), which has 1000 time samples.
(c) DFT of a truncated time domain response — the first 500 samples in (a).

spectrum that contains only the resonant frequencies supported by the resonator, but rather a superposition of the *sinc* functions which may interfere with each other such that their maxima are slightly shifted. A systematic way to correct truncation error in this kind of situation is given in [37]. Figure 5.2 depicts the typical effects of truncation error on the frequency domain response of the structures with absorbing boundaries. In general, when enough time samples are obtained from the TLM simulation, the corresponding frequency response should converge to a smooth curve with no unexplainable ripples.

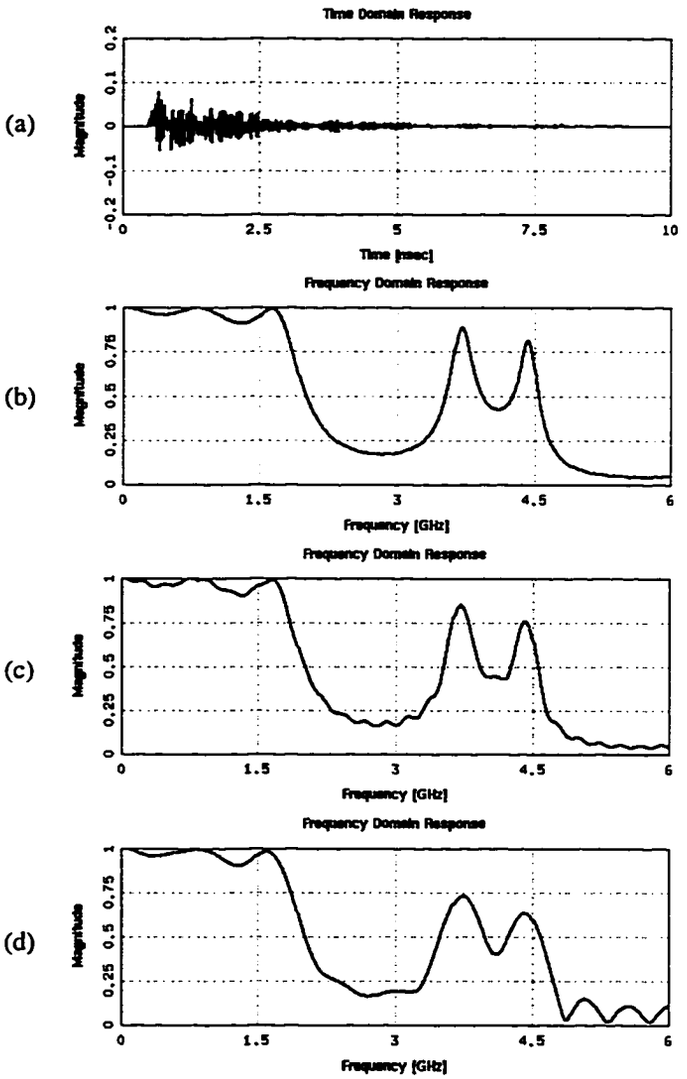


Figure 5.2: The effect of truncating a time domain response on its Fourier transform.
(a) A typical time domain response for structures with absorbing boundaries.
(b) DFT of the time domain response in (a), which has 4000 time samples.
(c) DFT of a truncated time domain response — the first 2000 samples in (a).
(d) DFT of a truncated time domain response — the first 1000 samples in (a).

5.2.2 Velocity Error

If the wavelength in the TLM network is large when compared with the discretization parameter, Δl, of the TLM mesh, then it can be assumed that the wave velocity in the TLM mesh is independent of frequency; i.e. the TLM mesh behaviour like a continuous nondispersive medium. However, when the wavelength decreases, the discrete behaviour of the mesh becomes more visible. As a result, the wave velocity becomes dispersive and depends on the direction of propagation. In TLM literature, this is known as velocity error.

Velocity error can be reduced by choosing a dense mesh such that the ratio between Δl and the guided wavelength, λ_g, is much less than one. The rule of thumb is $\Delta l/\lambda_g \leq 0.1$. Furthermore, if the wavefront in the structure propagate essentially in a single direction, the velocity error can be corrected using the appropriate dispersion relations, this is discussed extensively in [48].

5.2.3 Coarseness Error

Coarseness error occurs when a TLM mesh is too coarse to:

- Resolve the fields with high spatial frequency and/or non-linear spatial variation at corners and wedges of a structure accurately.
- Represent the fine details of a structure properly.
- Position the various elements of a structure precisely.

Coarseness error is the dominant source of inaccuracies when analyzing structures with thin metallic strips, especially, when the sizes and positions of the strips cannot be represented properly by the discretized space. A possible but impractical solution to this is to use a very fine mesh. However, this would lead to very large memory and CPU time requirements. An adaptive mesh (see *Chapter 6*) can be used to model the sizes and positions of the strips properly, but this could not resolve the fields with high spatial frequency and/or non-linear spatial variation properly. Al-Mukhtar's Graded mesh [34] can be introduced to provide higher resolution in the high spatial frequency region, but that requires complicated programming. A practical method, proposed by Shih and Hoefer [35] is to analysis the structure with successively refined meshes; and compute the final result by applying the intermediate results to an extrapolation procedure. This method can be used in conjunction with adaptive meshes and Al-Mukhtar's Graded mesh to significantly reduce the coarseness error.

5.3 Superb Characteristics of TLM

Despite TLM has the above mentioned sources of error, the method does have some superb characteristics which set TLM apart from other numerical methods.

5.3.1 Unambiguous Boundary Conditions

Unlike FDTD or the expanded node 3D-TLM, in which the electric and magnetic field components are staggered in space, both 2D-TLM and 3D-TLM (symmetrical condensed node) algorithms define the electric and magnetic field components at the same point in space. This allows a more accurate description of the positions of boundaries and hence directly affects the accuracy of the simulation result.

5.3.2 Cause-and-Effect Mechanism

TLM has a cause-and-effect mechanism. Cause and effect are easily identifiable. Therefore, partitioning of big structures into smaller substructures for simulations and then combining the responses via convolution to compute the final response is a straightforward procedure in TLM. This is one of the basic properties of TLM which Johns Matrix and Diakoptics [28] rely upon. Applying Johns Matrix or Diakoptics to FDTD requires extra processing work at the interfaces between the substructures.

5.3.3 Conservation of Energy and Numerical Stability

The scattering procedure of TLM satisfies the law of conservation of energy; i.e. the system has the same amount of energy before and after the TLM scattering. Furthermore, all the coefficients in the scattering matrix are less than unity; this makes TLM less susceptible to instability problems related the property of finite precision computer arithmetic. In fact, numerical unstable TLM simulation* has not been reported in literature. On the other hand, the relationship between numerical stability and the simulation time step, Δt, is an issue that cannot be ignored in the FDTD area.

---

* TLM simulation could become unstable if energy is continuously injected into the system by either:
(a) coupling the TLM mesh to a continuous source,
(b) placing active elements in the system, or
(c) coupling the TLM mesh to any element which is itself numerical unstable.

5.3.4 Reversible Scattering Processes

From a purely mathematical point of view, TLM has another unique characteristic — the TLM process is reversible in time [20], [48]. This is true because the scattering matrix of the TLM process is invertible; in the lossless situation, the scattering matrices are self-inverse.

The time inversion of the TLM process can be understood by considering the following scattering equation:

$$ {}_{k+1}V^{r} = S \cdot {}_{k}V^{i} \qquad \text{Eq 5.1} $$

The equation relates the reflected voltage impulses at time $(k+1)\Delta t$ to the incident voltage impulses at the previous time step $k\Delta t$. For 2D-TLM in free space, it can be easily verified that the impulse scattering matrix is identical to its inverse, i.e:

$$ S = S^{-1} = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \qquad \text{Eq 5.2} $$

Therefore, the reverse scattering process is:

$$ {}_{k}V^{i} = S^{-1} \cdot {}_{k+1}V^{r} = S \cdot {}_{k+1}V^{r} \qquad \text{Eq 5.3} $$

This implies that the 2D-TLM process can actually be reversed without any change in the scattering algorithm. Practically speaking, once a forward TLM process has been stopped, a backward TLM process can be started be reversing the execution order of the scattering and connection equations. This time reversal process has great potential for the synthesis of electromagnetic structures, [13].

5.4 Conclusion

The sources of errors of the TLM method have been discussed; they must be observed properly in order to obtain reliable and accurate results by using the TLM method. TLM also has some special characteristics such as the *cause-and-effect* and *time-reversible* properties which open special areas of applications for the TLM method; for instance, synthesis of electromagnetic structures via the time reversal TLM process is just one of them.

CHAPTER 6 Special Modelling Features

6.1 Introduction

The fundamental TLM features presented in the previous chapters are sufficient to model realistic design problems as long as the structures under study can be discretized easily with the staircase representation. In general, a smaller Δl gives a better representation but requires a more difficult discretization process. However, for efficiency reasons, one should not use a Δl that is unnecessary small.

It is also very common that structures with complicated boundaries cannot be properly discretized by a mesh with a regular tile pattern. Examples are the arbitrary pattern often found in microstrip printed circuit boards or a waveguide filter with tuning screws. In some restricted situations, elliptical or circular meshes can be used to provide a conformal representation of the structures. However, interfacing these meshes with themselves or with the rectangular meshes is not a trivial task. Another way to provide conformal representation of these structures is via local mesh refinement. However, the local submeshes always have smaller time steps than the global mesh; and the smallest time step among them must be used. This slows down the whole simulation process to an unacceptable level.

Automatic discretization of elements with an adaptive mesh is one of the solutions to the above problems.

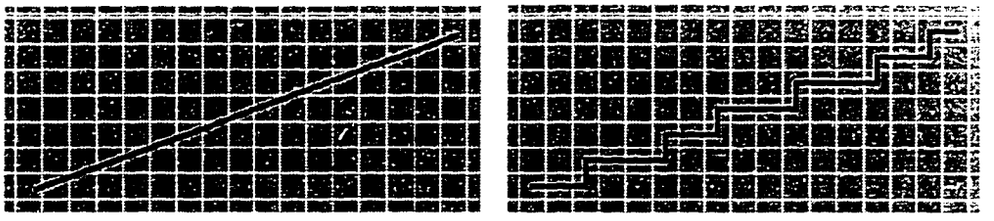


Figure 6.1: A straight boundary and its staircase approximation in the discretized coordinates. The white mesh lines represent the TLM link lines, and the intersections of these lines are the TLM nodes.

6.2 Automatic Discretization of Structures

One of the difficulties of using TLM and similar numerical methods is to represent objects in the discrete coordinate system. This is not a problem if the objects to be modelled need to be discretized only once; however, this is generally not the case. In many situations, one needs to solve the same problem with successively finer meshes, and the results can be used in the Richardson extrapolation procedure [43] to estimate the result one would obtain from an infinitely fine mesh. Therefore, automatic discretization of elements is an essential feature of an advanced modelling tool.

Figure 6.1 depicts a straight boundary and its staircase approximation in discretized coordinates. The white mesh lines represent the TLM link lines and the intersection of these lines are the TLM nodes. The staircase approximation of the boundary is created by placing horizontal and vertical boundaries of $1\Delta l$ long half-way between two nodes where the link line connecting them is cut by the boundary. This simple algorithm works as long as the boundaries to be discretized do not intersect directly with the TLM nodes; if they do, it fails completely because it has no provision to decide where to place the $1\Delta l$ long boundaries (Figure 6.2).

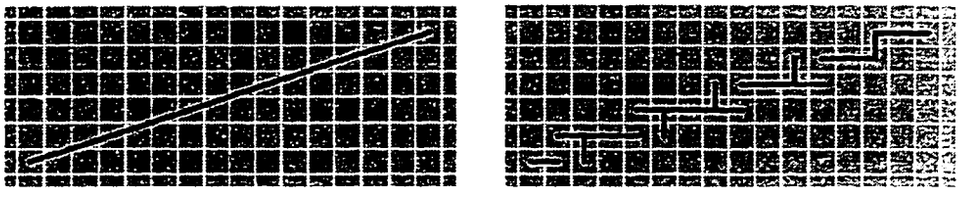


Figure 6.2: A simple discretization algorithm fails when the boundary intersect with the TLM nodes because the algorithm has no provision to decide where to place the $1\Delta l$ boundaries.

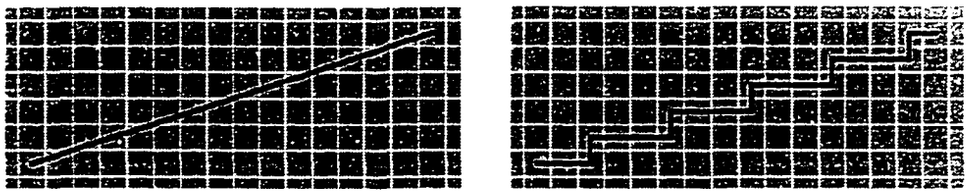


Figure 6.3: The boundary in Figure 6.2 is discretized with the same discretization algorithm but the origin of the mesh is shifted from $(0, 0)$ to $(0.0013\Delta l, 0.0031\Delta l)$. The shifting of the origin causes the nodes to be shifted away form the boundary so that the discretization algorithm can work properly.

This problem causing situation can be circumvented by simply adding a small perturbation factor in the discretization algorithm; for example, by shifting the origin of the mesh from $(0, 0)$ to (δ_x, δ_y). The staircase approximation in Figure 6.3 is generated with $\delta_x = 0.0013\Delta l$ and $\delta_y = 0.0031\Delta l$; these values are chosen with the following guidelines:

- The values must be much smaller than Δl, so that the only effect they have on the structure is to displace the nodes from the boundaries.
- The values must not be divisible by Δl, i.e. Δl must not be an integer multiple of them.
- The values should not be equal to nor an integer multiple of each other to avoid nulling out their effect. For instance, if they are equal, and the boundary is a 45° electric wall intersecting a number of nodes, then by shifting the electric wall with the same magnitude in both directions would not eliminate all the intersections.

Once a reliable line discretization algorithm has been implemented, more complicated structures can be discretized by using small piecewise linear segments. Figure 6.4 depicts a triangle and an ellipse discretized in such a manner.

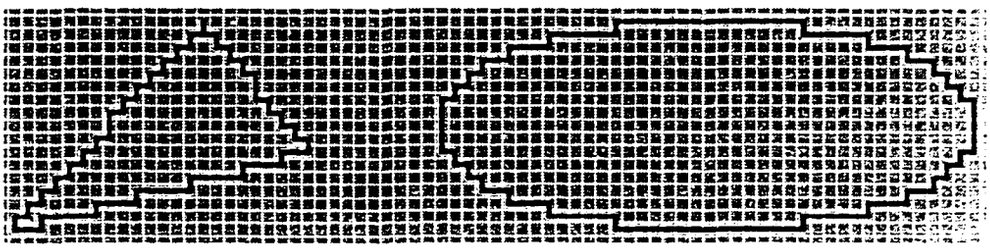


Figure 6.4: Discretized structures generated by using small linear segments and the algorithm described in Figure 6.3.

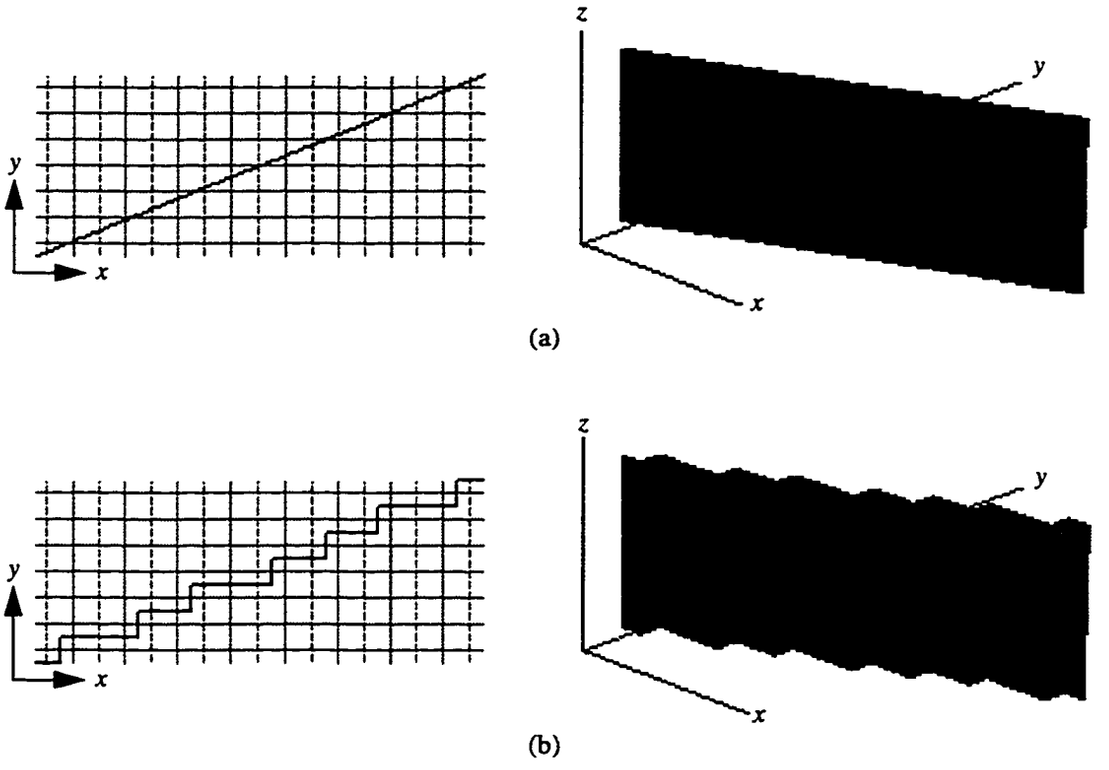


Figure 6.5: A plane in three-dimensional space that can be represented by a line in the two-dimensional space (xy-plane) and a depth in the z-direction can be discretized with the two-dimensional discretization algorithm.
(a) The original planar object.
(b) The discretized planar object.

The above discretization algorithm can be extended to discretize three-dimensional structures that can be represented by two-dimensional objects with a *depth* — from now on, such objects will be is called restricted 3D objects. For example, the rectangular planar object in Figure 6.5 is such a structure; it can be represented by a line in the xy-plane and a depth in the z-direction. In this case, the two-dimensional discretization algorithm is used to discretize the line in the xy-plane, and the discretized line is extended over the whole range of z-coordinates covered by the depth of the structure. This three-dimensional discretization procedure is given in the C++ format in *Appendix* A.1.

Curved boundaries in three-dimensional space can be discretized by using small piecewise rectangular segments with this algorithm as long as these piecewise segments are restricted 3D objects. Figure 6.6 depicts a small piece of coaxial line discretized by this method.

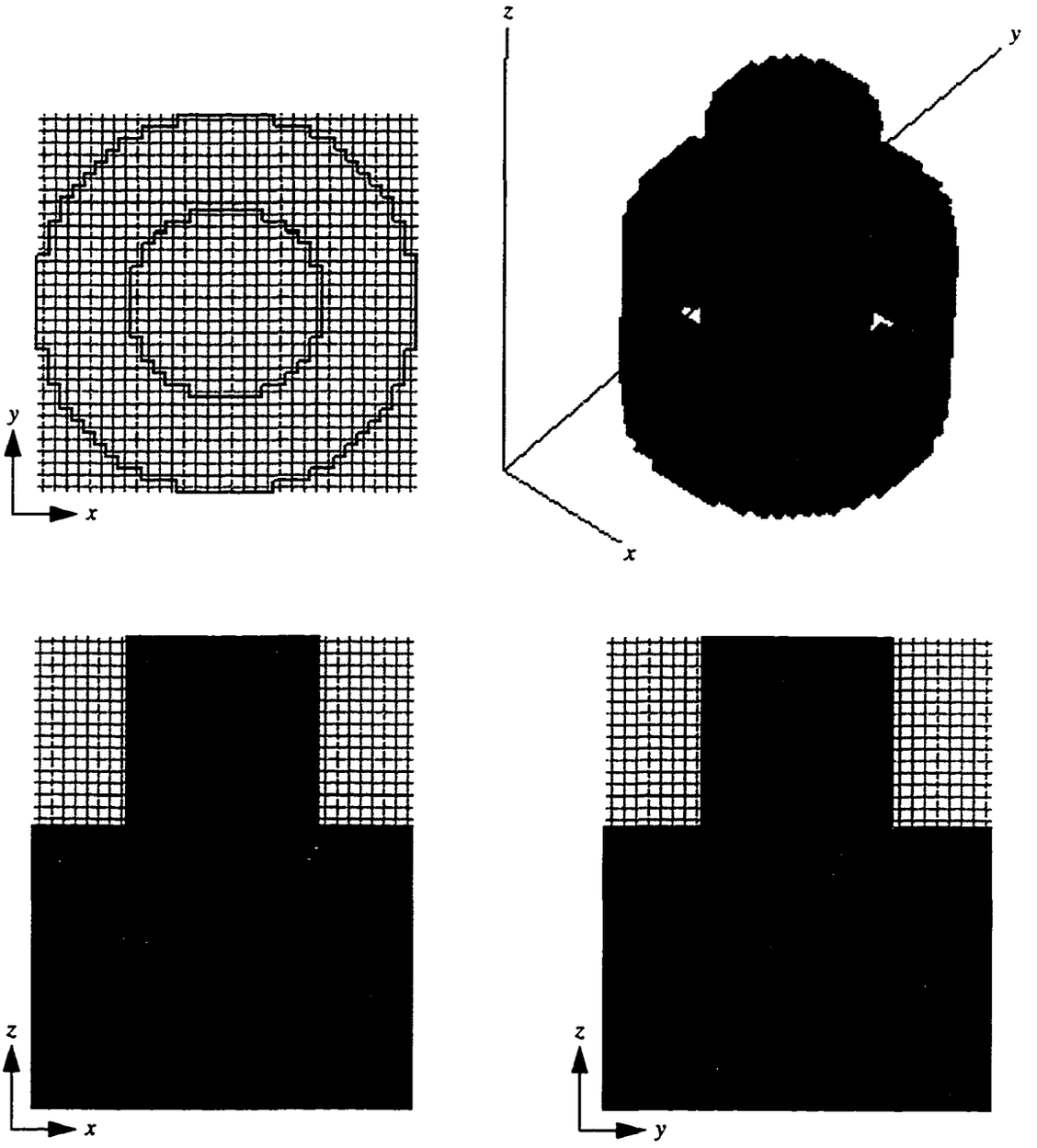


Figure 6.6: A small piece of coaxial line discretized by the discretization algorithm given in *Section* 6.2.

6.3 Adaptive Cell

An adaptive cell is a square TLM cell with a regular impulse scattering matrix and a regular transit time, but its link line impedances are modified in such a way that it simulates the characteristics of a fraction of a regular square cell — a fractional cell.

To connect an adaptive cell to a regular cell, a scattering procedure must be introduced at the interface. The same applies to the connection between two different adaptive cells. The interface scattering procedure will depend on whether the adaptive cell will be terminated by an electric or a magnetic wall. The theory and properties of adaptive cells are given in the following sections.

6.3.1 Fractional Cell Terminated by an Electric Wall

Figure 6.7a shows a parallel plate waveguide consisting of a chain of regular square 2D-TLM shunt cells terminated by an electric wall at a distance δ from the last regular cell; δ is a fraction of the cell size Δl. Figure 6.7b shows an electrically identical situation in which the fractional cell has been replaced by an adaptive cell of regular size, $\Delta l \times \Delta l$. The properties of the adaptive cell must be such that the impedance seen when looking to the right at the interface is the same in both cases.

The wave impedance of the fractional cell presented at the interface is:

$$Z_{ir} = jZ_r \tan(\beta_r \delta) \approx jZ_r \beta_r \delta \qquad \text{Eq 6.1}$$

The wave input impedance of the adaptive cell at the interface is:

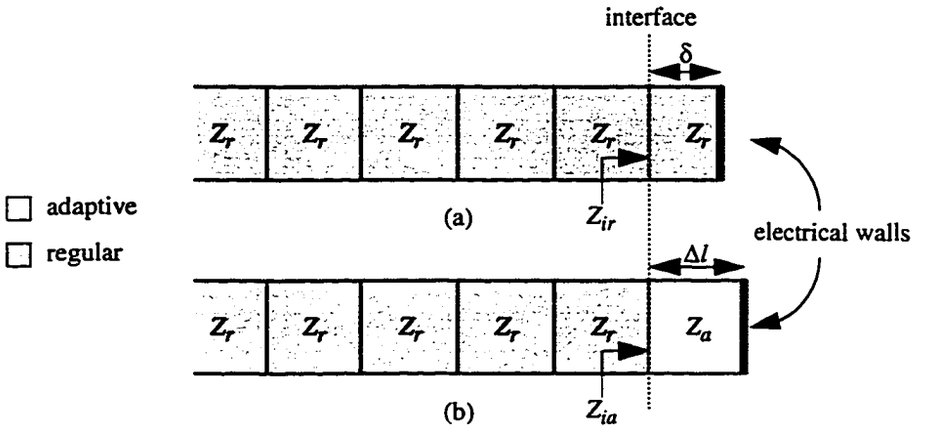


Figure 6.7: The equivalence between a fraction of the regular TLM cell (fractional cell) and the adaptive cell; both are terminated by electric walls.

$$Z_{ia} = jZ_a \tan(\beta_a \Delta l) \approx jZ_a \beta_a \Delta l \qquad \text{Eq 6.2}$$

For equivalence of the two cases, both Z_{ir} and Z_{ia} must be identical. In terms of the characteristics of the cells shown in Figure 6.8, the quantities in Eq 6.1 and Eq 6.2 are:

Characteristic impedance of the regular cell	$Z_r = \sqrt{L_o/(2C_o)}$	
Propagation constant of the regular cell	$\beta_r = \omega\sqrt{2L_oC_o}$	
Characteristic impedance of the adaptive cell	$Z_a = \sqrt{L_x/(C_x+C_y)}$	
Propagation constant of the adaptive cell	$\beta_a = \omega\sqrt{L_x(C_x+C_y)}$	Eq 6.3

L_o, C_o, L_x, C_x, L_y and C_y are the inductances and capacitances per unit length of the link lines. Introducing Eq 6.3 into Eq 6.1 and Eq 6.2 and equating the two impedances yields:

$$\sqrt{\frac{L_o}{2C_o}} \cdot \omega\sqrt{2L_oC_o} \cdot \delta = \sqrt{\frac{L_x}{C_x+C_y}} \cdot \omega\sqrt{L_x(C_x+C_y)} \cdot \Delta l \qquad \text{Eq 6.4}$$

which can be simplified to give:

$$L_x = L_o \frac{\delta}{\Delta l} \qquad \text{Eq 6.5}$$

Conservation of the scattering matrix and of the transit time of the adaptive cell requires that the link line impedances and the link line speeds must be the same in x and y-directions, i.e.:

$$\sqrt{L_x/C_x} = \sqrt{L_y/C_y} \quad \text{and} \quad \sqrt{L_xC_x} = \sqrt{L_yC_y} \qquad \text{Eq 6.6}$$

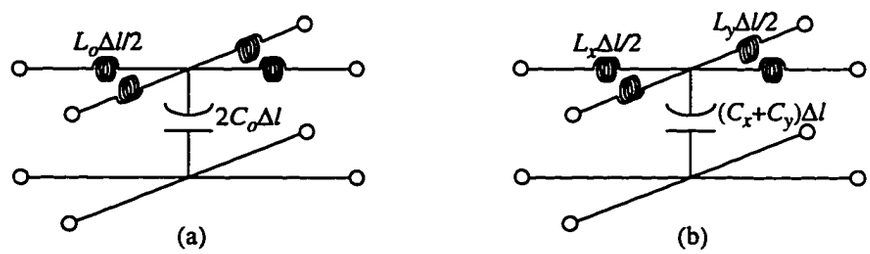


Figure 6.8: The equivalence of the regular cell and the adaptive cell.
(a) A regular cell with shunt node of two link lines of $Z_o = \sqrt{L_o/C_o}$.
(b) An adaptive cell with shunt node of two link lines of $Z_x = \sqrt{L_x/C_x}$ and $Z_y = \sqrt{L_y/C_y}$.

which requires

$$L_x = L_y \qquad \text{and} \qquad C_x = C_y \qquad \text{Eq 6.7}$$

Finally, since the transit time for the adaptive cell must be the same as that of a regular cell, i.e.:

$$L_o C_o = L_x C_x \qquad \text{Eq 6.8}$$

Combining Eq 6.5 and Eq 6.8 yields:

$$C_x = C_o \frac{L_o}{L_x} = C_o \frac{\Delta l}{\delta} = C_y \qquad \text{Eq 6.9}$$

Thus, the properties of the short-circuit terminated adaptive cell for any deformation ratio $\delta/\Delta l$ is:

Link Line Impedances: $$Z_x = Z_y = \sqrt{\frac{L_x}{C_x}} = \sqrt{\frac{L_y}{C_y}} = Z_o \frac{\delta}{\Delta l}$$

Wave Impedance: $$Z_a = \sqrt{\frac{L_x}{2C_x}} = Z_r \frac{\delta}{\Delta l} \qquad \text{Eq 6.10}$$

Connection of a regular cell to an adaptive cell requires the introduction of the following scattering procedure at the interface:

$$\begin{bmatrix} B_i \\ B_{i+1} \end{bmatrix} = \begin{bmatrix} \frac{n-1}{n+1} & \frac{2}{n+1} \\ \frac{2n}{n+1} & -\frac{n-1}{n+1} \end{bmatrix} \cdot \begin{bmatrix} A_i \\ A_{i+1} \end{bmatrix} \qquad \text{Eq 6.11}$$

where $n = \delta/\Delta l$ can be smaller or larger than 1, and the definition of A and B are given in Figure 6.9.

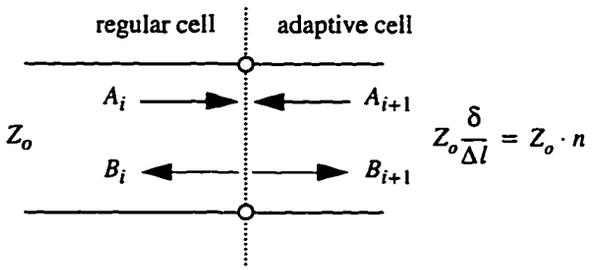


Figure 6.9: Connection of a regular cell to an adaptive cell.

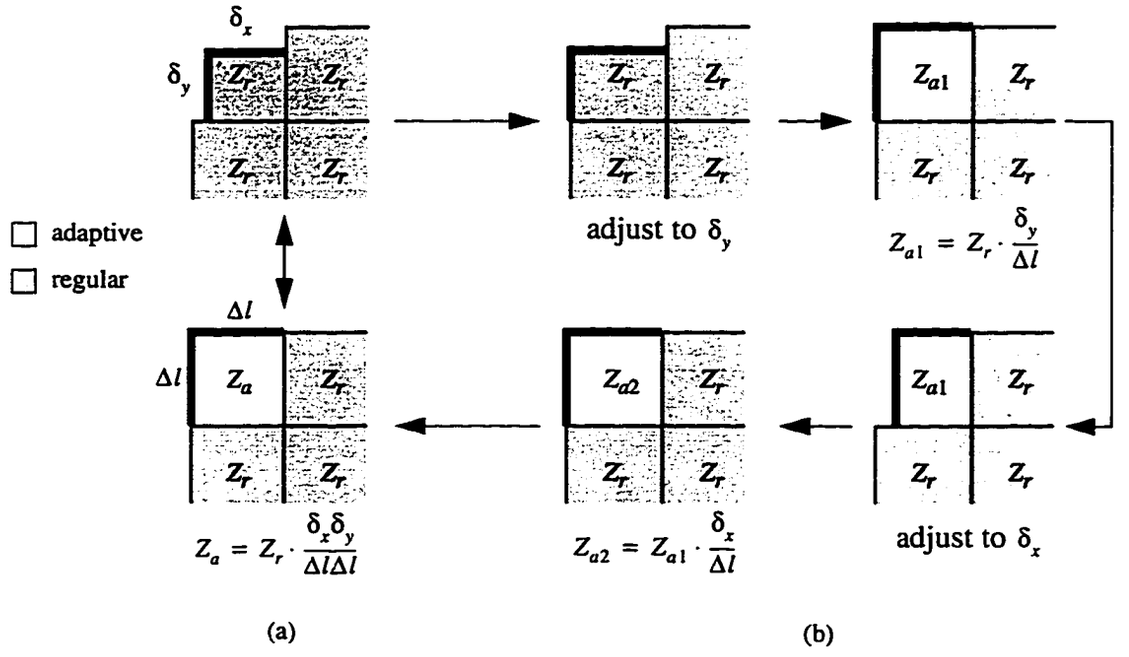


Figure 6.10: Modelling of a fractional cell bounded by two electric walls by means of an adaptive cell.
(a) A fractional cell terminated by two electric walls can be represented by an adaptive cell with impedance $Z_a = Z_r \delta_x \delta_y / (\Delta l \Delta l)$.
(b) The equivalence between Z_r and Z_a is derived by two successive applications of the adaptive cell theory, first in the y-direction, then in the

6.3.2 Fractional Cell Terminated by Two Electric Walls

The adaptive cell theory derived in *Section* 6.3.1 can be applied to the adaptive cells themselves; this allows a fractional cell terminated on two adjacent sides by electric walls to be modelled by on adaptive cell; Figure 6.10a shows the equivalence between a fractional cell bounded by two electric walls at some arbitrary positions, and an adaptive cell bounded by two electric walls at the regular positions. Figure 6.10b shows the successive applications of the adaptive cell theory, first by assuming that the fractional cell has been shortened only along one direction, and then by shortening the resulting adaptive cell again in the other direction; and the result is:

$$Z_a = Z_r \cdot \frac{\delta_x \delta_y}{\Delta l \Delta l} \qquad \text{Eq 6.12}$$

That means, Eq 6.11 must be applied to the interfaces between domains with Z_r and Z_a. In this situation, $n = \delta_x \delta_y / (\Delta l \Delta l)$.

6.3.3 Fractional Cell Terminated by Magnetic Walls

The derivations for the adaptive cell terminated by the magnetic walls are similar to that terminated by the electric walls. Replacing in Figure 6.7a and b the electric wall by a magnetic wall leads to the following requirement for equivalence:

$$jY_r\beta_r\delta = jY_a\beta_a\Delta l \qquad \text{Eq 6.13}$$

or, in detail:

$$\sqrt{\frac{2C_o}{L_o}}\cdot\omega\cdot\sqrt{2C_oL_o}\cdot\delta = \sqrt{\frac{C_x+C_y}{L_x}}\cdot\omega\cdot\sqrt{L_x(C_x+C_y)}\cdot\Delta l \qquad \text{Eq 6.14}$$

which amounts to

$$2C_o\delta = (C_x+C_y)\Delta l \qquad \text{Eq 6.15}$$

Substituting Eq 6.7 and Eq 6.8 into the above equations yields:

$$C_x = C_y = C_o\frac{\delta}{\Delta l} \qquad \text{and} \qquad L_x = L_y = L_o\frac{\Delta l}{\delta} \qquad \text{Eq 6.16}$$

The properties of the adaptive cell are thus in the open-circuited case:

Link Line Impedance: $$Z_x = Z_y = \sqrt{\frac{L_x}{C_x}} = \sqrt{\frac{L_y}{C_y}} = Z_o\cdot\frac{\Delta l}{\delta}$$

Wave Impedance: $$Z_a = \sqrt{\frac{L_x}{2C_x}} = Z_r\frac{\Delta l}{\delta} \qquad \text{Eq 6.17}$$

For the scattering at the interface between the regular cell and the adaptive cell, n in Eq 6.11 must be replaced by $1/n$, which gives:

$$\begin{bmatrix} B_i \\ B_{i+1} \end{bmatrix} = \begin{bmatrix} \frac{1-n}{1+n} & \frac{2n}{1+n} \\ \frac{2}{1+n} & -\frac{1-n}{1+n} \end{bmatrix} \cdot \begin{bmatrix} A_i \\ A_{i+1} \end{bmatrix} \qquad \text{where} \qquad n = \frac{\delta}{\Delta l} \qquad \text{Eq 6.18}$$

Similarly, replacing impedance with admittance in Eq 6.12 yields the admittance ratio between the adaptive cell terminated by two adjacent magnetic walls:

$$Y_a = Y_r\cdot\frac{\delta_x\delta_y}{\Delta l\Delta l} \qquad \text{Eq 6.19}$$

That means, Eq 6.18 must be applied to the interface between Y_r and Y_a. In this situation, $n = \delta_x\delta_y/(\Delta l\Delta l)$.

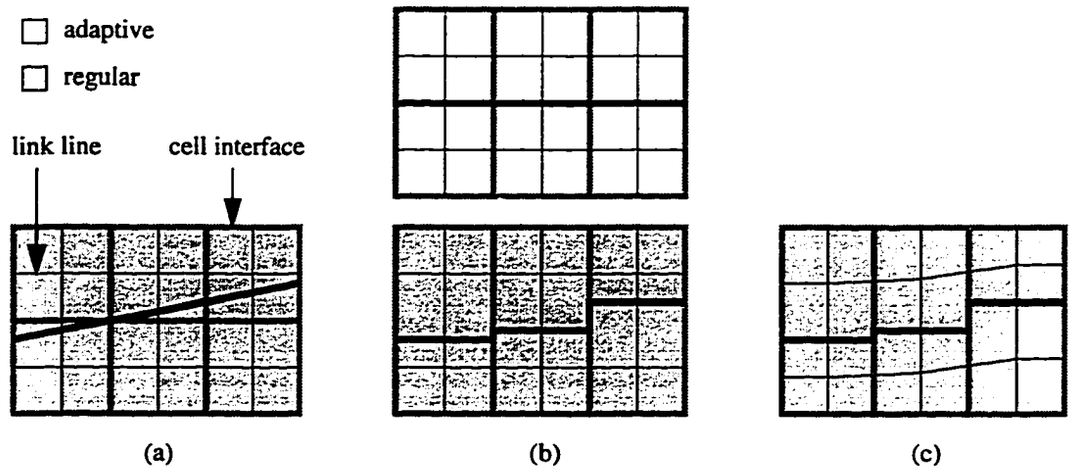


Figure 6.11: Replacing regular cells in a TLM mesh with adaptive cells of the appropriate impedance values produces an adaptive mesh.
(a) A regular TLM mesh with an inclined boundary.
(b) Ideally, the inclined boundary should be represented by regular cells of the appropriate dimensions (bottom). However, time synchronism would not be preserved in this case. Therefore, adaptive cells having the appropriate wave impedance values (top) are used to realize the equivalent response.
(c) Since all the link lines of an adaptive cell have the same impedance and the equivalent response is realized via some interface scattering operations, the field values at the center of the adaptive cells are those that would exist at the center of the corresponding fractional cells.

6.4 Adaptive Mesh

Whenever a boundary is not located exactly halfway between two nodes, adaptive cells can be used to replace the fractional regular cells that are affected by the boundary, Figure 6.11a depicts such a situation. Replacing the regular cells affected by the boundary in a TLM mesh with the adaptive cells of the appropriate impedance values produces a TLM mesh with cells having the centers electrically displaced from their original positions (Figure 6.11c); this yield an adaptive mesh. The example in Figure 6.11 contains adaptive cells that represent fractional cells in one direction only. In fact, a similar adjustment can be done in the other direction as well.

The interface between two adaptive cells must be handled properly as well because their impedance values may not be the same. Consider the situation shown in Figure 6.12 where two adaptive cells with two different compression ratios are connected to each other. As mentioned in *Section* 6.3.1, the link lines in the adaptive cells with short-circuited termination have the following properties:

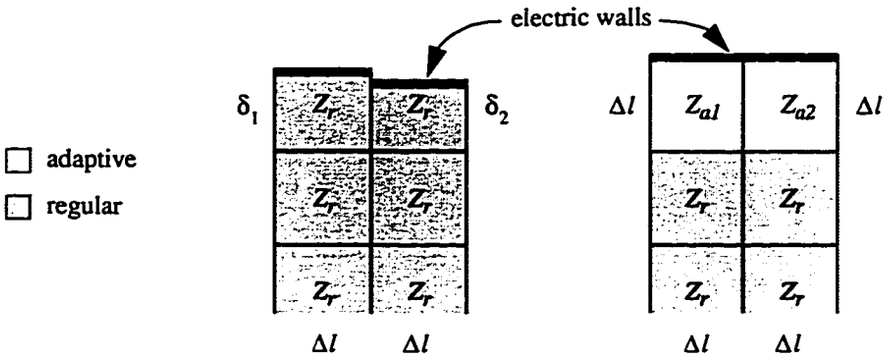


Figure 6.12: Connection of two adaptive cells for different compression ratios.

Adaptive Cell 1 $$Z_{x1} = Z_{y1} = Z_o \frac{\delta_1}{\Delta l}$$

Adaptive Cell 2 $$Z_{x2} = Z_{y2} = Z_o \frac{\delta_2}{\Delta l}$$ Eq 6.20

Therefore, one must implemented a scattering procedure at the interface between the two adaptive cells as follows, Figure 6.13:

$$\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = \begin{bmatrix} \frac{n-1}{n+1} & \frac{2}{n+1} \\ \frac{2n}{n+1} & -\frac{n-1}{n+1} \end{bmatrix} \cdot \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \quad \text{where} \quad n = \delta_2/\delta_1 \qquad \text{Eq 6.21}$$

In general, when the adjacent adaptive cells are associated with different compression ratios in both directions, then:

$$\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = \begin{bmatrix} \frac{n-1}{n+1} & \frac{2}{n+1} \\ \frac{2n}{n+1} & -\frac{n-1}{n+1} \end{bmatrix} \cdot \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \quad \text{where} \quad n = (\delta_{2x}\delta_{2y})/(\delta_{1x}\delta_{1y}) \qquad \text{Eq 6.22}$$

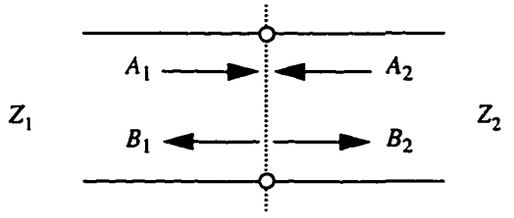


Figure 6.13: Connection between two adaptive cells for different compression ratios.

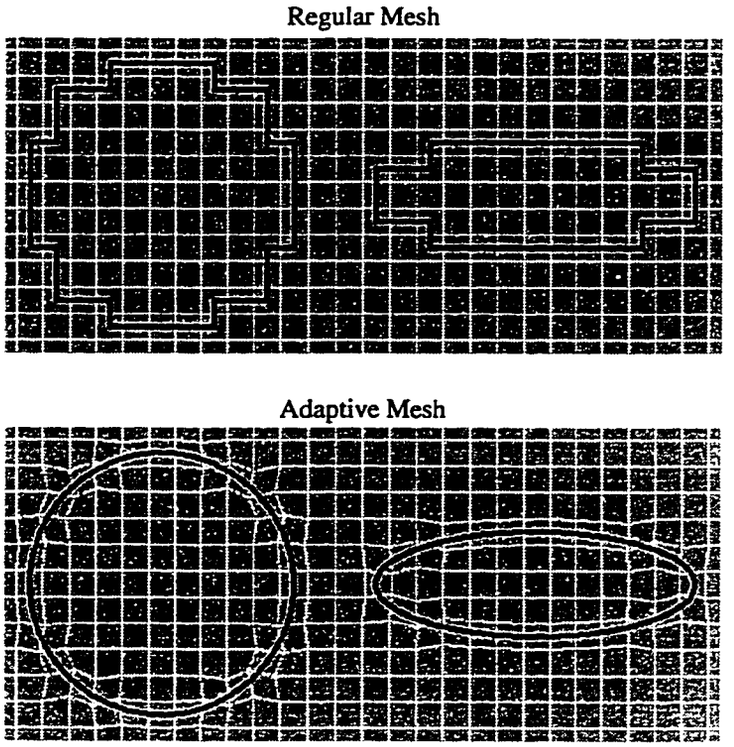


Figure 6.14: A circular and an elliptical waveguide discretized by using the regular and the adaptive mesh. It is obvious that the structures discretized with the adaptive mesh resemble more the actual structures. The adaptive mesh solution is easier to implement than a circular mesh and an elliptical mesh interfacing with a regular square mesh.

6.5 Automatic Discretization of Structures using an Adaptive Mesh

The automatic discretization algorithm discussed in *Section* 6.2 and the theory of the adaptive mesh in *Section* 6.4 can be used separately or together to discretize complicated structures for TLM modelling. In fact, they have been integrated together to provide an innovative meshing feature for TLM — automatic discretization of structures using an adaptive mesh for TLM.

The versatility of this new discretization method is demonstrated in Figure 6.14. A circular waveguide and an elliptical waveguide are positioned side-by-side in one mesh. It is obvious that the adaptive mesh discretization gives a better representation of the original structures than a staircase model. The adaptive mesh solution is easier to implement than using a locally circular mesh and a locally elliptical mesh because no interfacing between different types of meshes is needed.

6.6 Verification via Simulation

The theories of the adaptive cell and the adaptive mesh are verified via a number of simulation examples in the following sections. The results show that the adaptive cell and the adaptive mesh models are accurate at least to the first order of Δl.

6.6.1 Validation of the Theory of the Adaptive Cell

The theory of the adaptive cell is validated in this section through a WR28 waveguide simulation example. The waveguide width is increased by 0.25, 0.50 and $0.75\Delta l$; i.e. a = 7.2009, 7.2898 and 7.3787 mm, respectively, for Δl = 0.3556 mm. Figure 6.15 depicts the waveguide with its width increased by $0.25\Delta l$. The figure shows that a TLM mesh consisting only of the regular cells with Δl = 0.3556 mm cannot represent the waveguide properly. However, the same TLM mesh with the last column of cells replaced by adaptive cells can represent the waveguide precisely. The cutoff frequencies of a number of such waveguides computed with and without using the adaptive cells at the last column are shown in Table 6.1; the data in the table indicate that adaptive cell technique gives more accurate results in all the cases except TE_{01}, which is not unreasonable because this mode does not depends on the width of the waveguide.

	TE_{10}	TE_{20}	TE_{01}	TE_{11}	TE_{21}	TE_{30}	TM_{11}	TM_{21}
Theoretical	20.82	41.63	42.15	47.01	59.25	62.45	47.01	59.25
Regular	21.07	42.07	42.07	47.09	59.64	62.93	47.10	59.61
Difference |%|	1.201	1.057	0.1898	0.1702	0.6582	0.7686	0.1914	0.6076
Adaptive	20.80	41.49	42.26	46.93	59.19	62.31	46.98	59.28
Difference |%|	0.09606	0.3363	0.2610	0.1702	0.1013	0.2242	0.06382	0.05063
Theoretical	20.56	41.12	42.15	46.90	58.89	61.69	46.90	58.89
Regular	20.06	40.06	42.10	46.63	58.16	59.98	46.65	58.21
Difference |%|	2.432	2.578	0.1186	0.5757	1.240	2.772	0.5330	1.155
Adaptive	20.55	40.97	42.17	46.80	58.84	61.37	46.87	58.87
Difference |%|	0.04863	0.3648	0.04745	0.2132	0.08490	0.5187	0.06397	0.03396
Theoretical	20.31	40.63	42.15	46.79	58.55	60.94	46.79	58.55
Regular	20.06	40.06	42.10	46.63	58.16	59.98	46.65	58.21
Difference |%|	1.231	1.403	0.1186	0.2780	0.6661	1.575	0.2992	0.5807
Adaptive	20.29	40.54	42.09	46.73	58.56	60.59	46.76	58.53
Difference |%|	0.09847	0.2215	0.1423	0.1282	0.01708	0.5743	0.06412	0.03416

Table 6.1: The cutoff frequencies (in GHz) of the waveguides shown in Figure 6.15. The table shows three groups of results from top to bottom which correspond to a = 7.2009, 7.2898 and 7.3787 mm, respectively. The size of the regular TLM cell is Δl = 7.112 mm.

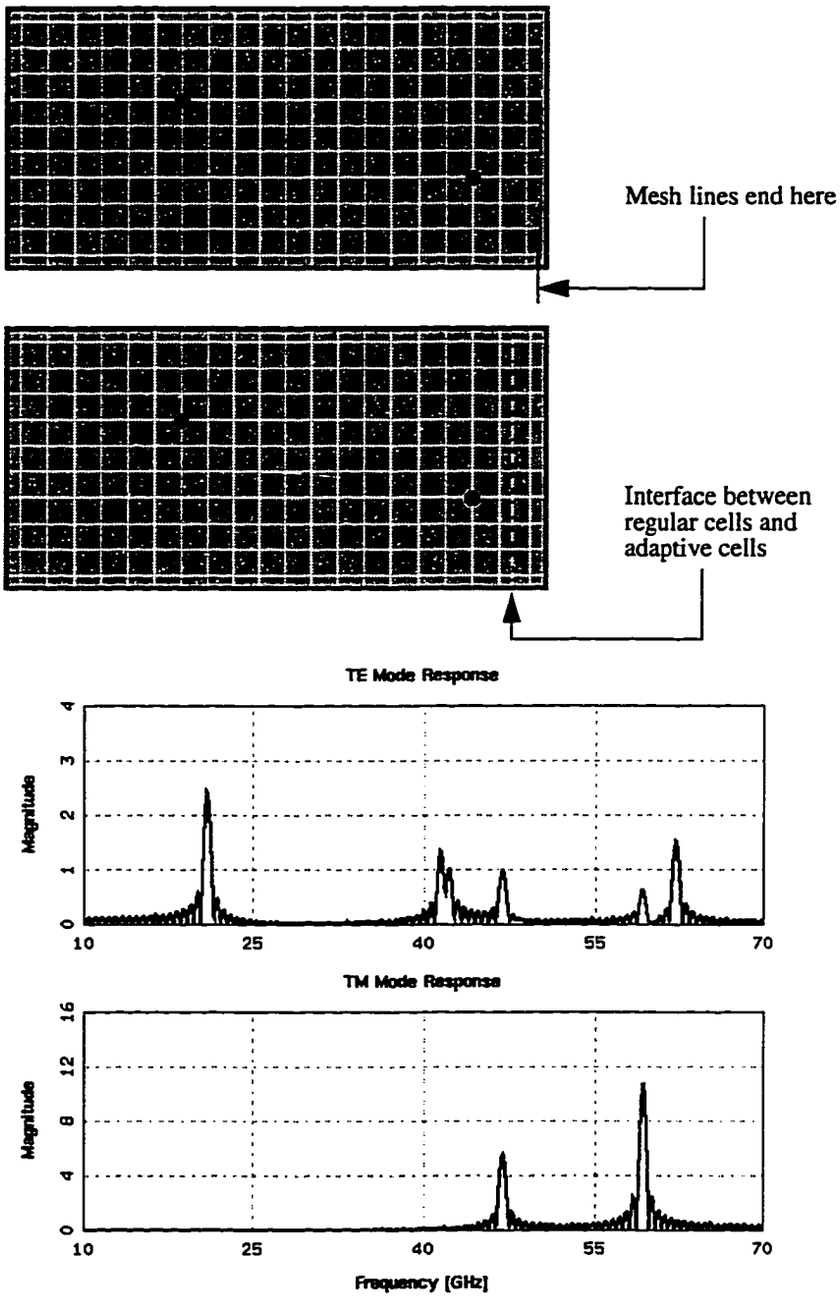


Figure 6.15: A WR28 waveguide with its width increased by $0.25\Delta l$. The top image shows that the TLM mesh consisting only of regular TLM cells with $\Delta l = 0.3556$ mm cannot represent the waveguide properly. However, the same TLM mesh with the last column of cells replaced by adaptive cells can represent the waveguide precisely. The frequency response shown above is obtained with the adaptive cell technique. The cutoff frequencies of a number of these waveguides computed with and without using the adaptive cells are shown in Table 6.1.

6.6.2 Validation of the Theory of the Adaptive Mesh

The theory of the adaptive mesh is validated in this section by two simulation examples. The first one is a WR28 waveguide rotated 45° with respect to the x-axis. The second one is a circular waveguide discretized with various mesh parameters.

6.6.2.1 A Rotated WR28 Waveguide

This example demonstrate the versatility of the adaptive mesh when modelling a rotated WR28 waveguide. Figure 6.16 depicts the discretized images of the waveguide together with their corresponding responses. It is easily noticeable that the two sets of responses are quite different. In fact, the one that is computed without the adaptive mesh feature gives incorrect cutoff frequencies. The cutoff frequencies obtained from these responses are given in Table 6.2.

The accuracy of the result can be improved by reducing Δl. Figure 6.17 depicts another set of simulation curves computed with Δl = 0.25 mm. The cutoff frequencies obtained form these curves are also given in Table 6.2.

The date in Table 6.2 indicated that the adaptive mesh gives better results than the regular mesh in most of the cases except for TE_{10} mode. However, the errors are decreasing with Δl.

	TE_{10}	$TE_{20/01}$	$TE_{20/01}$	TE_{11}	TE_{21}	TE_{30}	TM_{11}	TM_{21}
Theoretical	21.08	42.15	42.15	47.13	59.61	63.23	47.13	59.61
Regular	20.19	37.37	40.15	41.01	52.39	59.83	43.89	56.95
Difference |%|	4.222	11.34	4.745	12.99	12.11	5.377	6.875	4.462
Adaptive	20.00	40.21	41.54	45.55	56.50	60.66	48.72	60.32
Difference |%|	5.123	4.603	1.447	3.352	5.217	4.065	3.374	1.191
Regular	20.68	39.83	41.32	44.14	56.19	61.90	45.57	58.45
Difference |%|	1.898	5.504	1.969	6.344	5.737	2.103	3.310	1.946
Adaptive	20.50	41.02	42.18	46.15	57.88	61.57	47.94	59.94
Difference |%|	2.751	2.681	0.07117	2.079	2.902	2.625	1.719	0.5536

Table 6.2: The cutoff frequencies obtained from the responses in Figure 6.16 (top) and Figure 6.17 (bottom), which corresponding to Δl=0.5 and 0.25 mm, respectively.

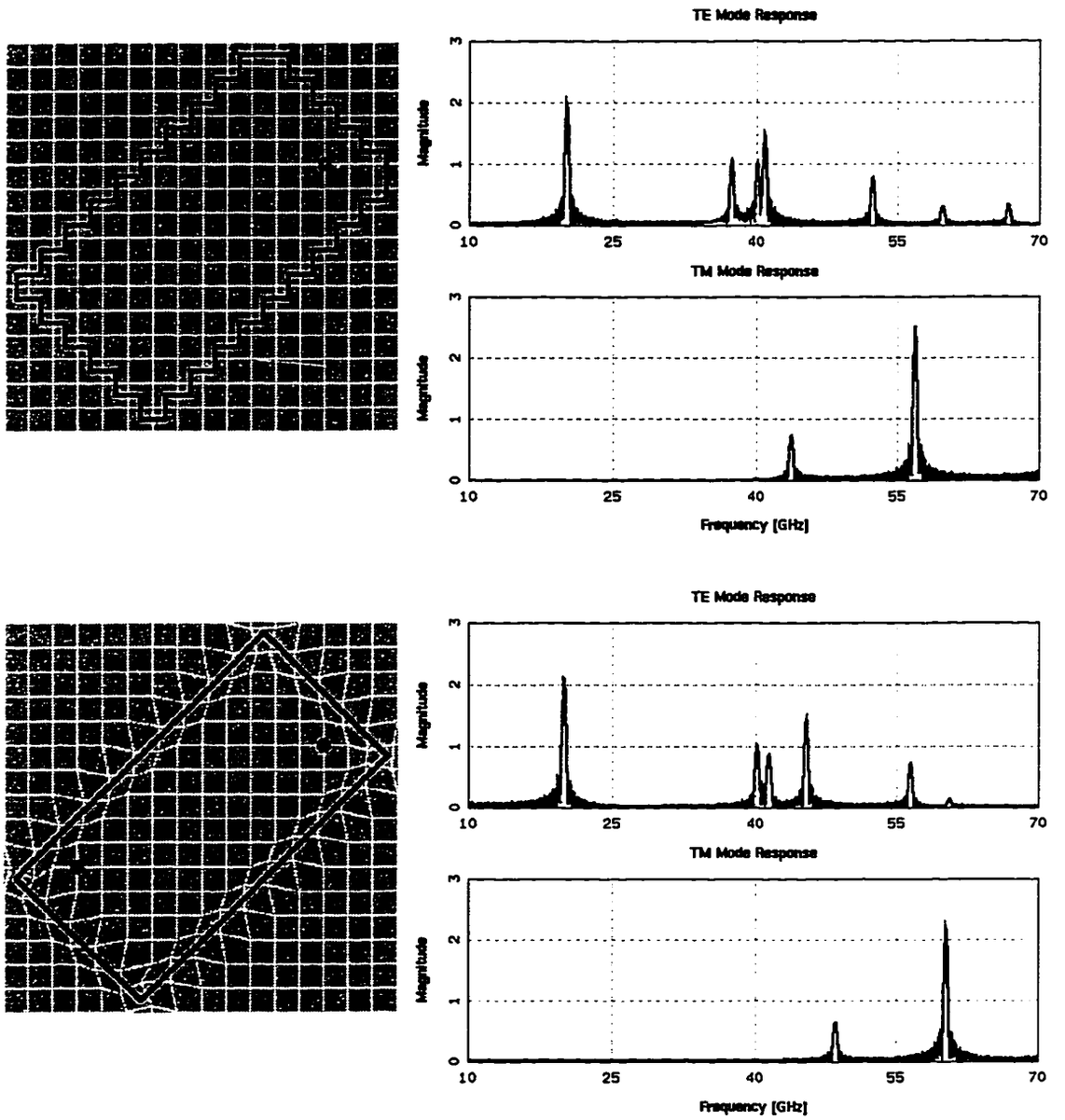


Figure 6.16: A WR28 waveguide rotated 45° with respected to the x-axis. The first set of responses is computed with the stair-cased regular mesh, the second set is computed with the adaptive mesh. There is a noticeable shift in the cutoff frequencies between the two sets of responses; in fact, the responses that are computed with the regular mesh give incorrect cutoff frequencies. The cutoff frequencies extracted from the above responses are given in Table 6.2.

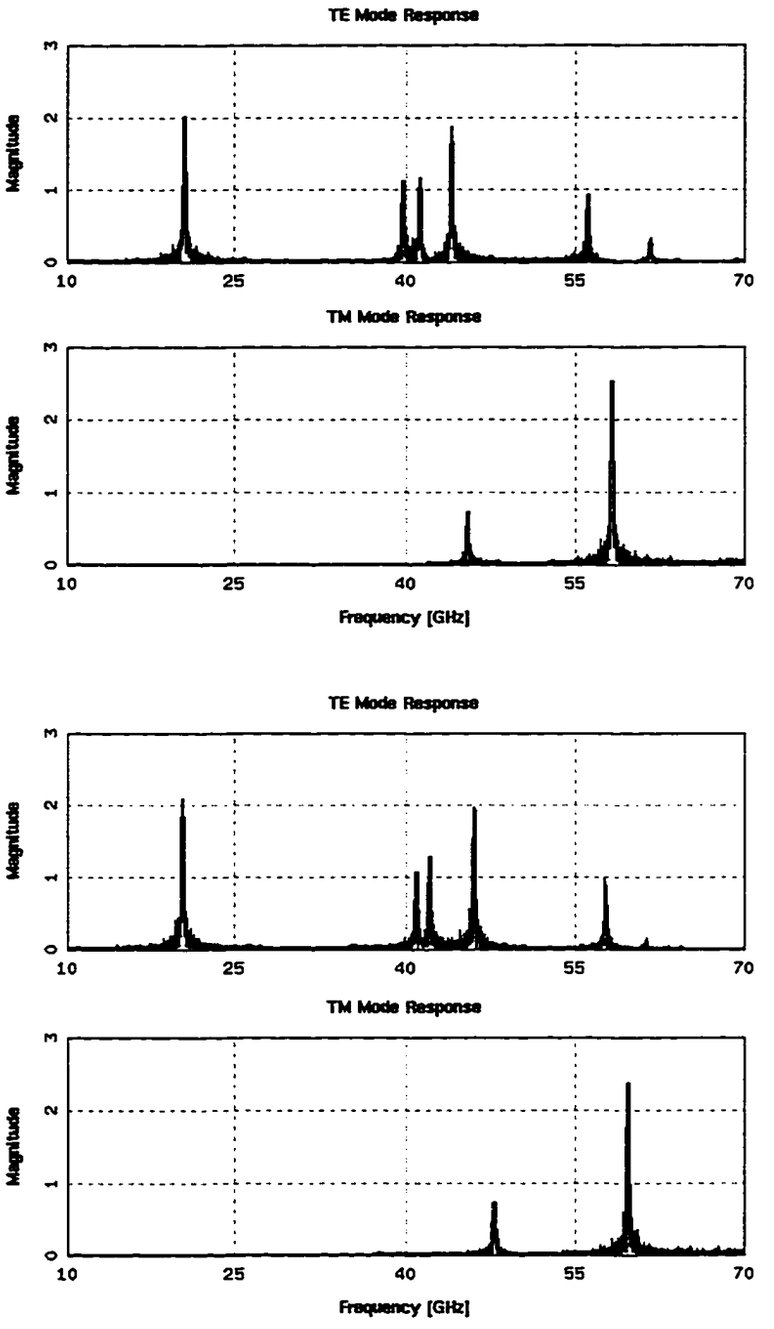


Figure 6.17: Frequency responses of the WR28 waveguide depicted in Figure 6.16. The responses are computed using $\Delta l = 0.25$ mm. The top two curves are computed with stair-cased regular mesh; the bottom two are obtained with the adaptive mesh.

6.6.2.2 A Circular Waveguide

Figure 6.18 depicts the TM and TE responses of a circular waveguide (23 cm diameter) under impulsive excitations. Meshes of 1.0 and 0.5 cm discretization resolution were used to compute its cutoff frequencies. Table 6.3 shows the agreement between the simulation results obtained with a staircase approximation, adaptive cell method, and the theoretical values [52].

The data in Table 6.3 show that all the TE mode results obtained with the adaptive mesh are better than those obtained with the regular mesh. For the TM cases, the adaptive mesh do not always give better results than the regular mesh, however, the errors is in the same order of magnitude as that of the regular mesh. More testing is needed to determine whether the error are introduced by the specific TM circular waveguide modes.

		TE_{11}	TE_{21}	TE_{01}	TE_{31}	TM_{01}	TM_{11}
Theoretical [52]	GHz	0.7638	1.267	1.600	1.743	0.9978	1.600
Regular (Δl = 1 cm)	GHz	0.7508	1.224	1.577	1.682	0.9957	1.586
Difference	|%|	1.702	3.394	1.438	3.500	0.2105	0.8750
Adaptive (Δl = 1 cm)	GHz	0.7591	1.238	1.592	1.708	1.003	1.596
Difference	|%|	0.6153	2.289	0.500	2.008	0.5211	0.2500
Regular (Δl = 0.5 cm)	GHz	0.7595	1.248	1.588	1.719	1.000	1.592
Difference	|%|	0.5630	1.500	0.7500	1.377	0.2205	0.5000
Adaptive (Δl = 0.5 cm)	GHz	0.7598	1.251	1.588	1.721	0.9985	1.590
Difference	|%|	0.5240	0.2404	0.7500	1.262	0.07015	0.6250

Table 6.3: The cutoff frequencies of a circular waveguide (23 cm diameter) obtained via analytical expressions and TLM simulations. The regular TLM simulation results are obtained with a staircase approximation of the contour.

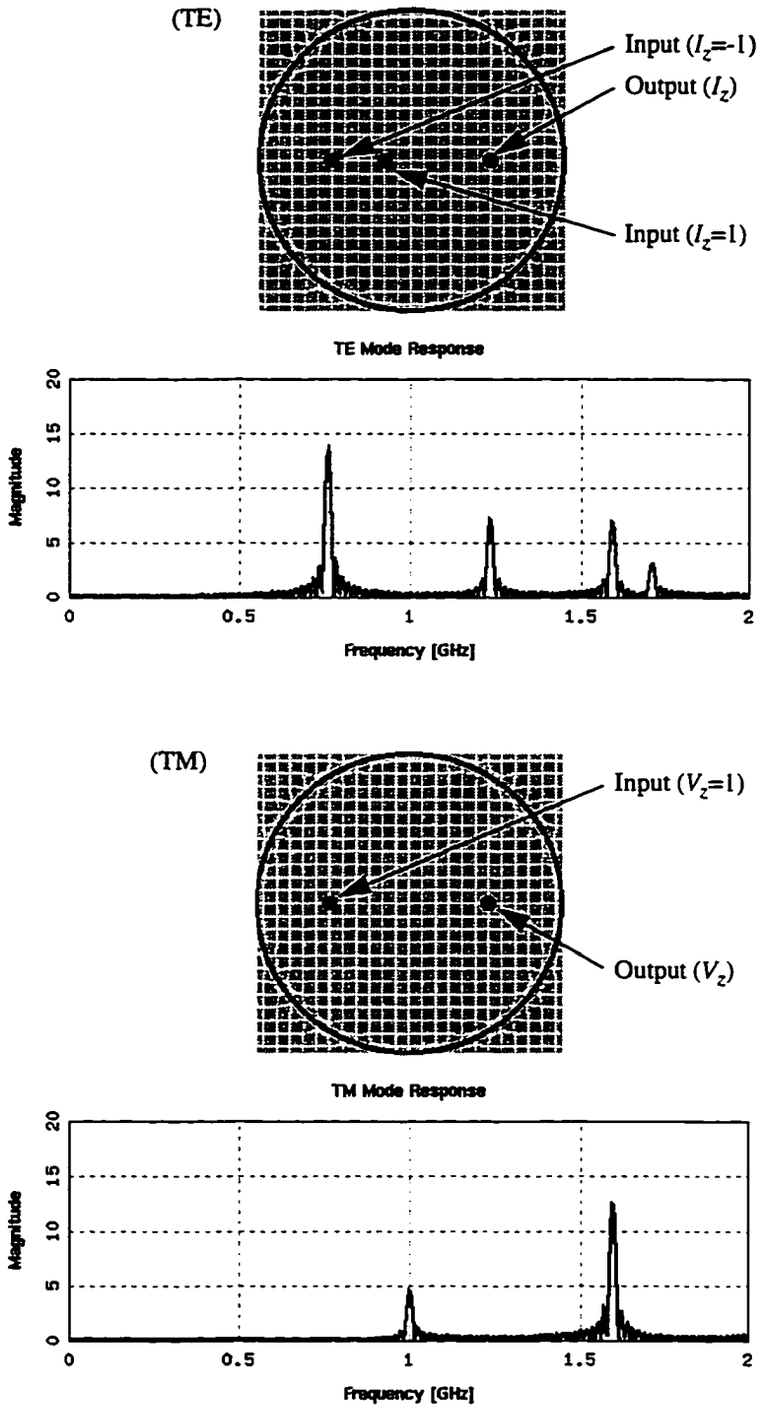


Figure 6.18: The TE and TM impulse responses of a circular waveguide (23 cm diameter). The cutoff frequencies of the waveguide are given in Table 6.3.

6.7 Extension of the Adaptive Theory to the 3D-TLM Symmetrical Condensed Node

The theory of adaptive cell and adaptive mesh can be extended easily to the 3D-TLM symmetrical condensed node. Since there are two polarizations in each propagation direction, the interface scattering procedures must be applied to both polarizations.

6.7.1 Adaptive Cell for 3D-TLM

Connection of a regular cell to an adaptive cell requires the introduction of Eq 6.11 and 6.12 to the voltages in both polarizations at the interface. When an adaptive cell is bounded by three electrical walls (for example, an adaptive cell at the corner of a rectangular resonator), n in Eq 6.11 is must be modified as follow:

$$n = \frac{\delta_x \delta_y \delta_z}{(\Delta l)^3} \qquad \text{Eq 6.23}$$

This is also applied to Eq 6.18 for the case of connection of a regular cell to an adaptive cell bounded by magnetic walls.

6.7.2 Adaptive Mesh for 3D-TLM

Connection of adaptive cells of different compression ratios requires the introduction of an interfacing scattering procedure to the voltages in both polarizations at the interface. Eq 6.21 and 6.22 are still valid, but n must be generalized to handle the case where the adaptive cells have been compressed in all three directions:

$$n = \frac{\text{volume}_1}{\text{volume}_2} \qquad \text{Eq 6.24}$$

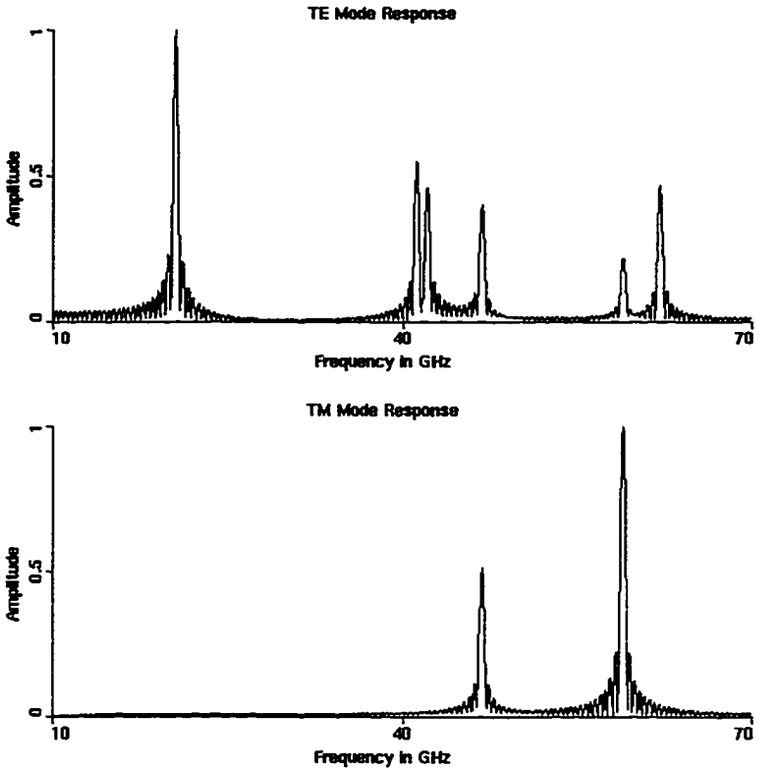


Figure 6.19: The frequency response of the a WR28 waveguide, its width is increased by $0.5\Delta l$, i.e. a = 7.2898 mm. The cutoff frequencies for a = 7.2009, 7.2898 and 7.3787 mm computed with and without using the adaptive cells are shown in Table 6.4.

6.7.3 Validation of the Theory of Adaptive Cell for 3D-TLM

The theory of adaptive cell for 3D-TLM is validated in this section using the same example that is used in *Section* 6.6.1 for 2D-TLM. The width of the waveguide is increased by 0.25, 0.50 and $0.75\Delta l$; i.e. a = 7.2009, 7.2898 and 7.3787 mm, respectively, for Δl = 0.3556 mm. The frequency response for a = 7.2898 mm and the cutoff frequencies at the three different waveguide widths are shown in Figure 6.19 and Table 6.4.

The data in Table 6.4 indicate that adaptive cells give more accurate results in all of the cases except TE_{01} mode. The reason is that this mode does not depends on the width of the waveguide, therefore the result for the regular mesh should be better. However, the result from the adaptive cell is still very small.

	TE_{10}	TE_{20}	TE_{01}	TE_{11}	TE_{21}	TE_{30}	TM_{11}	TM_{21}
Theoretical	20.82	41.63	42.15	47.01	59.25	62.45	47.01	59.25
Regular	21.07	42.16	42.16	47.08	59.44	63.23	47.08	59.49
Difference |%|	1.201	1.273	0.00237	0.1489	0.3207	1.249	0.1489	0.4051
Adaptive	20.82	41.64	42.21	46.97	59.14	62.60	46.98	59.16
Difference |%|	0.000	0.02402	0.1423	0.08509	0.1857	0.2402	0.06382	0.1519
Theoretical	20.56	41.12	42.15	46.90	58.89	61.69	46.90	58.89
Regular	20.07	40.13	42.17	46.64	58.10	60.22	46.66	58.10
Difference |%|	2.283	2.408	0.04745	0.5544	1.3415	2.3829	0.5117	1.341
Adaptive	20.57	41.21	42.20	46.86	58.84	62.03	46.87	58.85
Difference |%|	0.04864	0.2189	0.1186	0.08529	0.08490	0.5511	0.06397	0.06792
Theoretical	20.31	40.63	42.15	46.79	58.55	60.94	46.79	58.55
Regular	20.07	40.13	42.17	46.64	58.10	60.22	46.66	58.10
Difference |%|	1.182	1.231	0.04745	0.3206	0.7686	1.181	0.2778	0.7686
Adaptive	20.34	40.81	42.19	46.74	58.52	61.53	46.77	58.56
Difference |%|	0.1477	0.4430	0.09490	0.1069	0.05124	0.9682	0.04274	0.01708

Table 6.4: The cutoff frequencies (in GHz) of the waveguides shown in Figure 6.15. The table shows three groups of results from top to bottom which corresponds to a = 7.2009, 7.2898 and 7.3787 mm, respectively.

6.8 Conclusion

An automatic discretization algorithm and the theory of adaptive cell and adaptive mesh have been developed in this chapter. These features have been integrated together to provide an innovative algorithm for automatic discretization of structures with adaptive mesh, which provides an unprecedented modelling flexibility and capability for the TLM method.

A number of simulation examples are used to verify the correctness of the theory of adaptive cell and adaptive mesh as well as the automatic discretization algorithms. The simulation results show that automatic discretization algorithms are working as expected. They also indicate that, when the structure to be study involve only straight line boundaries, the adaptive mesh yields better results than the regular mesh.

CHAPTER 7 Special Computing Techniques

7.1 Introduction

In practical engineering applications, the material to be modelled are generally moderate to highly inhomogeneous, therefore, flexibility in modelling highly inhomogeneous material with TLM is an important issue. However, flexibility and efficiency are always two conflicting issues. Once an algorithm is design to be flexible, it becomes less efficient in terms of execution speed.

Speed of execution is of particularly important when the TLM field analysis is to be coupled with optimization, which may require up to several hundred analyses to be performed within a reasonable time. A good implement of the TLM algorithm which avoid unnecessary numerical operations will improve the run time of TLM field analysis. Parallel and distributed computing techniques can also be applied to reduce the run time. Signal processing on the other hand can be used the speed up a modelling analysis by eliminating unnecessary TLM computation.

This chapter continue with a discussion of a versatile implementing of the TLM method which will enable TLM to be used to model highly inhomogeneous material without too much scarify in efficient. It then move onto the computing techniques which can be used to either speed up execution speed or eliminate unnecessary TLM computation.

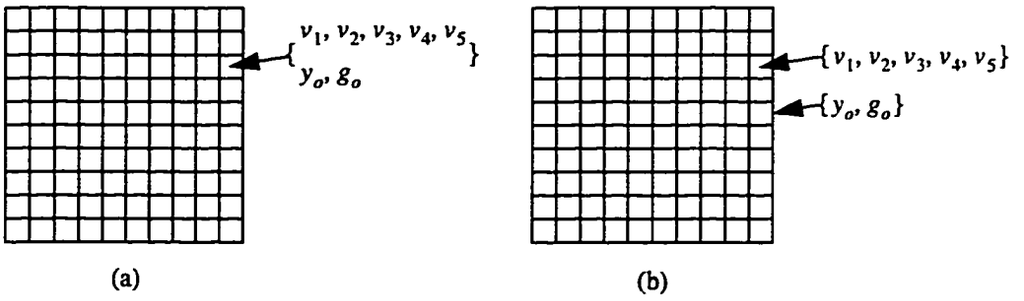


Figure 7.1: Implementation of the 2D-TLM shunt mesh via the array data structure.
(a) All entries of the array contain $v_1, v_2, v_3, v_4, v_5, y_o$ and g_o; this implementation is good for modelling highly inhomogeneous materials.
(b) All entries of the array contain v_1, v_2, v_3, v_4, v_5; y_o and g_o are common to all the nodes in the mesh; this implementation is good for modelling homogeneous materials.

7.2 Versatile Implementation of the TLM Method

Versatility and efficiency are usually two conflicting issues. In engineering applications, the electromagnetic structures to be designed are generally quite complex, therefore, flexibility is an important feature. However, the efficiency of the algorithm must not be ignored because it is directly related to the size of the problems that can be solved by the tool. Hence, a well balanced compromise is the best way to resolve the conflict.

7.2.1 Problems of Using Simple Array Data Structures

The most straightforward way to implement the TLM algorithm is via the simple array data structure. Figure 7.1 depicts two different ways of representing the 2D-TLM shunt mesh by array data structures. The representations are good for either highly inhomogeneous or homogeneous materials. However, in most engineering applications, the materials are rarely highly Inhomogeneous. Therefore, the representation in Figure 7.1a is very inefficient from the memory point of view because y_o and g_o are the same for many of the nodes in the mesh.

The representation in Figure 7.1b is only good for homogeneous materials; however, inhomogeneous materials can be represented by using a number of homogeneous meshes. The problem in using such an approach resides in the representation of non-rectangular regions such as the varactor diode element depicted in Figure 3.11.

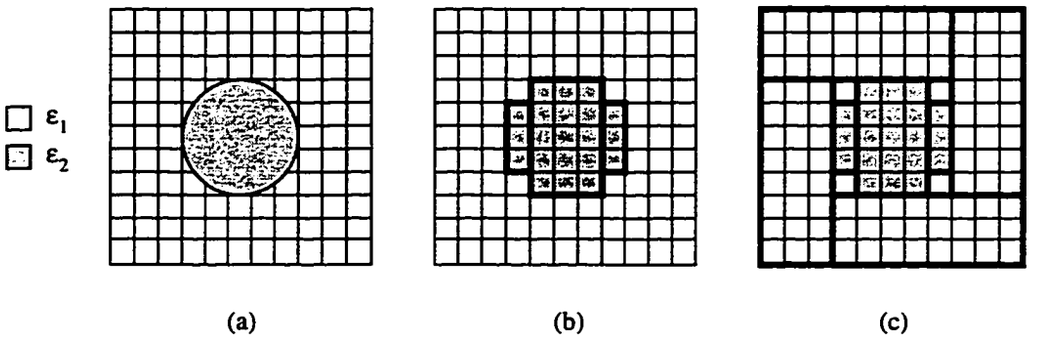


Figure 7.2: The representation of a non-rectangular region using a number of rectangular homogeneous meshes.
(a) A circular region, ε_2, embedded in a rectangular region, ε_1.
(b) The circular region is decomposed into three rectangular regions.
(c) 11 homogeneous rectangular meshes are needed to represent the entire structure.

In order to represent a non-rectangular region using a number of rectangular homogeneous meshes, the non-rectangular region must be decomposed into rectangular sub-regions. These rectangular sub-regions can be represented by rectangular homogeneous meshes. These meshes must be properly connected together to form a single non-rectangular mesh. This procedure is illustrated in Figure 7.2. This is a workable procedure, but a simple structure like the one in Figure 7.2a already requires 11 homogeneous rectangular meshes; the required number of meshes would increase drastically as the complexity of the structure and/or the discretization resolution increase.

7.2.2 A Versatile Implementation

A better way to implement the inhomogeneous mesh is to use a two-dimensional array of pointers in conjunction with a number of one-dimensional arrays of nodes, node buffers (Figure 7.3). The size of a 32-bit pointer is equal to the size of a single precision floating point number. The maximum memory overhead of this implementation is:

- 20 percent for 2D shunt and series TLM.
- 5.5 percent for 3D condensed node TLM

The disadvantage of memory overhead is compensated by the advantages that this implementation can offer. Figure 7.4 shows how this node buffer structure can be used to represent the structure in Figure 7.2.

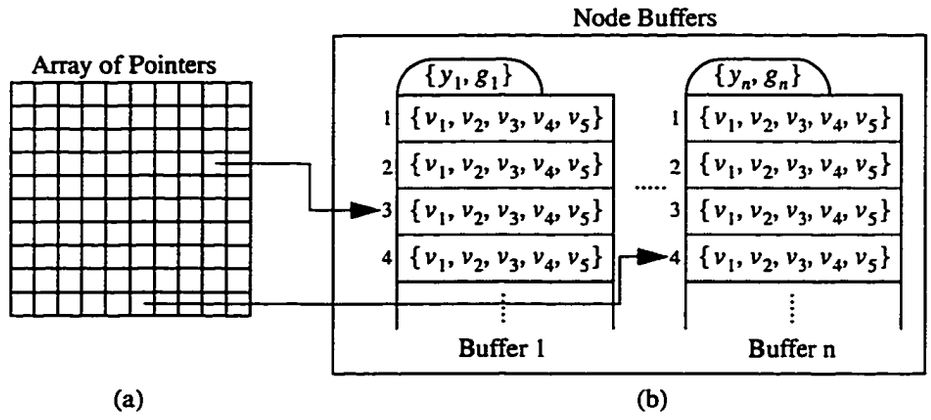


Figure 7.3: A versatile implementation of the 2D-TLM shunt mesh.
(a) A two-dimensional array of pointers to TLM node.
(b) A number of node buffers.

The number of node buffers required to represent a structure is proportional to the number of distinct material regions. Therefore, the number of buffers will not increase when the complexity of the structure is increased or when the resolution of discretization is increased.

The total amount of memory required by the two node buffers in Figure 7.4 is less than the total amount of memory required by the eleven rectangular meshes in Figure 7.2; because in addition to storing the voltage impulses, each mesh in Figure 7.2 has to store the values of a set of material parameters, y_o and g_o. This is true for the node buffers as well; however, there are only two node buffers in this case.

This versatile implementation has a run time overhead — the code segment that is associated with the array structure needs to de-reference the pointers in order to locate the actual address of the variable that store the voltage values; this is emphasized in Figure 7.3 with two arrows pointing from the array to the node buffers. Such an overhead can be reduced to a minimum if the code for the scattering operation is associated with the node buffer rather than to the array. This idea can be re-expressed with the C++ jargon: the code for the scattering operation should be placed in the member function of the node buffer object rather than in the member function of the array object. An example is shown in *Appendix A.2*.

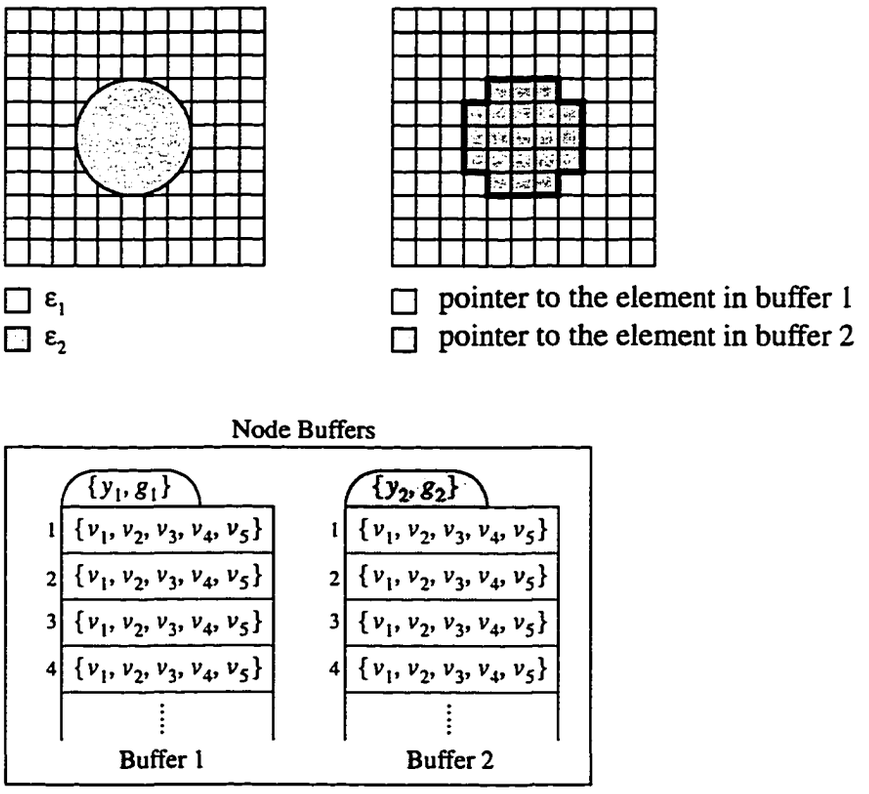


Figure 7.4: A node buffer representation of the structure depicted in Figure 7.2. This representation requires only two node buffers regardless of the discretization resolution.

7.3 Efficient Implementation of the TLM Features

The formulae for the TLM features given in the previous chapters are not necessarily in the best formats for efficient computer implementation. For example, the scattering formula for the 2D-TLM shunt node is given in simple matrix format (Eq 3.8) requires 25 multiplications/divisions and 25 additions/ subtractions. If it is rewritten as follows:

$$
\begin{aligned}
t &= \frac{2}{y} \times (v_1 + v_2 + v_3 + v_4 + v_5 y_o) \\
v_1 &= t - v_1 \\
v_2 &= t - v_2 \\
v_3 &= t - v_3 \\
v_4 &= t - v_4 \\
v_5 &= t - v_5
\end{aligned}
\qquad \text{Eq 7.1}
$$

it requires only 3 multiplications/divisions and 9 additions/subtractions. Similarly, the scattering formula for the 2D-TLM series node given in Eq 3.12 can be rewritten as:

$$
\begin{aligned}
t &= \frac{2}{z} \times ((v_2 + v_3) - (v_1 + v_4 + v_5)) \\
v_1 &= v_1 + t \\
v_2 &= v_2 - t \\
v_3 &= v_3 - t \\
v_4 &= v_4 + t \\
v_5 &= v_5 + t z_o
\end{aligned}
\qquad \text{Eq 7.2}
$$

A similar rearrangement of the 3D-TLM condensed node scattering formulae have also been discussed in *Section 4.4.*

Indeed, this kind of rearrangement can be applied to other formulae as well. For instance, the interface scattering formula for the interface between a regular cell and an adaptive cell (Eq 6.11) requires 8 additions/substraction and 8 multiplication/division operations; it can be rewritten to achieve a higher level of computing efficiency as follows:

$$
\begin{aligned}
B_i &= \frac{n-1}{n+1}A_i + \frac{2}{n+1}A_{i+1} \\
&= \frac{n-1}{n+1}A_i + \left(\frac{2}{n+1}A_{i+1} - A_{i+1}\right) + A_{i+1} \\
&= \frac{n-1}{n+1}A_i + \frac{1-n}{n+1}A_{i+1} + A_{i+1} \\
&= \frac{1-n}{1+n}(A_{i+1} - A_i) + A_{i+1}
\end{aligned}
\qquad \text{Eq 7.3}
$$

Similarly

$$
\begin{aligned}
B_{i+1} &= \frac{2n}{n+1}A_i - \frac{n-1}{n+1}A_{i+1} \\
&= \frac{1-n}{1+n}(A_{i+1} - A_i) + A_i
\end{aligned}
\qquad \text{Eq 7.4}
$$

Therefore, a computationally efficient format is:

$$
\begin{aligned}
K &= \frac{1-n}{1+n}(A_{i+1} - A_i) \qquad \text{where} \qquad n = \delta/\Delta l \\
B_i &= K + A_{i+1} \\
B_{i+1} &= K + A_i
\end{aligned}
\qquad \text{Eq 7.5}
$$

This rearranged formulation only requires 5 additions/subtractions and 2 multiplications/divisions. Similarly, Eq 6.21 can be rewritten as:

$$
\begin{aligned}
K &= \frac{1-n}{1+n}(A_{i+1} - A_i) \qquad \text{where} \qquad n = \delta_2/\delta_1 \\
B_i &= K + A_{i+1} \\
B_{i+1} &= K + A_i
\end{aligned}
\qquad \text{Eq 7.6}
$$

7.4 Parallel Computing

Physical reality, which is closely emulated by the TLM process, requires voltage impulses to travel at a finite speed. Any change in the electrical state of the TLM mesh can thus only propagate over a finite distance during a small time interval. By definition, the elementary time step Δt corresponds to a distance Δl covered at the speed of the voltage impulse on the link line at a speed v. Therefore, both scattering and transfer of TLM impulses are highly localized operations. The former is totally localized within one node, and the latter is localized within a region formed by that node and its immediate neighbours. Hence, scattering of impulses can be carried out in each node simultaneously; and transfer of impulses can be carried out in the region of each node simultaneously as long as synchronization among adjacent nodes is preserved. In other words, these two operations can be carried out in parallel, at each node and in the region of each node. Before the parallel TLM algorithms and their implementations can be discussed, it is better to describe, in general terms, the architecture of the parallel computer that was used during the course of this study.

7.4.1 The DECmpp 12000 Massively Parallel Computer

The architecture of the DECmpp 12000 is shown in Figure 7.5. It has a master processor and a large number of parallel slave processors. The master processor is called the Array Control Unit (ACU) and must be driven by a workstation front-end (FE). The parallel slave processors are called the Processor Elements (PEs). These PEs have a small amount of random access memory and are connected to each other in a rectangular toroidal wraparound fashion [47]. It is a (Single Instruction Multiple Data (SIMD) machine. Massively Parallel Language (MPL) and Massively Parallel Fortran (MPF) are the two high level languages available for programming this machine.

MPL is a C-like language. It has serial and parallel instructions which are executed on the ACU and PEs, respectively. It also has singular and plural data which are allocated on the ACU and PEs, respectively, Figure 7.6a. Communications among PEs are achieved via the MPL's X-net or global router instructions, [47]. Object libraries written in MPL can be called from other popular serial high level languages (C, C++ and Fortran) through the MPL `callRequest` function. In that case the ACU and PEs are collectively called the back-end (BE) and the workstation that executes the serial module is called the front-end (FE) (Figure 7.6b).

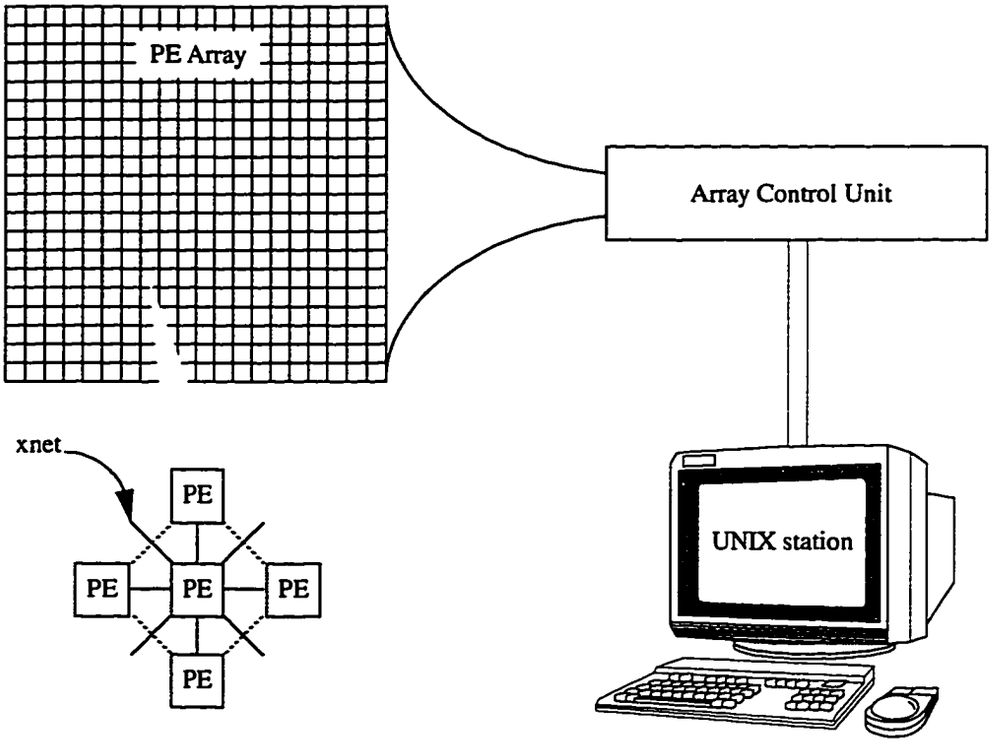


Figure 7.5: DECmpp 12000 internal hardware details in a simplified form.

MPF is a Fortran-77 like language with Fortran-90 array features. Data declared and/or used in the Fortran-90 array formats are allocated on the PEs, otherwise on the FE. Instructions written in the Fortran-90 array formats are executed on the PEs, otherwise on the FE. The ACU is not available for use by MPF programs; the PEs are also transparent to MPF programs. MPF modules cannot be called by other languages but they can call MPL modules (Figure 7.6c).

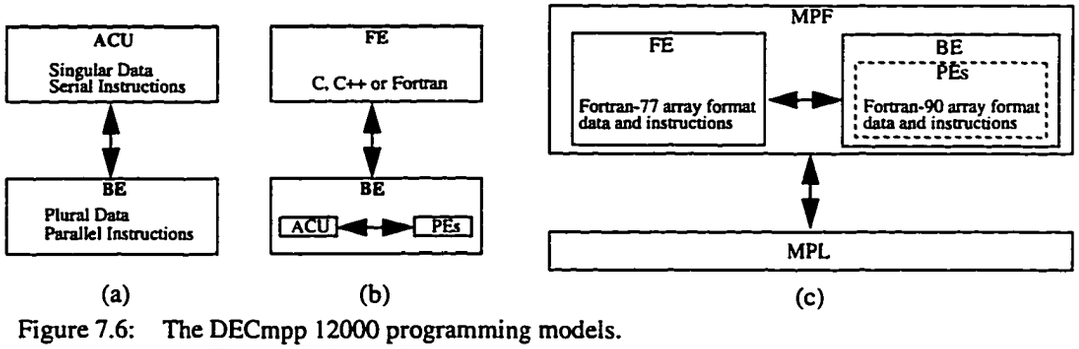


Figure 7.6: The DECmpp 12000 programming models.
(a) The stand-alone MPL programs.
(b) Modules driven by other high level front-end language.
(c) The stand-alone MPF programming model.

MPL and MPF languages have their advantages and disadvantages that depend on the types of applications. MPL programs have better control over the hardware (via low level system functions which are callable only by MPL programs) and allow a better modular design. On the other hand, MPF programs have a rich array syntax which makes the program easier to understand and is independent of the architecture of the DECmpp 12000; thus the program can be easily ported to other parallel computers with minor or no modification. A MPL implementation of the TLM method is given in [11]. Because of the elegant array syntax of MPF, MPF is used in this thesis to illustrate the parallel TLM algorithms.

7.4.2 Parallel Implementation of the 2D-TLM Algorithm

Traditional serial computers have no provision to take advantage of the intrinsic parallel property of the TLM method. No matter what programming language is used, the computer has to iterate through the whole mesh, one node at a time, to perform TLM impulse scattering and transfer operations. This idea is best illustrated by using some 2D-TLM computer code. After expanding Eq 3.2, the formulae for the scattering operation are:

$$ {}_{k+1}V_n^r = {}_kV_z - {}_kV_n^i \quad \text{for} \quad n = 1, 2, 3, 4. \qquad \text{Eq 7.7} $$

where ${}_kV_z = \frac{1}{2}({}_kV_1^i + {}_kV_2^i + {}_kV_3^i + {}_kV_4^i)$. Thus, a typical Fortran-77 subroutine for a 2D-TLM mesh of size NX × NY would be:

```
Do x = 1, NX
  Do y = 1, NY
    Vz = 0.5*(V1(x,y) + V2(x,y) + V3(x,y) + V4(x,y))
    V1(x,y) = Vz - V1(x,y)
    V2(x,y) = Vz - V2(x,y)
    V3(x,y) = Vz - V3(x,y)
    V4(x,y) = Vz - V4(x,y)
  End Do
End Do
```

Code 7.1

The above code can be rewritten using the array feature of Fortran-90 as:

```
Vz = 0.5*(V1 + V2 + V3 + V4)
V1 = Vz - V1
V2 = Vz - V2
V3 = Vz - V3
V4 = Vz - V4
```

Code 7.2

In a serial computer, Code 7.1 and Code 7.2 perform TLM scattering in a sequential manner. However, Code 7.2 requires more data memory and execution

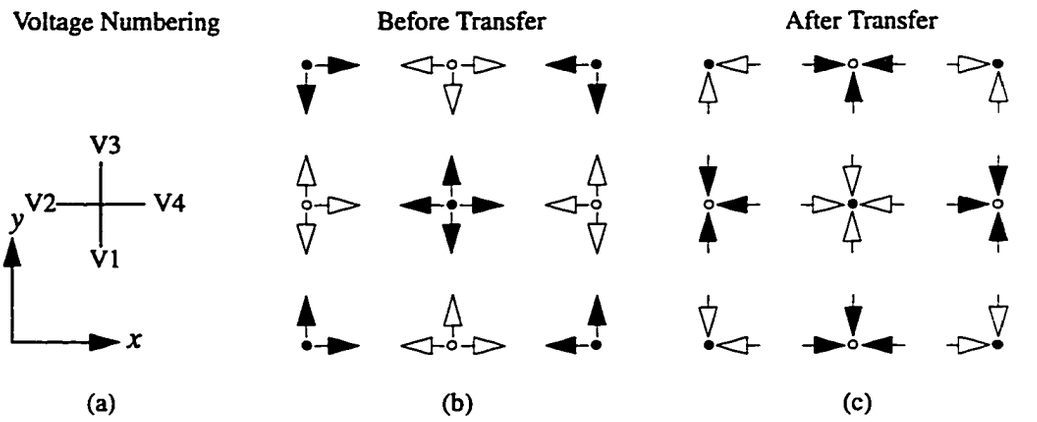


Figure 7.7: A mesh of size 3×3, i.e. substitute NX=NY=3 to Code 7.3. Note that, there is no boundary in the mesh; hence, the Where statements in Code 7.3 will return TRUE in all the processors.
(a) Voltage numbering scheme.
(b) Position of impulses before execution of Code 7.3.
(c) Position of impulses after execution of Code 7.3.

time than Code 7.1 because Vz in the second case is an array rather than a simple REAL variable. Despite the compactness of the Fortran-90 version, the compiler has to expand the array version to an equivalent loop version so that the computer can perform the required computing task in a sequential manner. However, in a massively parallel computer such as the DECmpp 12000, Code 7.2 can be executed in parallel.

Similarly, the transfer of impulses between nodes can also be performed in parallel. The following code fragment illustrates how this can be done by using the array feature of MPF:

```
T1(1:NX,1:NY-1) = V1(1:NX,2:NY  )
T2(1:NX,2:NY  ) = V3(1:NX,1:NY-1)
Where (R1.EQ.0.0) V1=T2
Where (R3.EQ.0.0) V3=T1
T1(1:NX-1,1:NY) = V2(2:NX,  1:NY)
T2(2:NX,  1:NY) = V4(1:NX-1,1:NY)
Where (R2.EQ.0.0) V2=T2
Where (R4.EQ.0.0) V4=T1
```

Code 7.3

Let us apply the above code fragment to the example depicted in Figure 7.7. T1 and T2 are temporary variables to shift the impulses to their neighbouring processors. R1, R2, R3 and R4 are variables to store boundary information; since the mesh in Figure 7.7 has no boundary, the Where statements will return TRUE in all the processors. That means, all processors associated with V1, V2, V3 and V4 are active

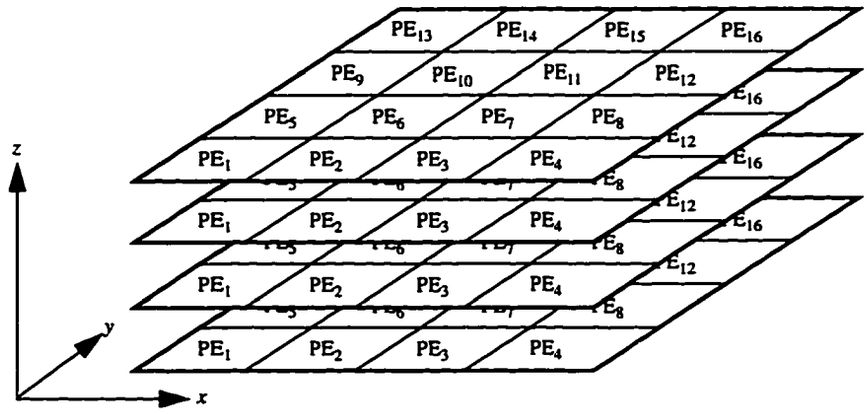


Figure 7.8: A hypothetical situation. A 16-PE mpp is used to represent a 3D mesh with $4 \times 4 \times 4$ nodes. Each PE represents 4 nodes in the mesh: PE_1 represents nodes (1,1,1), (1,1,2), (1,1,3) and (1,1,4), PE_2 represents nodes (2,1,1), (2,1,2), (2,1,3) and (2,1,4), and so on. The volume is thus processed slice by slice.

and take part in the corresponding conditional assignment statements. The overall effect of the parallel shift and parallel conditional assignment statements is the desired parallel TLM impulse transfer operation.

7.4.3 Parallel Implementation of the 3D-TLM Algorithm

The concept of the parallel 3D-TLM algorithm is similar to that of the parallel 2D-TLM, but there is an important difference in its execution due to the architecture of the DECmpp 12000. As mentioned in *Section 7.4.1*, the processor array of the DECmpp 12000 contains PEs connected with each other in a rectangular toroidal wraparound fashion. In effect, this is a two-dimensional connection! Therefore, the physical layout of the PEs can only be used to represent a slice, say in the *xy*-plane, of a volume in the 3D-TLM mesh. The third dimension, say *z*-direction, must be represented by a one-dimensional array in each PE. That is, each PE of the DECmpp 12000 is used to represent a column of nodes in the *z*-direction as shown in Figure 7.8. This can be done easily by using the multi-dimensional array feature of MPF, for example:

```
Integer   NX,NY,NZ
Parameter (NX=64,NY=64,NZ=16)
Real      V1(NX,NY,NZ)
```

Code 7.4

declares an array of $64 \times 64 \times 16$. The default MPF three-dimensional array mapping is (`xbits`, `ybits`, `memory`), that means the *x* and *y*-directions of the array `V1` are mapped to the *x* and *y*-directions of the processor array, the *z*-direction is mapped to the memory of the PEs.

Case	Simulation Feature	Serial (sec.)	Parallel (sec.)	Speed up (times)
1	Plain Simulation	54.5	2.0	27
2	Impulse, 1 Output Point	54.5	2.3	24
3	Gaussian, 1 Output Point	54.5	3.0	18

Table 7.1:Performance comparison between the serial and parallel versions of the 2D-TLM Simulator. The mesh size is 128×64, which is equal to the number of PEs available in the DECmpp 12000 that is used. The serial version is executed on the DEC5000 front-end and the parallel version is on the DECmpp 12000 with its GUI module executed on the DEC5000 front-end. The number of computation steps for all the cases is 1000 and the number of frequency points for cases 2 and 3 is 100.

To achieve maximum performance, the 3D-TLM mesh should be oriented so that the number of nodes assigned to each PE is reduced to a minimum. For example, if the size of the processor array is 64×64 and the size of the mesh is $16 \times 64 \times 64$, then each PE should be used to represent 16 nodes in the x-direction. The MAP compiler directive can be used to override the default compiler mapping by a new mapping as shown below:

```
        Integer   NX,NY,NZ
        Parameter (NX=16,NY=64,NZ=64)
        Real      V1(NX,NY,NZ)
CMPF    MAP       V1(memory,xbits,ybits)
```

Code 7.5

7.4.4 Acceleration Ratio and Limitations of the Parallel TLM Implementation

A MPL parallel 2D-TLM simulation module has been combined with a UNIX version of the 2D-TLM simulator described in [48]. Table 7.1 summarizes the execution times of the serial and parallel versions of the 2D-TLM simulator for three test cases. The mesh size used in the simulation is 128×64, which is equal to the number of PE's available in the DECmpp 12000 that is used. In all the cases, only the execution time related to the 2D-TLM simulation, sampling of results and performing Fourier Transform (if applicable) is recorded, i.e. execution time related to graphical I/O operations is subtracted from the total time. The serial version is executed on the DEC5000 front-end and the parallel version is run on the DECmpp 12000 with its GUI module executed on the DEC5000 front-end. The number of computation steps for all the cases is 1000 and the number of frequency points for cases 2 and 3 is 100.

The result in Table 7.1 shows that the parallel version of the 2D-TLM simulator performs very well for plain simulation. The performance of the parallel version starts to decrease when the signal is sampled from (case 2), or injected into

the mesh (case 3). This decrease in performance does not appear in the serial version because most of the work is done to perform the 2D-TLM simulation; the work for the sampling and injection of the signal is negligible. In the parallel version, the 2D-TLM simulation is performed in parallel in the DECmpp 12000 BE while the injection and sampling of the signal are performed in series in the DEC 5000 FE. A major reason for such a decrease in performance is the overhead incurred in the communication between the DEC5000 and the DECmpp 12000. Communication functions are called before and after each computation step to inject signals into the mesh and sample signals from the mesh. Another reason is that the output point causes the FE to do some extra work, such as performing a Fourier Transform. The former problem can be solved by keeping the sampled data and the input signal in the BE and transferring the sampled data to the FE after the simulation is completed. The latter problem can be solved by implementing a parallel Fourier Transform algorithm and execute it in the BE.

7.5 Distributed Computing

Massively parallel computing is one way to increase the computational speed of the TLM programs. This computing technique is only available to those who have access to modern massively parallel computers. Porting of the TLM codes, especially 3D code, to those computers is complicated by the fact that a standard parallel computing language remains to be defined.

Another way to accelerate the execution speed of TLM is to divide a large computation tasks into smaller sub-tasks and instruct a group of computers to work on those sub-tasks concurrently; this can be achieved by distributed computing. This computing technique allows a group of computers, both serial and parallel, to work together seamlessly to solve CPU time intensive TLM computations and signal processing applications concurrently. In the UNIX computing environment, distributed computing can be realized by *piping* among different computers.

7.5.1 Piping in the Unix Environment

The UNIX *piping* idea is not difficult to understand and can be easily illustrated by the following command* combination: `ps | sort`. The `ps` command prints the information about all processes† associated with the current terminal. The |

* From now on, command means command in the UNIX operating system.

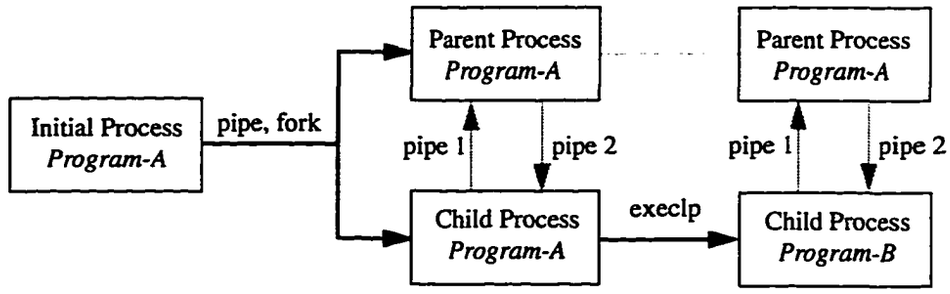


Figure 7.9: The sequence of events that take place when the pipe, fork and execlp commands are executed by *Program-A* to start a *Program-B* child process.

operator sends the output of the `ps` command to a sorting program `sort`. The fascinating thing, which is not obvious to the normal UNIX users, is that the `ps` and `sort` commands are executed concurrently; the data produced by the `ps` command are sent directly to the `sort` command via the internal memory of UNIX. Therefore, no temporary file is needed, and transfer of data between the two programs is very fast. This is a unidirectional *piping* example: `ps` can send data to `sort` but `sort` has no means to send data back to `ps`. This is only a limitation of the command interpreter, or *shell* as it is called, not that of the UNIX operating system.

The `pipe` command allows a process to create a pipe for communication between two processes. Therefore, bidirectional communication between two processes is possible if two pipes are created. Once a process has created two pipes, it could then execute the `fork` and `execlp` commands to start another process. In the UNIX environment, these two processes are called the parent and child process, respectively. The above ideas are illustrated in Figure 7.9 and are discussed in detail in the UNIX literature (see [49] for example).

In order for a field simulation program to take advantage of distributed computing, the program must be broken down into a parent program and one or more child programs. The parent and child processes can communicate with each other through the bidirectional pipes. Distributed computing can be achieved by passing the `rsh`* command, a host name, and a child module name as arguments to the `execlp` command. This would start a child process in the machine specified with the host name argument through a remote shell.

---

† In the UNIX environment, a running instance of a program is called a process.

* The `remsh` command should be used in the HP-UX operating system.

7.6 Optimization of Electromagnetic Structures by Parallel and Distributed Computing

The parallel and distributed computing techniques mentioned in the previous sections have been successfully employed to combine the computing power of a workstation with a DECmpp 12000 massively parallel computer to perform optimization of electromagnetic structures. Figure 7.10 shows a commercially available microwave CAD program, OSA90/hope™, interacting indirectly with a massively parallel TLM program, *tlm-engine*, via the UNIX pipes. *3dtlm-pipe* and *3dtlm-server* are programs that handle structure parameterization and computer communications. This computing setup can be used to optimize the dimensions of microwave structures using TLM [8].

7.6.1 Optimization Examples

The following two optimization examples demonstrate the capability of this computing arrangement. As a first example, the dimensions of an inductive iris bandpass filter are optimized with the random optimizer of OSA90/hope. The geometry and dimensions of the filter are shown in Figure 7.11. The design specifications are:

- Center frequency — 33 GHz.
- Pass band — 32.5 to 33.5 GHz.
- Minimize $|S_{11}|$ and maximize $|S_{21}|$ in the passband and vice versa in the stopband.

The initial length of the cavities is set to $\lambda_g = 11.81\text{mm}$ (rounded off to the nearest Δl). The initial width of the irises is chosen to be small so that the opening

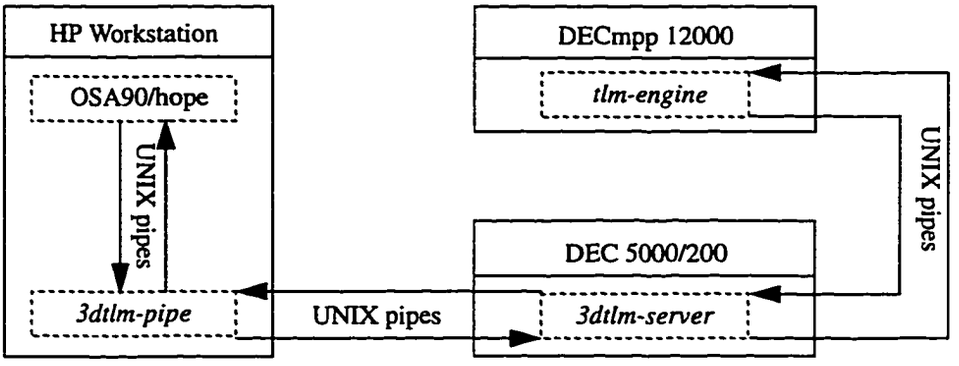
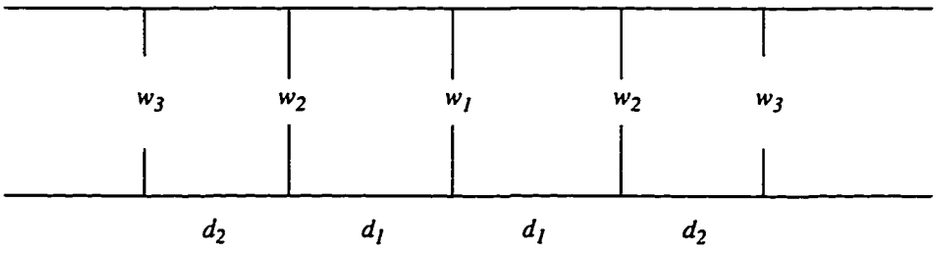


Figure 7.10: Schematic diagram depicts the interaction of OSA90/hope with a massively parallel TLM program, *tlm-engine*, via the UNIX's pipes. The *3dtlm-pipe* and *3dtlm-server* are programs that handle structure parameterization and computer communications.

will not affect the characteristic of the resonators. The initial response of the filter is shown in Figure 7.12(a). The random optimizer of OSA90/hope is used to optimize the values of the parameters. The values of the parameters after 30, 60 and 90 simulations are shown in the table in Figure 7.11; their corresponding responses are shown in Figure 7.12(b), (c) and (d), respectively. The circuit file used for this optimization process is given in *Appendix B.1*.

The second example demonstrates the effect of inserting two tuning screws into the two center cavities of the filter in the above example. The location and depth of insertion are shown in Figure 7.13 together with the corresponding filter responses. The circuit file used for this optimization process is given in *Appendix B.2*.



	Initial	30 Random	60 Random	90 Random
w_1 in Δl	2	4	4	4
w_2 in Δl	2	4	6	6
w_3 in Δl	2	10	10	10
d_1 in Δl	23	21	21	21
d_2 in Δl	23	23	22	23

Figure 7.11: The top view of an inductive iris bandpass filter. The initial and successive optimized values of the parameters are show in the above table, Δl=0.508mm. The desired center frequency is 33 GHz; the passband is from 32.5 GHz to 33.5 GHz. The optimization goal is to have minimum $|S_{11}|$ and maximum $|S_{21}|$ in the passband and vice versa in the stopband.

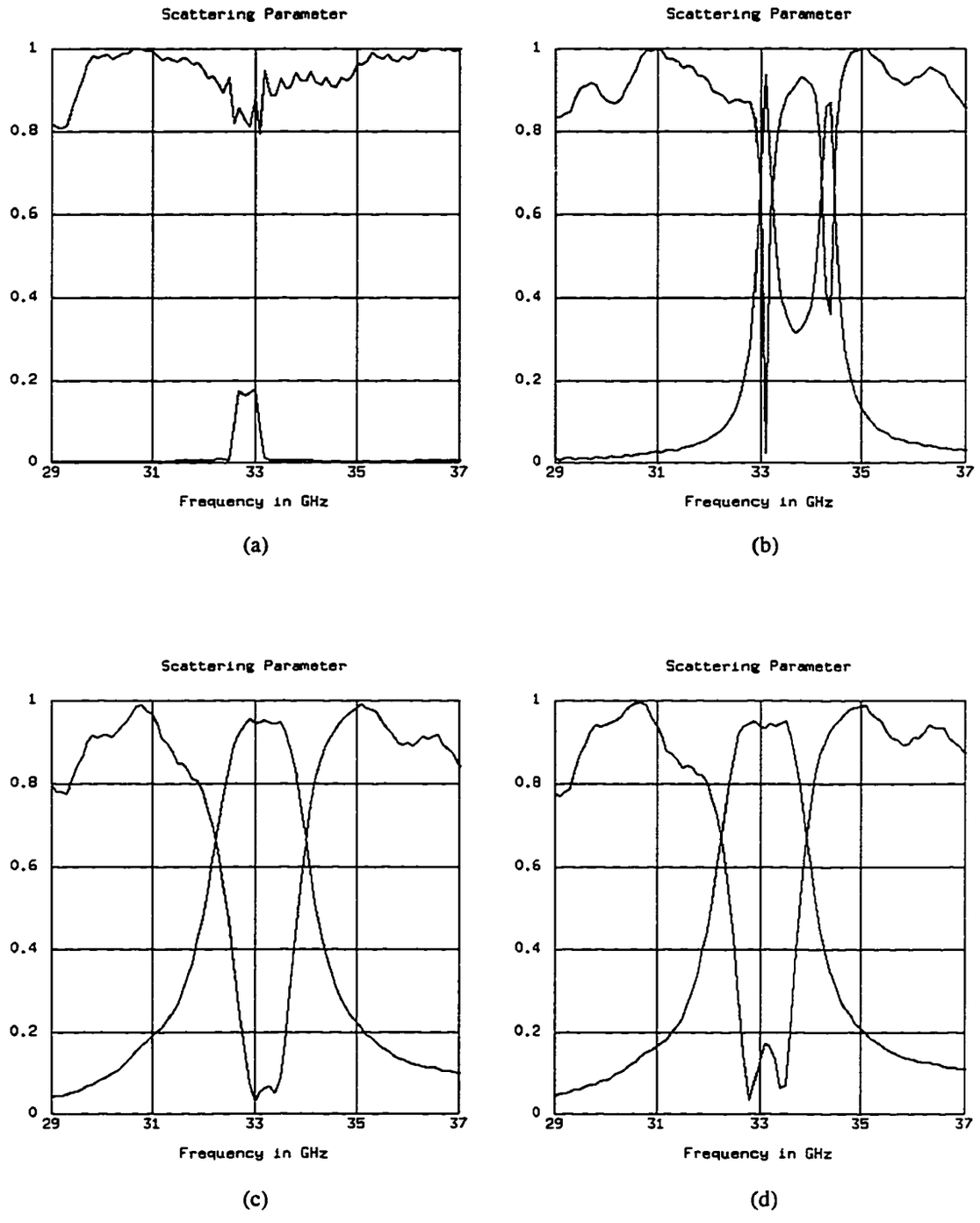
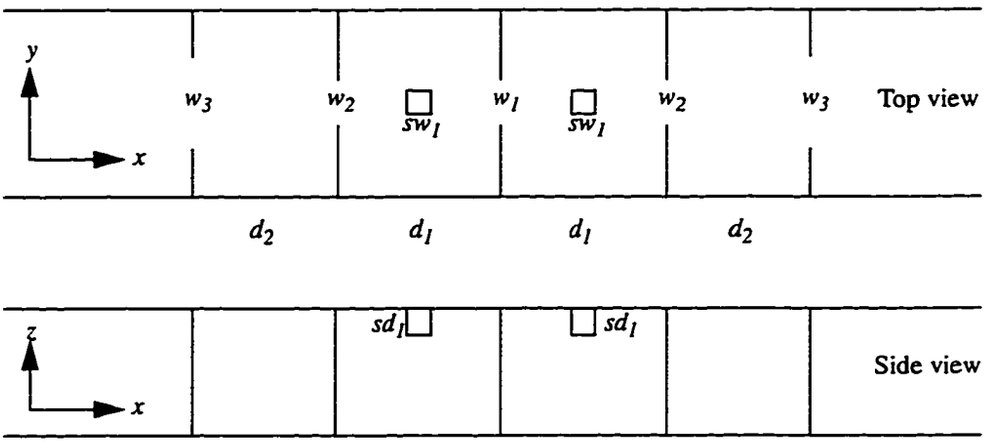
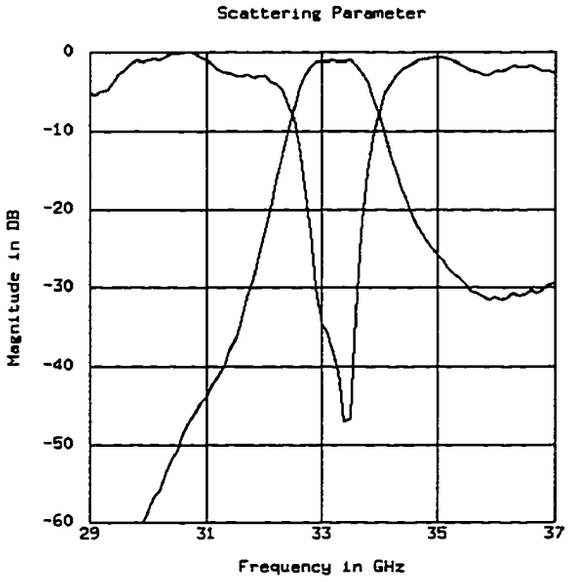


Figure 7.12: The responses of the inductive iris filter shown in Figure 7.11. (a) is the initial response; (b), (c) and (d) are the best responses after the 30^{th}, 60^{th} and 90^{th} simulations with parameter values randomly chosen by the random optimizer of OSA90/hope.



(a)



(b)

Figure 7.13: (a) Top and side views of a rectangular waveguide iris coupled bandpass filter. The values of w_1, w_2, w_3, d_1 and d_2 are given in the last column of the table in Figure 7.11. sw_1 and sd_1 are both 2 Δl.
(b) The response of the filter in (a) after 8000 time steps.

7.7 Signal Processing Techniques

Parallel and distributed computing can accelerate the execution speed of TLM computation. However, reliance on computer power alone is not sufficient since the computational requirement to solve realistic problems could become intractable as the resolution in the computational domain is increased by increasing the density of node in the domain. On the other hand, spectral estimation techniques such as Prony's method and the autoregressive moving average (ARMA) technique can be used to reduce the amount of required TLM computation for narrow band and high quality factor circuits, especially those with sharp metal edges, and hence increases the efficiency of the TLM method. Parallel implementations of these algorithms are also possible. These techniques avoid unnecessary TLM computations by predicting the future response from a short record of the past time response, or predicting the transfer function of the circuit from short time responses at the input and output ports of the circuit. The following sections give a brief description of the Prony and autoregressive moving average (ARMA) technique described in [3], which is the work of Dr. Eswarappa.

7.7.1 Prony's Method

In this method, a signal in the time domain is represented as a sum of complex exponentials representing damped sine waves, [6], [17] and [25]:

$$u(n\Delta t) = \sum_{k=1}^{p} A_k e^{j\phi_k} e^{(\alpha_k + j2\pi f_k) n\Delta t} \qquad n = 1, 2, \ldots, N \qquad \text{Eq 7.8}$$

where A_k is the modal amplitude, ϕ_k is the modal phase, α_k is the damping factor per second, f_k is the resonant frequency in Hertz of the mode, p is the order of the model which is twice the number of resonant modes.

Estimation of these modal parameters mainly consists of three steps: first, a linear model of suitable order is fitted to the data, then the roots of the associated characteristic polynomial formed by the linear prediction coefficients are computed to yield the estimates of the damping factors and sinusoidal frequencies of each of the exponential terms, and finally data are used again to estimate the amplitudes and initial phases of the complex exponentials by solving a linear least square problem.

Once the modal parameters have been computed using the time samples from the early time domain response, the time and frequency responses of the

given signal can be computed using the following equations (for complex conjugate roots):

$$u_n = \sum_{k=1}^{q} A_k \, e^{-\alpha_k n \Delta t} \cos(2\pi f_k n \Delta t + \phi_k), n = 1, 2, \ldots, N \qquad \text{Eq 7.9}$$

$$u(\omega) = \frac{1}{2} \sum_{k=1}^{q} \left(\frac{A_k e^{j\phi_k}}{\alpha_k + j(\omega - \omega_k)} + \frac{A_k e^{-j\phi_k}}{\alpha_k + j(\omega + \omega_k)} \right) \qquad \text{Eq 7.10}$$

where q is the number of resonant modes. Note that the Fourier transform of the time response is available in closed form Eq 7.10, so that frequency domain results are immediately available when the modal parameters have been determined. This avoids computation of the time-consuming numerical Fourier transform.

7.7.2 ARMA Modeling

The ARMA technique is a more general pole-zero model; with this technique the scattering parameters of microwave circuits can be computed directly from the ARMA coefficients [5], [14] and [30]. The short time records picked up at the input and output ports of the structure under test are used to estimate these ARMA coefficients.

The input and output time signals $[x_k]$ and $[y_k]$ of any discrete linear system are related by the following linear difference equation:

$$y_k = \sum_{n=0}^{L} b_n x_{k-n} - \sum_{n=1}^{L} a_n y_{k-n} \qquad \text{Eq 7.11}$$

where L is the order of the system. The transfer function $H(z)$ of this discrete model (digital filter) is given by

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_L z^{-L}}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_L z^{-L}} \qquad \text{Eq 7.12}$$

where $X(z)$ and $Y(z)$ are the z-transforms of the data sequences $[x_k]$ and $[y_k]$.

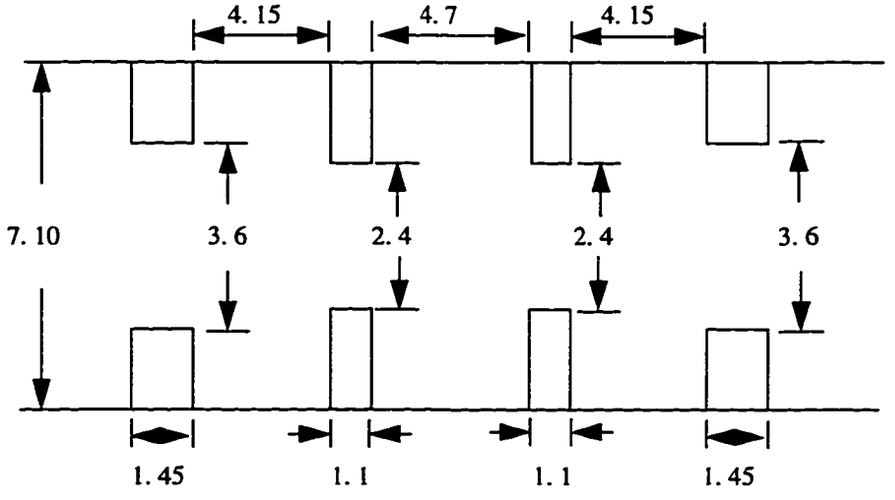
Using Eq 7.11, and the knowledge of the previous time samples at the output as well as the excitation at the input, the future samples can be predicted. The frequency domain response (scattering parameters) can be obtained by substituting z in Eq 7.12 by $e^{j\omega T}$, thus eliminating the use of Fourier transform.

The pure order recursive least-squares ladder algorithm (PORLA) developed by Strobach [30] can be used to estimate the ARMA coefficients. This algorithm requires only the auto-variances and cross-variances of the input and output data sequences. The ARMA model is derived through a two-channel AR predictor. The parameters of this two-channel AR predictor are obtained from the ladder reflection matrices using Levinson-type recursion.

The time samples obtained at the input and output ports from a TLM analysis can be fed directly into an algorithm which computes the ARMA coefficients, in real time. The TLM algorithm can be stopped once the ARMA coefficients converge to their final values. It is to be noted that the above PORLA algorithm of ARMA coefficients estimation can utilize the parallel processing capabilities of a massively parallel processor computer [30].

7.7.3 Simulation Examples

A three-section WR28 waveguide bandpass filter at 35 GHz, [10], has been used to demonstrate the above concepts. The filter geometry and the dimensions are shown in Figure 7.14. Only one-half of the structure is needed to be discretized because of



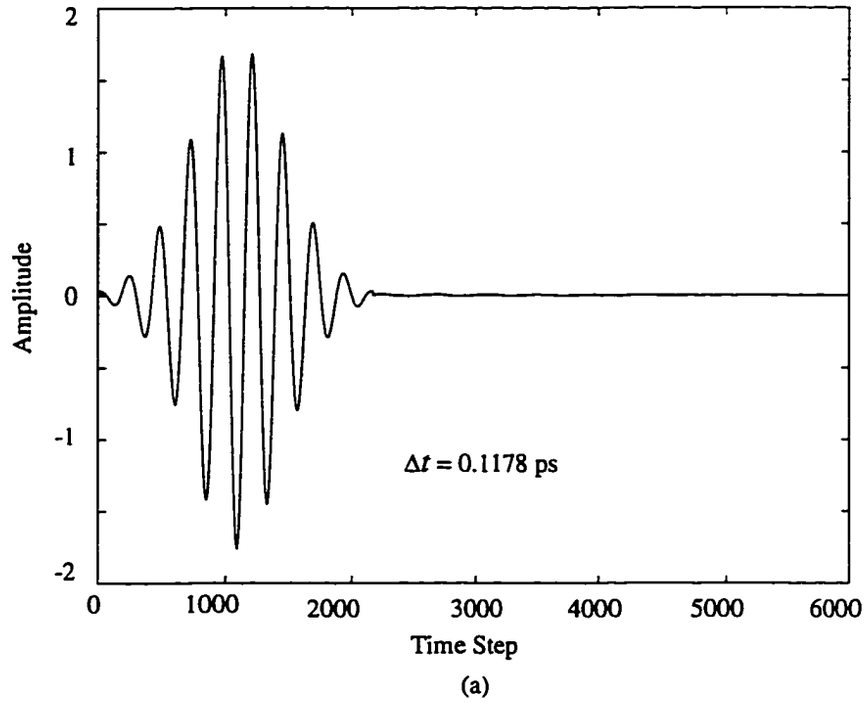
Mode Number	Frequency [GHz]	Damping Factor	Amplitude	Phase [rad]
1	34.558	-0.01538	0.2728	-1.6360
2	35.743	-0.01885	0.3335	-0.9823
3	35.097	-0.03357	0.5737	1.8717

Figure 7.14: The geometry and modal parameters of a waveguide bandpass filter used to demonstrate the signal processing concepts in the previous sections; the dimensions are in mm.

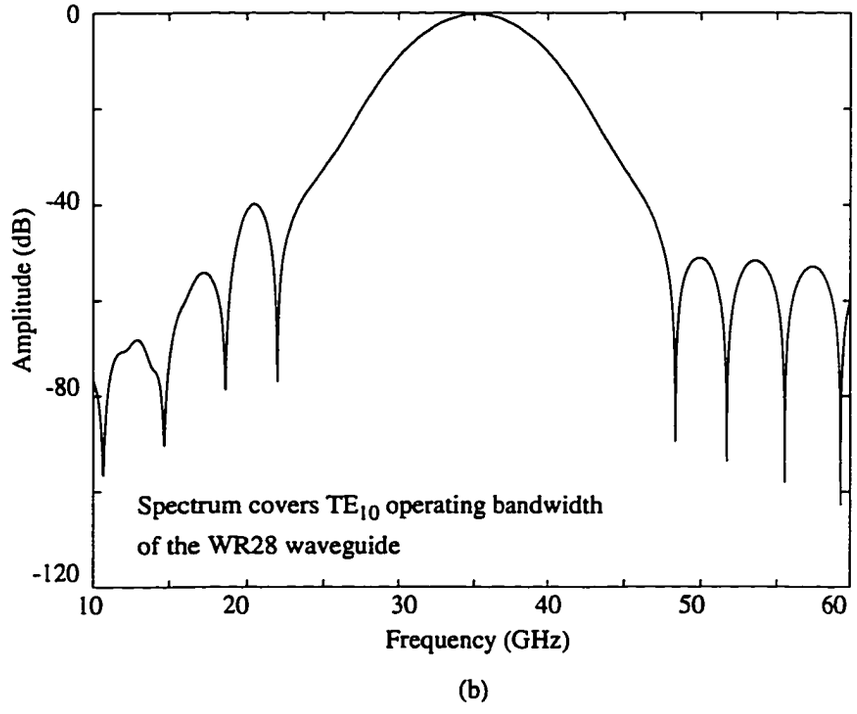
its symmetry about the longitudinal axis. The size of the TLM mesh is 564×73 with a 0.05 mm node spacing, or Δl, (the time step Δt is therefore 0.11785 ps). The filter is excited with a cosine modulated Gaussian pulse. The carrier frequency and the width of the Gaussian pulse are chosen such that only the frequencies that cover the passband of the TE_{10} mode of the WR28 waveguide are excited. It is very important not to excite the frequencies below the first cutoff frequency (21 GHz) of the waveguide. These steps are essential to minimize the noise in the TLM time domain responses. Figure 7.15(a) shows the response of a uniform waveguide (terminated with one-way equation absorbing boundaries at the input and output ends) excited with a cosine modulated Gaussian pulse. This incident waveform is required for the S-parameter extraction. Its frequency spectrum is plotted in Figure 7.15(b); it can be seen that it covers only the passband TE_{10} mode of the WR28 wave guide. The TLM time domain response obtained at the output port of the filter shown in Figure 7.14 is plotted in Figure 7.16(a) for 50,000 time steps.

The time step used in the TLM analysis of the filter shown in Figure 7.14 is 0.11785 ps. This corresponds to a sampling rate of 8488 GHz. But according to the sampling theorem, the sampling rate required is equal to or greater than twice the highest frequency component of the desired filter response (this is approximately 80 GHz). Hence, it is very clear that the original TLM signal is over sampled by more than 100 times. The original TLM samples are decimated at a rate of 1 in 100 to obtain the input data for Prony's and ARMA models.

The modal parameters obtained using Prony's method from 100 such time samples are given in Figure 7.14. These three modes correspond to the three fundamental resonances of the resonators of the filter. The time response computed using these modal parameters and Eq 7.9 is plotted in Figure 7.16(b). This agrees very well with the original TLM response shown in Figure 7.16(a). The insertion loss, S21, of the filter computed using these modal parameters and Eq 7.10 (after normalizing with respect to the response of the uniform guide) is plotted in Figure 7.17(a). Also plotted in this figure are the curves obtained through discrete Fourier transform (DFT) of the 12,000 and 50,000 TLM time samples. It is clearly shown in the figure that the result obtained with Prony's method agrees well with the curve obtained with the 50,000 samples. The result computed with DFT based on 12,000 samples is far from convergence, while Prony's method yields the correct result with just 100 samples (decimated from 10,000 TLM samples). In this particular example, Prony's method has increased the computational efficiency of the TLM analysis by more than five times.

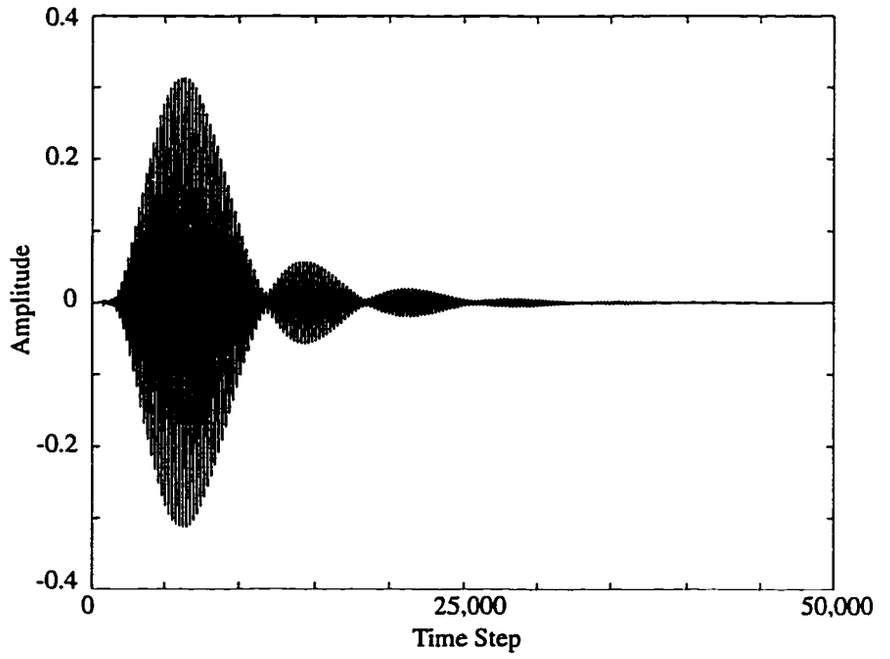


(a)

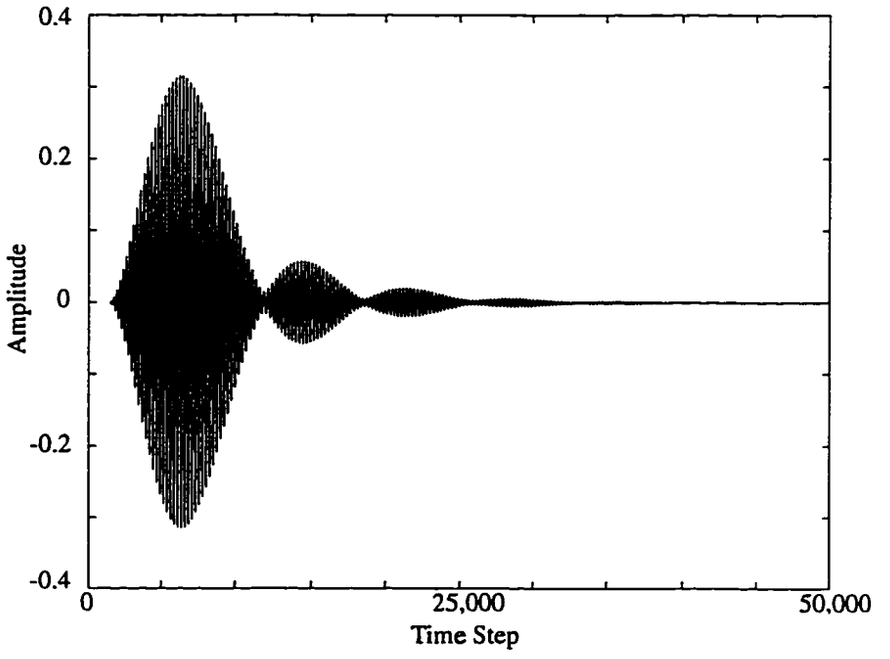


(b)

Figure 7.15: Response of a uniform waveguide to a cosine modulated Gaussian pulse excitation.
(a) Time domain response.
(b) Spectrum of the time domain response in (a).

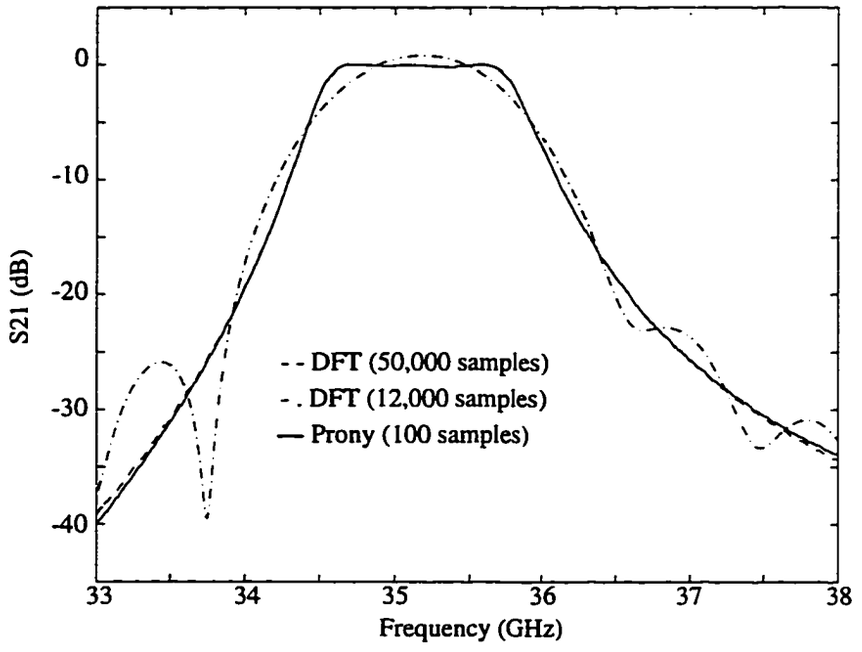


(a)

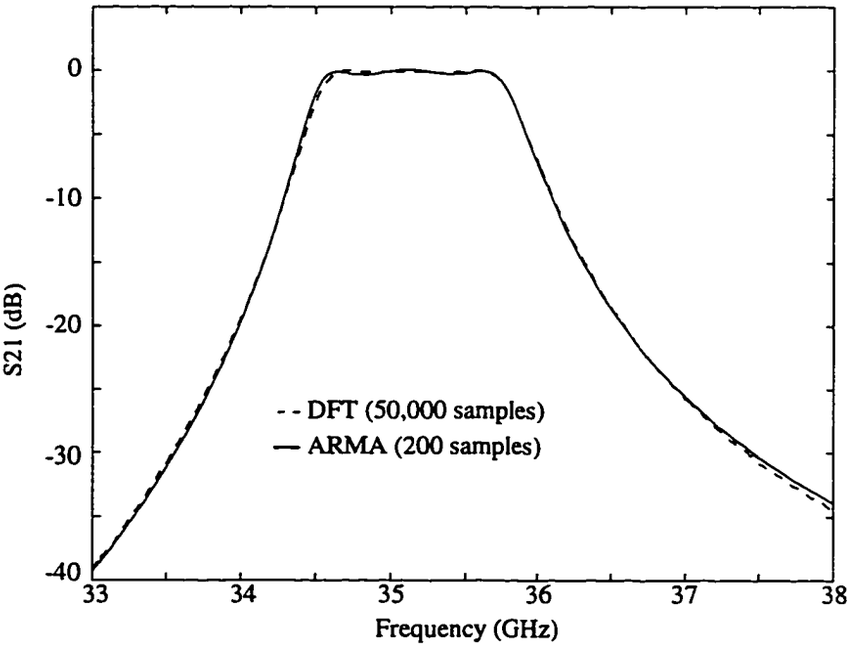


(b)

Figure 7.16: Time domain response at the output of the filter in Figure 7.14,
(a) Obtained using direct TLM.
(b) Obtained using TLM with Prony's method, Eq 7.9. The input excitation is the signal in Figure 7.15a.



(a)



(b)

Figure 7.17: Comparison of S-parameters of the filter in Figure 7.14.
(a) Computed using Prony's method and DFT of 12,000 and 50,000 time samples.
(b) Computed using ARMA and DFT of 50,000 time samples.

For ARMA modeling, the input data sequence $[x_k]$ is obtained by decimating the TLM time-domain response of the uniform waveguide, Figure 7.15(a), at the rate of 1 in 100. The output data sequence $[y_k]$ is obtained by decimating the transmitted wave, Figure 7.16(a). The insertion loss computed directly from the ARMA coefficients, using Eq 7.12, is plotted in Figure 7.17(b); the number of time samples and the order of the ARMA model are 200 and 50, respectively. The result agrees well with the loss computed with DFT based on 50,000 original TLM samples.

7.8 Conclusion

Special computing algorithms that can increase the modelling flexibility as well as the execution speed of the TLM method are presented. The versatile implementation of the TLM method based on pointers and node buffers allows TLM to model highly inhomogeneous materials easily. The parallel and distributed computing algorithms can accelerate the computation performance of TLM by more than an order of magnitude. Indeed, these two computing techniques are combined together to form a distributed parallel computing algorithm.

An optimization procedure for electromagnetic structures based on the above algorithms and OSA90/hope is also presented. This procedure has been used to optimize the geometry of a waveguide bandpass filter with tuning screws. The result demonstrated that this optimization procedure is powerful feature for a general purpose microwave engineering CAD tool.

Parallel and distributed computing by themselves cannot accelerate the performance of TLM drastically if the structure under study has a very high quality factor; in this case signal processing techniques are required to extract the appropriate response from a short TLM time response. The signal processing techniques based on Prony's method and ARMA modelling [3] have also been discussed.

CHAPTER 8 Discussion and Future Work

8.1 Introduction

The design of complex electromagnetic structures with TLM is a very challenging task. Therefore, advanced TLM features and computation techniques have been developed in this thesis to overcome some of the difficulties. In the order of their introduction in this thesis, they are:

- Recursive Generation of Johns Matrix,
- Modelling of Non-Linear Devices,
- Corner Node,
- New 3D-TLM node Numbering Schemes,
- 3D-TLM Impulse Splitting Procedure,
- Parallel Computing Algorithms,
- Distributed Computing Algorithms,
- Optimization Procedures,
- Adaptive Cell,
- Adaptive Mesh,
- Automatic Discretization of Elements using Adaptive Mesh.

These features and techniques do increase the modelling capabilities as well as the computation efficiency of the TLM method. However, they must be incorporated

into a simulation tool before TLM can be used strategically in conjunction with linear circuit simulators to perform analysis and optimization. Since the commercially available simulation tools do not provide provisions for incorporating these feature effectively into them, a simulator has been created to verify the correctness and flexibility of these features. This chapter gives a high level description of the simulator and an assessment of the contribution of this thesis as well as the possibilities for future research.

8.2 The Design of a Multi-Purpose Electromagnetic Field Simulator

This section identifies some features of the simulator which can be executed independently of each other as well as in collaboration with each other via a master program to form a simulation tool. Figure 8.1 depicts a number of these programs and shows how they can work together in a sequential manner. The graphical editor produces a high level structure description file for the structure to be analyzed. The mesh generator translates that file to a lower level structure description syntax which can be used by the field solver directly. With the data stored in a file provided by the signal generator, the field solver can then perform the required computation and store the result in an output file. This output file can then be used by the signal processor.

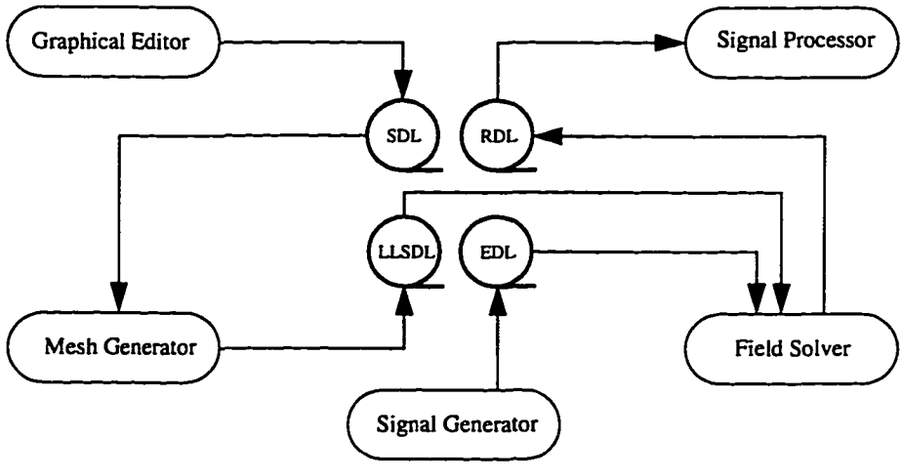


Figure 8.1: The graphical editor, mesh generator, field solver, signal generator and signal processor are separated programs which can work independently of each other as well as in collaboration with each other via a controlling program. The above abbreviations stand for:

SDL	Structure Description Language.
LLSDL	Low Level Structure Description Language.
EDL	Excitation Description Language.
RDL	Response Description Language.

The four description languages (SDL, LLSDL, EDL and RDL) depicted in Figure 8.1 contain provisions that allow incorporation of third party software into the simulation tool. These languages must be of very high level and easy to understand so that the user of the tool can either produce these files with the provided programs or with other professional third party programs.

The programs in Figure 8.1 can also work together concurrently; if the intermediate data files are replaced by pipes (see *Distributed Computing* on page 89) then the interactions among these programs can be controlled by a master program. The gray circle in Figure 8.1 represents such a master program.

8.2.1 The Graphical Editor

The graphical editor takes human input via the mouse and keyboard and produces an SDL output file in either the ASCII or a binary format. It must be user friendly and powerful enough to handle highly complex three-dimensional structures.

Figure 8.2. depicts such a graphical editor with it various components. The event processor respond interactively to human input via the input devices attached to the computer. The processor also directs other components of the graphical editor to perform element editing, storing as well as displaying the elements stored in the element stacks, on the drawing windows.

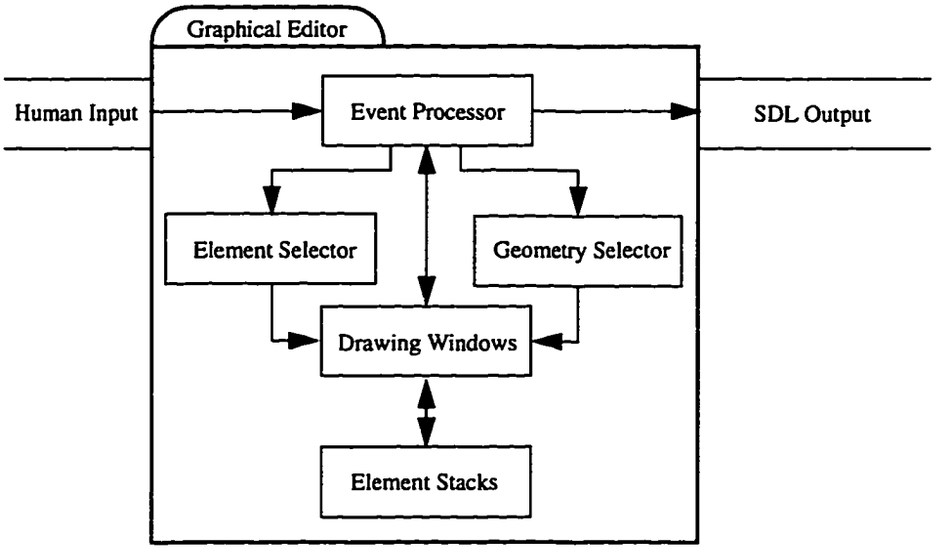


Figure 8.2: A graphical editor with its various components. The event processor is the heart of the editor, it respond interactively to human input and directs other components of the editor to perform various actions.

8.2.2 The Mesh Generator

The mesh generator of this simulator must be powerful enough to handle structures of highly irregular three-dimensional geometry. Figure 8.3 depicts such a mesh generator with some of its components.

After reading in the SDL input, the mesh generator must extract and discretize the geometrical data in order to produce an input data file for the field solver. Because the SDL file can be produced by using an external ASCII editor or other third party programs, the mesh generator must verify the correctness of this file. The grayed path in Figure 8.3 is an error handling mechanism; instead of creating incorrect output data, the mesh generator invokes an external error handling module. This module can be provided by a third party package so that its SDL data producing program can be notified if error is detected.

In general, it is quite impossible to recover from errors without causing problems down the simulation path. Therefore, a simple way to handle error is to report it to the user and then terminate the execution of the mesh generator and/or the simulator.

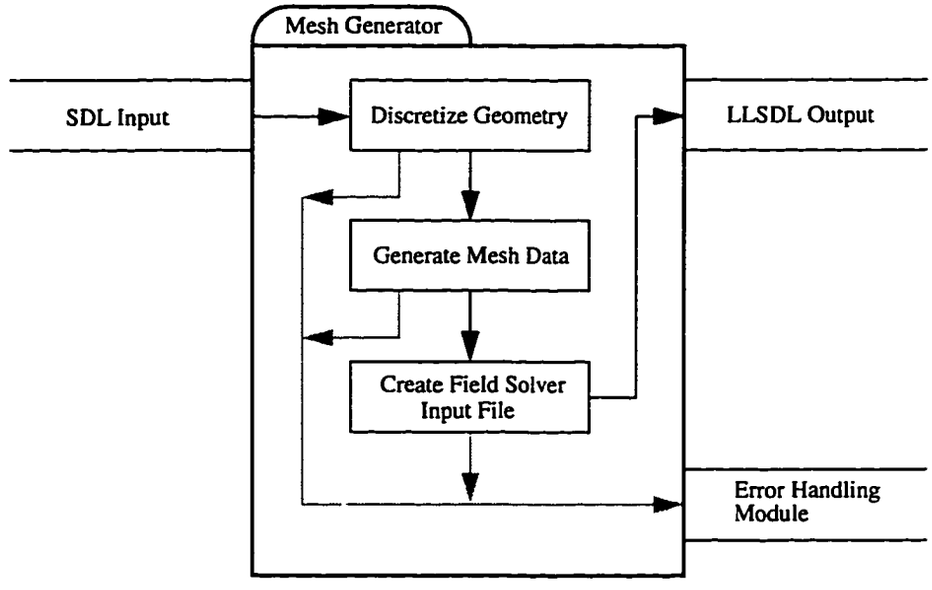


Figure 8.3: A mesh generator with some of its components. The grayed path is an error handling mechanism; instead of creating incorrect output data, the mesh generator invokes an external error handling module.

8.2.3 The Signal Generator

The signal generator accepts human input from the mouse and keyboard, then produces EDL output data in either the ASCII or a binary format. Once the EDL output is produced, execution of the signal generator can be terminated. Figure 8.4 shows some of the components of this signal generator.

The function selector and attribute selector behave like the event processor of the graphical editor; they respond interactively to the human input to determine the appropriate function and its attributes (such as magnitude, frequency, etc.) the user has selected. The information is passed on to the function sampler which generates a discrete version of the function based on the time step of simulation, (see *Definition of Time Step* on page 20), provided internally by the simulation tool and produces the appropriate EDL output file.

The signal generator should provide as many functions as possible, but there is no way to pre-determine and implement all the functions one would like to use to excite the structure under study. Therefore, the function selector should provide a mathematical expression parser option to allow the user to specify some uncommon expressions.

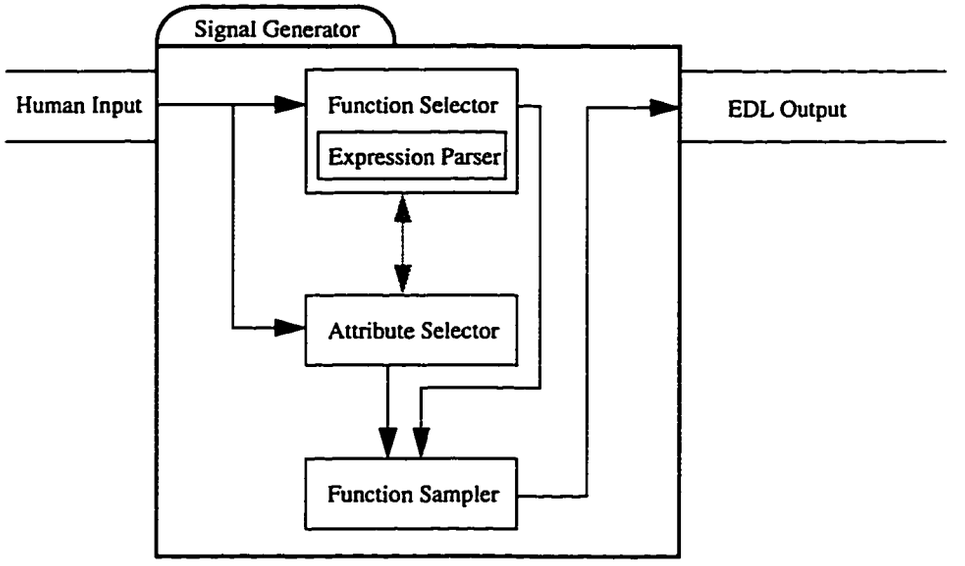


Figure 8.4: A signal generator with its various components. In the event that the built-in option does not include the function that is needed, the function selector should invoke the expression parser to compute the desired function.

8.2.4 The Field Solver

The field solver must take advantage of the existing well-proven time domain methods such as TLM and FDTD as well as analytical methods, so that the best of these methods can be combined whenever and wherever possible. Figure 8.5 depicts such a field solver with its various analytical and numerical methods. These methods should be able to work independently as well as together seamlessly with each other.

A method selection mechanism must be implemented to select the appropriate method whenever possible. It must also provide an option to the user to override its selection.

Because the SDL and EDL input can be produced by third party software, the field solver must verify the correctness of these inputs. The grayed path in Figure 8.5 is an error handling mechanism; instead of creating incorrect output data, the field solver invokes an external error handling module. This module can be provided by a third party package so that its LLSDL and EDL data producing program can be notified if error is detected. However, a simple way to handle error is to terminate the execution of the field solver.

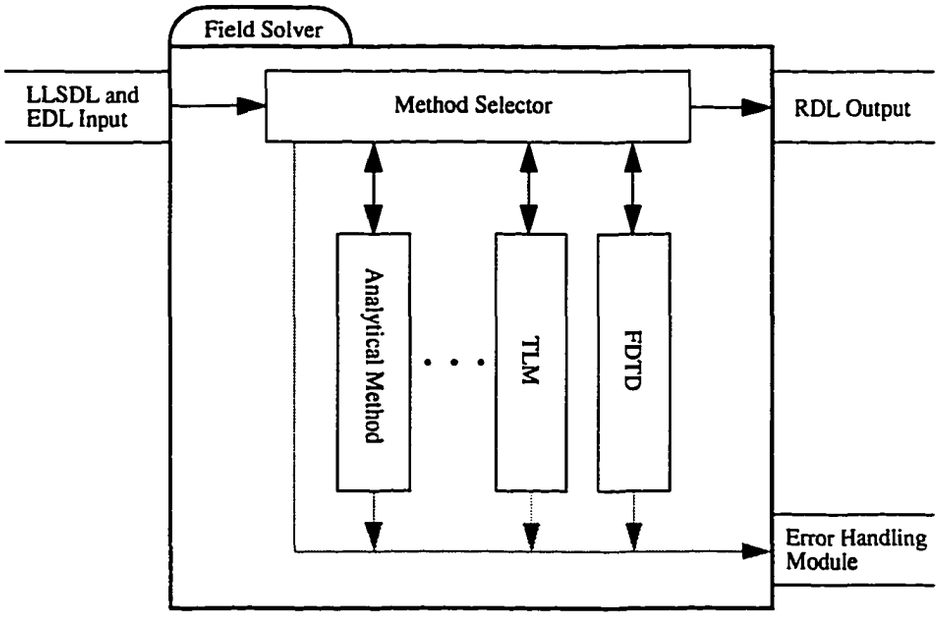


Figure 8.5: A field solver with its various analytical and numerical methods. Because the LLSDL and EDL input can be provided by external programs, an error handling path is provided in case those files contain erroneous data.

8.2.5 The Signal Processor

The signal processor of this simulator must be able to display the computed responses in the time, frequency as well as spatial domains. It must also have some advanced filtering capabilities in these domains so that unwanted signals can be eliminated from the responses. Figure 8.6 shows such a processor with its various internal components.

The output selector in Figure 8.6 allows the user to select the type or types of outputs to be displayed. Once a complete set of RDL input is received, this signal processor should be able to function independently of the field solver. In other words, the signal processor should perform all the required computations with the current RDL input and store the computed result in its internal buffer for interactive viewing and processing. The signal transformation blocks in the figure should contain a null (i.e. no operation, just let the signal pass through with no transformation) and a Fourier transform function, whereas, the signal filtering blocks should contain a null as well as some practical filtering functions.

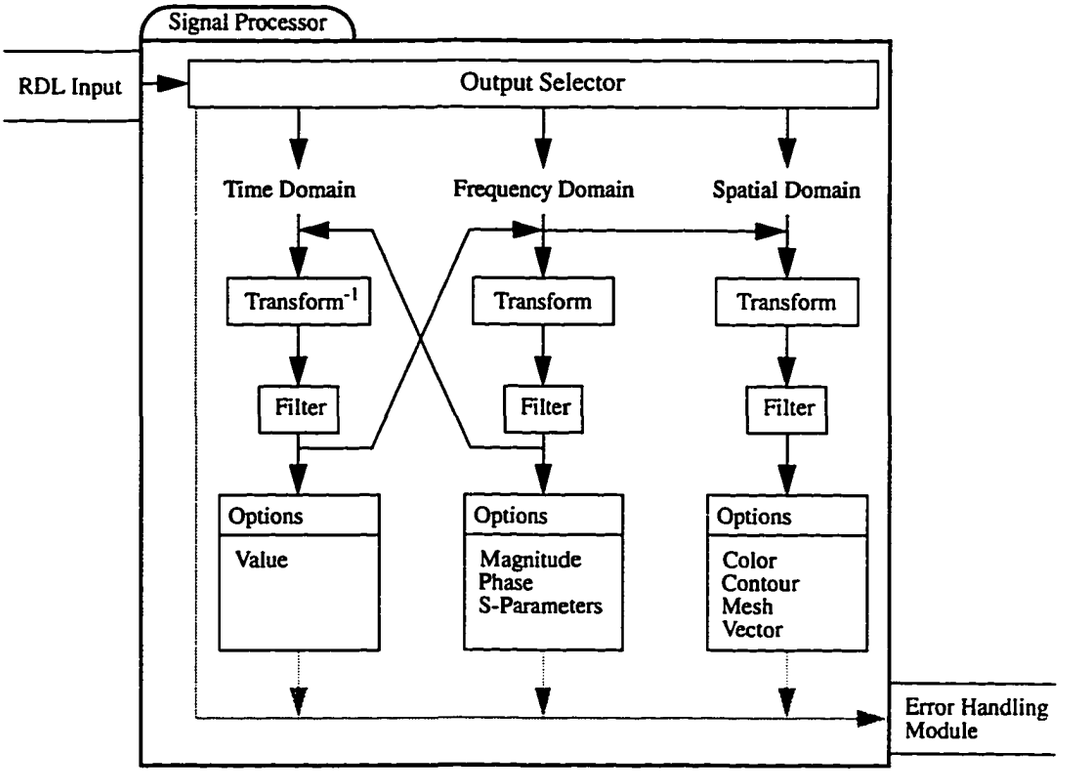


Figure 8.6: A signal processor with its various internal components. The grayed blocks are not physical modules, but they are used as logical tags to group the functionalities of the three different domains together.

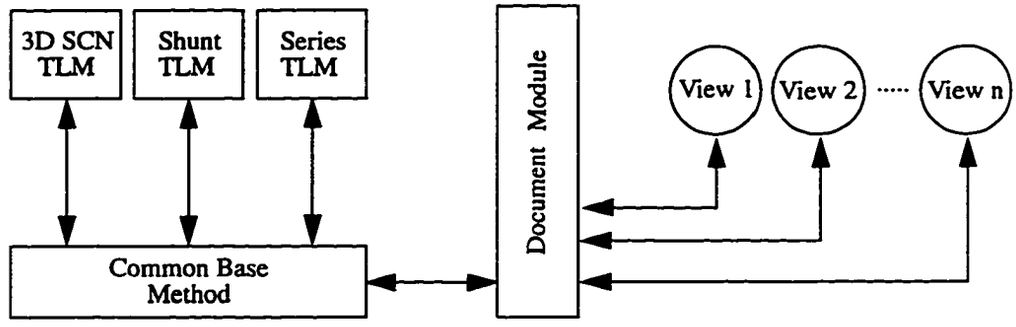


Figure 8.7: The *document-view* programming model used to implement this simulator. The *document* module isolates the base numerical method from the various graphical *view* modules. Therefore, compatible numerical methods can be easily incorporated into the simulator.

8.3 A Prototype Multi-Purpose Electromagnetic Field Simulator

A prototype multi-purpose electromagnetic filed simulation tool based on the framework given in the previous section has been implemented. It incorporates most of the advanced concepts and procedures described in this thesis using the object-oriented programming paradigm [46]. The connections between the graphical interface and the numerical computation modules are carefully designed so that other compatible numerical methods such as finite difference time domain (FDTD), finite difference (FD) and finite elements (FE) methods can be easily integrated into this simulator.

This simulator uses a *document-view* programming model [42]. The *document* part contains C++ objects which encapsulate the data and procedures of the TLM method. The *view* part of the model contains various C++ graphical interface objects to display the geometry of the structure under study as well as its responses to the applied electromagnetic excitations. Because of this clean separation between the document and its associated views, compatible numerical methods can be easily integrated into the document object by simply deriving new methods from a common base method. This idea is depicted in Figure 8.7.

The graphical interface front end of this simulator consists of a main window and five types of subwindows (each subwindow has an associated view object): Editor, Generator, Analyzer, Animator and Johns-Matrix. Figure 8.8 depicts the main control window together with the editor and its associated tool boxes.

The main window of the simulator has a menu system and a button bar which allow users to enter commonly used commands. The other subwindows mimic various kinds of instruments that microwave engineers would normally find

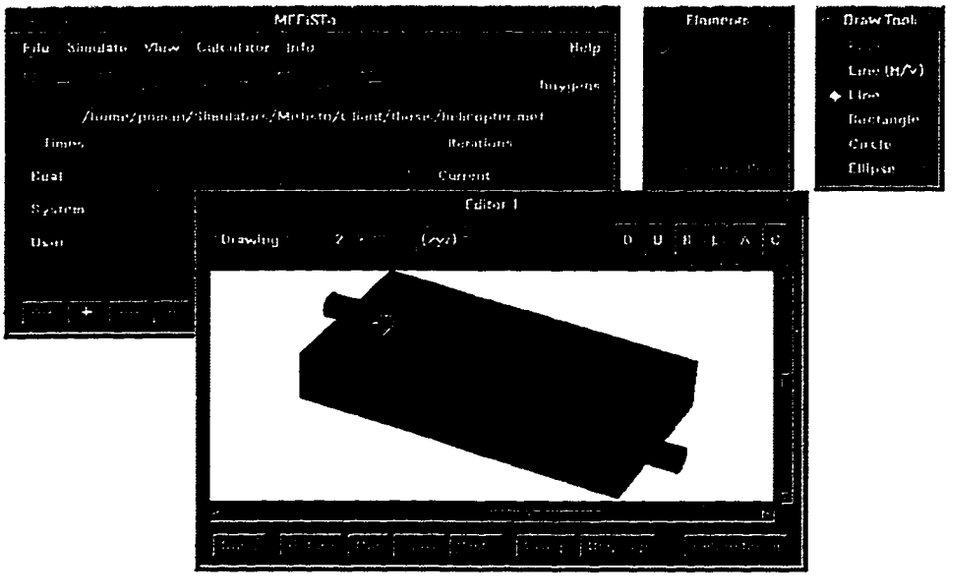


Figure 8.8: The graphical front end of the simulator. The tool boxes on the right of the Editor window allows the user to select the desired drawing element.

in their laboratories, such as drafting table, generator, scope and network analyzer, as well as a hard-to-find instrument — the field animator.

The Editor window mimics the drafting table, allowing users to input the geometry of the structure to be studied via a mouse-based user-friendly drawing mechanism. The tool boxes on the right of the Editor window allow users to select the desired drawing element. Once the geometry is entered, the Editor can discretize the structure to various degrees of coarseness as specified by users. This feature allows problems to be solved with successively finer meshes, and the results can be used in the Richardson extrapolation procedure [43] to estimate the result one would obtain from an infinitely fine mesh.

The Generator window models the signal generator one would find in the microwave laboratory. It allows users to specify the characteristics of the excitation waveform. The Analyzer window models the oscilloscope. It lets users measure the electrical response of the structure. The Animator window (there is nothing like it in the microwave laboratory) allows users to visualize the field evolution in the structure to be studied; it thus enriches the user's perception of electromagnetic wave propagation to an extent rarely achieved by any other tool in the microwave laboratory. The Johns-Matrix window models a high quality matched load. It lets users terminate the structure to be studied with a minimum reflection at the input and output planes. These windows are shown in Figure 8.9.

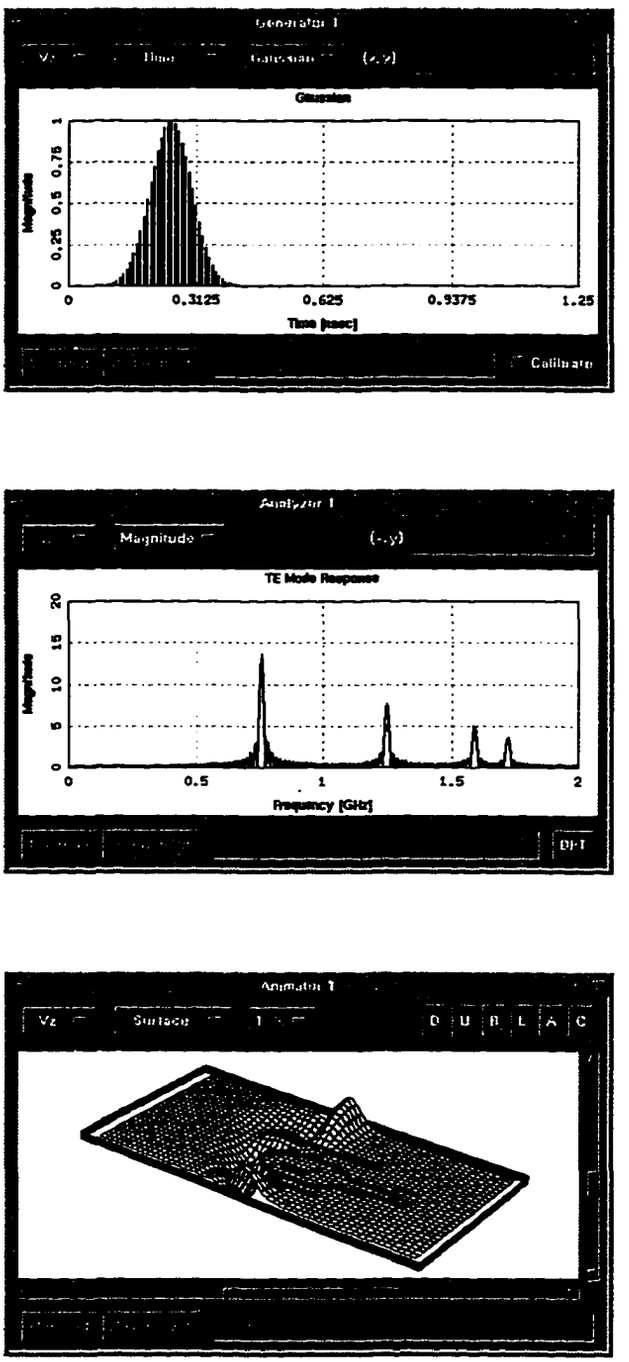


Figure 8.9: The Generator, Analyzer and Animator windows of the simulator. These windows allow users to interactively control the input and output characteristics of the simulator.

8.4 Conclusion

Advanced features and computational techniques for the TLM method have been developed in the previous chapters. A design framework for a multi-purpose electromagnetic simulator as well as a prototype version based on the suggested design framework have been presented in the previous sessions. Figure 8.10 summaries the contributions of this thesis in a graphical representation. This diagram is very similar to Figure 1.1; the major difference between this figure and the previous one is in the grey oval area in which the "EM Simulation Tool" has been replaced by *Mefisto*.

Mefisto, a very good name suggested by Dr. Hoefer, is an acronym for **M**ulti-purpose **E**lectromagnetic **F**ield **S**imulation **To**ol.

There are some advanced features for the TLM method available in the literature. The corner node [15], non-linear active region [24], frequency domain TLM [12] and the alternating TLM [1] are just some of them; these features should be included in the future versions of *Mefisto*. Other numerical methods that are

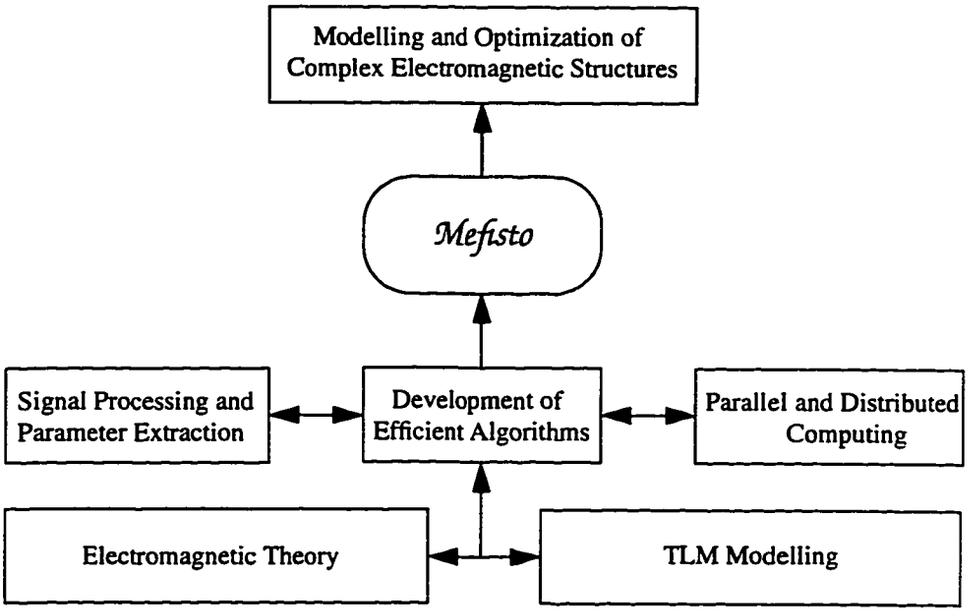


Figure 8.10: Contributions of this thesis to the modelling and optimization of complex electromagnetic structures with TLM.

based on discretization of space and time, such as FDTD, such be included in the future versions as well. The implementation framework of *Mefisto* allows this to be done easily.

More theoretical research is needed in order to increase the modelling capability of *Mefisto*. Since no one numerical method is best for all modelling applications, it is desirable to use a hybrid approach:

- Divide a complex structure into multiple substructures.
- Use the best numerical method to model each substructure.
- Compute the overall characteristic of the structure by combining the characteristics of all the substructures.

Despite the existing version of *Mefisto* does not have all the advanced features mentioned above, it does served its purpose in demonstrating the advanced features and techniques developed in thesis can be easily integrated into a well designed tool. Further testing by independent design practitioners is required to refine the various features of *Mefisto*.

The current version of *Mefisto* runs on HP-UX and IBM AIX, and requires Motif and PEX run-time libraries. Since Windows 95 and NT are gaining wide acceptance in the computing industry, there is a need to port *Mefisto* to Windows 95 and NT by using Microsoft Foundation Class (MFC) and OpenGL libraries. Furthermore, in order for *Mefisto* to be a useful design and optimization tool for microwave and millimeter-wave engineering, *Mefisto* must support third party professional drafting programs, such as AutoCAD, because many engineering drawings for EM circuit components have been created using them. The author believes that the line between research and development should be drawn here. Any further development of *Mefisto* should be done by the microwave CAD industry. However, the research and development of advanced TLM features and algorithms as well as computational techniques will continue be done by researchers and *Mefisto* shall play an important role as a test bed for the new modelling techniques.

APPENDIX A Program Listings

A.1 Pseudo Code for the Discretization of Straight Boundary

```
Bool TLMmesh::AddLineOrthoZ(Real refl,
                            double x1, double y1, double z1,
                            double x2, double y2, double z2)
{
  RealA3D &xw(*xwa), &yw(*ywa);
  RealA3D &xm(*xrc), &ym(*yrc);
  Bool okay=True;

  if (z1>z2) Swap(z1,z2);
  const int k1=(_3D_)?Dz(z1):0, k2=(_3D_)?Dz(z2):1;

  if (y1==y2)
  {
    if (x1>x2) { Swap(x1,x2); Swap(y1,y2); }
    int i,i1=Dx(x1), i2=Dx(x2),j=Dy(y1),J=j+1,k;
    Real yc=Dy(y1);
    for (i=i1; i<i2; i++)
    {
      if (i>=0 && i<x_size && j>=0 && j<=y_size)
      {
        for (k=k1; k<k2; k++)
        {
          yw(i,j,k) = refl;
        }
      }
    }
    if (i1==i2) okay=False;
  }
  else if (x1==x2)
  {
    if (y1>y2) { Swap(x1,x2); Swap(y1,y2); }
    int i=Dx(x1),j,j1=Dy(y1),j2=Dy(y2),k;
    Real xc=Dx(x1);
    for (j=j1; j<j2; j++)
    {
      if (i>=0 && i<=x_size && j>=0 && j<y_size)
      {
        for (k=k1; k<k2; k++)
        {
          xw(i,j,k) = refl;
        }
      }
    }
    if (j1==j2) okay=False;
  }
  else
  {
    x1+=dx*_xcf_; x2+=dx*_xcf_;
    y1-=dy*_ycf_; y2-=dy*_ycf_;
```

```
    if (x1>x2) { Swap(x1,x2); Swap(y1,y2); }
    double m=(y2-y1)/(x2-x1), b=(y1*x2-y2*x1)/(x2-x1);
    int i,j,k, i1=Dx(x1), i2=Dx(x2);
    for (i=i1; i<i2; i++)
    {
      Real y=yfx_Line(b,m,(i+0.5)*dx+xo);
      j = Max(0,Min(y_size,Dy(y)));
      if (i>=0 && i<x_size)
      {
        for (k=k1; k<k2; k++)
        {
          yw(i,j,k) = refl;
        }
      }
    }

    if (y1>y2) { Swap(x1,x2); Swap(y1,y2); }
    m = (x2-x1)/(y2-y1);
    b = (x1*y2-x2*y1)/(y2-y1);
    int j1=Dy(y1), j2=Dy(y2);
    for (j=j1; j<j2; j++)
    {
      Real x=yfx_Line(b,m,(j+0.5)*dy+yo);
      i = Max(0,Min(x_size,Dx(x)));
      if (j>=0 && j<y_size)
      {
        for (k=k1; k<k2; k++)
        {
          xw(i,j,k) = refl;
        }
      }
    }

    if (i1==i2 && j1==j2) okay=False;
  }
  return okay;
}
```

A.2 A Node Buffer Based 2D-TLM Scattering Algorithm

The algorithms for the scatter functions in the following code is given in Eq 7.1 and 7.2 in *Chapter 7 Special Computing Techniques.*

```
//-------------------------------------------------------------
// Mesh::Scatter
//-------------------------------------------------------------
void Mesh::Scatter()
{
  int i,size=node_buffer.Top();
    for (i=0;i<size;i++)
      node_buffer(i)->Scatter();
}
```

```
//-------------------------------------------------------------
// TLMmesh::NodeBuffer::Scatter
//-------------------------------------------------------------
void TLMmesh::NodeBuffer::Scatter()
{
  register int i;
  if (size<=0) return;
  if (buffer->ClassName()==ShuntNode::_classname_)
  {
    register ShuntNode *shunt=(ShuntNode*)buffer;
    register Real mv=shunt->ScatterMv();
    for (i=0; i<size; i++) shunt[i].scatter(mv);
  }
  else if (buffer->ClassName()==SeriesNode::_classname_)
  {
    register SeriesNode *series=(SeriesNode*)buffer;
    register Real mv=series->ScatterMv();
    for (i=0; i<size; i++) series[i].scatter(mv);
  }
  else
  {
    Mesh::NodeBuffer::Scatter();
  }
}
```

APPENDIX B Circuit Files

B.1 Inductive Iris Bandpass Filter OSA90/hope Circuit File-1

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! File: Ind-Iris-Bandpass-Filter-1.ckt
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Expression
 MF = 9;
 NY = 14;          NX = NY*MF;          NZ = 1;
 Y_DIM = 7.112e-3;
 X_DIM = Y_DIM*MF;
 Z_DIM = NZ*Y_DIM/NY;
 ER = 1.0;         UR = 1.0;            FC = 3;
 N_SIM = 8000;     N_F = 141;
 F_MIN = 26;       F_MAX = 40; ! GHz
 ELEM = 1; ! Inductive iris bandpass filter
 F0 = 33;          S11 = 1;             S21 = 2;
 v1 = ? 1  2  6?;
 v2 = ? 1  3  6?;
 v3 = ? 1  5  6?;
 u1 = ?18 21 24?;
 u2 = ?18 23 24?;
 W1 = Y_DIM*floor(v1)*2/NY;
 W2 = Y_DIM*floor(v2)*2/NY;
 W3 = Y_DIM*floor(v3)*2/NY;
 D1 = X_DIM*floor(u1)/NX;
 D2 = X_DIM*floor(u2)/NX;

 Datapipe: SIM FILE="3dtlm-pipe"
     N_INPUT =22 INPUT=(NX,NY,NZ,X_DIM,Y_DIM,Z_DIM,
                        ER,UR,FC,N_SIM,N_F,F_MIN,
                        F_MAX,FREQ,ELEM,F0,S11,W1,
                        W2,W3,D1,D2)
     N_OUTPUT=1 OUTPUT=(MAG_S11);

 Datapipe: SIM FILE=SAME
     N_INPUT =22 INPUT=(NX,NY,NZ,X_DIM,Y_DIM,Z_DIM
                        ER,UR,FC,N_SIM,N_F,F_MIN,
                        F_MAX,FREQ,ELEM,F0,S21,W1,
                        W2,W3,D1,D2)
     N_OUTPUT=1 OUTPUT=(MAG_S21);

 DB_S11 = 20*Log(MAG_S11);
 DB_S21 = 20*Log(MAG_S21);

End
```

B.2 Inductive Iris Bandpass Filter OSA90/hope Circuit File-2

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! File: Ind-Iris-Bandpass-Filter-2.ckt
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Expression
 MF = 9;
 NY = 14;          NX = NY*MF;       NZ = 7;
 Y_DIM = 7.112e-3;
 X_DIM = Y_DIM*MF;
 Z_DIM = NZ*Y_DIM/NY;
 ER = 1.0;         UR = 1.0;         FC = 3;
 N_SIM = 8000;     N_F = 141;
 F_MIN = 26;       F_MAX = 40; ! GHz
 ELEM = 2; ! Inductive iris bandpass filter with
           ! tuning two screws.
 F0 = 33;           S11= 1;          S21= 2;
 v1 = ? 1  2 6?;  v2 = ? 1  3  6?; v3 = ? 1 5 6?;
 u1 = ?18 21 24?; u2 = ?18 23 24?;
 sv1= ? 1  2 6?   sv2= ? 1  2  6?
 W1 = Y_DIM*floor(v1)*2/NY;
 W2 = Y_DIM*floor(v2)*2/NY;
 W3 = Y_DIM*floor(v3)*2/NY;
 D1 = X_DIM*floor(u1)/NX;
 D2 = X_DIM*floor(u2)/NX;
 SW1= Y_DIM*floor(sv1)*2/NY;
 SD1= Z_DIM*floor(sv2)/NZ;

 Datapipe: SIM FILE="3dtlm-pipe"
     N_INPUT =22 INPUT=(NX,NY,NZ,X_DIM,Y_DIM,Z_DIM,
                        ER,UR,FC,N_SIM,N_F,F_MIN,
                        F_MAX,FREQ,ELEM,F0,S11,W1,
                        W2,W3,D1,D2,SW1,SD1)
     N_OUTPUT=1 OUTPUT=(MAG_S11);
 Datapipe: SIM FILE=SAME
     N_INPUT =22 INPUT=(NX,NY,NZ,X_DIM,Y_DIM,Z_DIM
                        ER,UR,FC,N_SIM,N_F,F_MIN,
                        F_MAX,FREQ,ELEM,F0,S21,W1,
                        W2,W3,D1,D2,SW1,SD1)
     N_OUTPUT=1 OUTPUT=(MAG_S21);

 DB_S11 = 20*Log(MAG_S11);
 DB_S21 = 20*Log(MAG_S21);

End
```

Index

Bibliography

Journal and Conference Papers

[1] P. Russer, *On the Field Theoretical Foundation of the Transmission Line Matrix Method*, First International Workshop on Transmission Line Matrix (TLM) Modeling — Theory and Applications, pp.3-12, August 1-3, 1995, Victoria, British Columbia, Canada.

[2] P.P.M. So and W.J.R. Hoefer, *Recent Advances in Transmission Line Matrix Method Electromagnetic Wave Modelling and Visualization*, First International Workshop on Transmission Line Matrix (TLM) Modeling — Theory and Applications, pp.183-186, August 1-3, 1995, Victoria, British Columbia, Canada.

[3] P.P.M. So, C. Eswarappa and W.J.R. Hoefer, *Distributed Parallel TLM Computation and Digital Signal Processing for Electromagnetic Field Modelling*, an invited paper, International Journal of Numerical Modelling — Electronic Networks, Devices and Fields, vol. 8, no. 3/4, May-August 1995, pp 169-185, John Wiley & Sons. Inc..

[4] H. Jin and R. Vahldieck, *Direct Derivations of TLM Symmetrical Condensed Node and Hybrid Symmetrical Condensed Node from Maxwell's Equations Using Centered Differencing and Averaging*, IEEE Trans. Microwave Theory Tech., vol. MTT-42, no. 12, pp. 2554-2561, December 1994.

[5] C. Eswarappa and W.J.R. Hoefer, *Characterization of Microwave Structures using ARMA System Identification and TLM Method*, IEEE AP-S Digest, vol. 2, pp. 1124-1127, Seattle, WA, June 19-24, 1994.

[6] C. Eswarappa, P.P.M. So, and W.J.R. Hoefer, *Efficient Field-Based CAD of Microwave Circuits on Massively Parallel Processor Computer using TLM and Prony's Methods*, IEEE MTT-S, vol. 3, pp. 1531-1534, May 23-27, 1994, San Diego, CA

[7] M. Righi and W.J.R. Hoefer, *Efficient 3D-SCN-TLM Diakoptics for Waveguide Components*, IEEE MTT-S, vol. 1, pp. 27-30, May 23-27, 1994, San Diego, California.

[8] P.P.M. So and W.J.R. Hoefer, *Optimization of Microwave Structures using a Parallel TLM Module*, 10^{th} Annual Review of Progress in Applied Computational Electromagnetics Digest, pp. 546-553, March 21-26, 1994, Monterey, California.

[9] P.P.M. So and W.J.R. Hoefer, *A New Look at the 3D Condensed Node TLM Scattering*, IEEE MTT-S, pp. 1443-1446, June 14-18, 1993, Atlanta, Georgia.

[10] W. Menzel, F. Alessandri, M. Mongiardo, R. Sorrentino, C. Eswarappa, P.P.M. So and W.J.R. Hoefer, *Analysis of a Millimeter-Wave Filter Using Transmission Lines Matrix and Mode Matching Methods and Comparison with the Measurements*, 9^{th} Annual Review of Progress in Applied Computational Electromagnetics, pp.289-296, March 1993, Monterey, California.

[11] P.P.M. So, C. Eswarappa and W.J.R. Hoefer, *Transmission Line Matrix Method on Massively Parallel Processor Computers*, 9^{th} Annual Review of Progress in Applied Computational Electromagnetics Digest, pp. 467-474, March 22-26, 1993, Monterey, California.

[12] H. Jin and R. Vahldieck, *The Frequency Domain Transmission Line Matrix Method — A New Concept*, IEEE Trans. Microwave Theory Tech., vol. MTT-40, no. 12, pp. 2207-2218, December 1992.

[13] M. Forest and W.J.R. Hoefer, *TLM Synthesis of Microwave Structures Using Time Reversal*, IEEE MTT-S Digest, vol. 2, pp. 779-782, June 1-5, 1992, Albuquerque, New Mexico.

[14] W. Kuempel and I. Wolff, *Digital signal Processing of Time-Domain Field Simulation Using the System Identification Method*, IEEE MTT-S Digest, vol. 2, pp. 793-796, June 1-5, 1992, Albuquerque, New Mexico.

[15] U. Müller, P.P.M. So and W.J.R. Hoefer, *The Compensation of Coarseness Error in 2D TLM Modeling of Microwave Structures*, IEEE MTT-S, vol. 1, pp. 373-376, June 1-5, 1992, Albuquerque, New Mexico.

[16] Z. Chen, M.M. Ney and W.J.R. Hoefer, *A New Finite Difference Time Domain Formulation and Its Equivalence with the TLM Symmetrical Condensed Node*, IEEE Trans. Microwave Theory Tech., vol. MTT-39, no. 12, pp. 2160-2169, December 1991.

[17] W. Ko and R. Mittra, *A Combination of FD-TD and Prony Methods for Analyzing Microwave Integrated Circuits*, IEEE Trans. Microwave Theory Tech., vol. MTT-39, no. 12, pp. 2176-2181, December 1991.

[18] P. Russer, *Overview Over Discrete Time Domain Methods in Electromagnetic Field Computation*, International Workshop on Discrete Time Domain Modeling of Electromagnetic Field and Network, Germany IEEE MTT/AP join chapter and CAS chapter, Munich, Oct 24-25, 1991.

[19] M. Celuch-Marcysiak and W.K. Gwarek, *Formal Equivalence and Efficiency Comparison of the FDTD, TLM and SN Methods in Application to Microwave CAD Programs*, 21st European Microwave Conference Digest, pp. 199-204, September 1991, Stuttgart, Germany.

[20] R. Sorrentino, P.P.M. So and W.J.R. Hoefer, *Numerical Microwave Synthesis by Inversion of the TLM Process*, 21st European Microwave Conference Digest, pp. 1273-1277, September 1991, Stuttgart, Germany.

[21] C.E. Tong and Y. Fujino, *An Efficient Algorithm for Transmission Line Matrix Analysis of Electromagnetic Problems Using the Symmetrical Condensed Node*, IEEE Trans. Microwave Theory and Tech. vol. MTT-39, no. 8, pp. 1420-1424, August 1991.

[22] P.P.M. So and W.J.R. Hoefer, *3D-TLM Time Domain Electromagnetic Wave Simulator for Microwave Circuit Modelling*, IEEE MTT-S, vol. 2, pp. 631-634, June 10-14, 1991, Boston, Massachusetts.

[23] N.R.S. Simons and E. Bridges, *Equivalence of Propagation Characteristics for the Transmission Line Matrix and Finite Difference Time Domain Methods in Two Dimensions*, IEEE Trans. Microwave Theory and Tech. vol. MTT-39, no. 2, pp. 354-457, February 1991.

[24] P. Russer, P.P.M. So and W.J.R. Hoefer, *Modelling of Nonlinear Active Region in TLM*, IEEE Microwave and Guided Wave Letters, January 1991.

[25] J.L. Dubard, D. Pompei, J.L. Roux, A. Papiernik, *Characterization of microstrip antennas using the TLM simulation associated with a Prony-Pisarenko method*, International Journal of Numerical modelling, Electronics Networks, Devices and Fields, vol. 3, no. 4, pp. 269-285, December 1990.

[26] Eswarappa, P.P.M. So and W.J.R. Hoefer, *New Procedure for 2D and 3D Microwave Circuit Analysis with the TLM Method*, IEEE MTT-S, vol 2, pp. 661-664, May 8-10, 1990, Dallas, Texas.

[27] P. Naylor and R.A. Desai, *New Three-Dimensional Symmetrical Condensed Lossy Node for Solution of Electromagnetic Wave Problems by TLM*, IEE Electronics Letters, vol. 26, no. 7, pp. 492-494, March 1990.

[28] P.P.M. So, Eswarappa and W.J.R. Hoefer, *A Two-dimensional Transmission Line Matrix Microwave Field Simulator Using New Concepts and Procedures*, IEEE MTT vol. 37, no 12, pp. 1877-1884, December 1989.

[29] P.P.M. So and W.J.R. Hoefer, *A General Planar Circuit Simulator Based on Two-dimensional TLM Method*, IEEE MTT-S, vol. 1, pp. 343-346, June 13-15, 1989, Long Beach, California.

[30] P. Strobach, *Recursive Covariance Ladder Algorithms for ARMA System Identification*, IEEE trans. Acoustics, Speech and Signal Processing, vol. 36, no. 4, pp. 560-580, April 1988.

[31] P.B. Johns, *A Symmetrical Condensed Node for the TLM Method*, IEEE Trans. Microwave Theory and Tech. vol. MTT-35, no. 4, pp. 370-377, April 1987.

[32] R. Allen, A. Mallik and P.B. Johns, *Numerical Results for the Symmetrical Condensed TLM Node*, IEEE Trans. Microwave Theory and Tech. vol. MTT-35, no. 4, pp. 378-382, April 1987.

[33] P.B. Johns, *On the Relationship between TLM and Finite Difference Methods for Maxwell's Equations*, IEEE Trans. Microwave Theory and Tech. vol. MTT-35, pp. 60-61, January 1987.

[34] D.A. Al-Mukhtar and J.E. Sitch, *Transmission Line Matrix Method with Irregularly Graded Space*, IEE Proc. vol. 128, Pt. H, no. 6, pp. 299-305, December 1981.

[35] Y.C. Shih and W.J.R. Hoefer, Dominant and Second-Order Mode Cutoff Frequencies in Fin Lines Calculated with a Two-Dimensional TLM Program, IEEE Trans. Microwave Theory and Tech. vol. MTT-28, no. 12, pp.1443-1448, December 1980.

[36] I. Getreu, *Modelling the Bipolar Transistor, Part II*, Electronics, pp. 71-75, October 31, 1974.

[37] P.B. Johns, Application of the Transmission Line Matrix Method to Homogeneous Waveguides of Arbitrary Cross-Section, Proc. IEE, vol. 119, no. 8, pp.1086-1091, August 1972.

[38] J.R. Whinnery, C. Concordia, W. Ridgway, and G. Kron, *Network Analyzer Studies of Electromagnetic Cavity Resonators*, IEE Proc. vol. 32, pp. 360-367, June 1944.

[39] J.R. Whinnery, and S. Ramo, *A New Approach to the Solution of High Frequency Field Problems*, Proc. I.R.E., vol. 32, pp. 284-288, 1944.

[40] G. Kron, *Equivalent Circuit of the Field Equations of Maxwell - I*, Proc. I.R.E., vol. 32, pp. 289-299, May 1944.

Other References

[41] P.P.M. So, C. Eswarappa and W.J.R. Hoefer, *Massively Parallel and Distributed Computing and Digital Signal Processing for TLM Electromagnetic Field Modeling*, a workshop paper, IEEE APS, June 1994, Seattle, Washington.

[42] *Borland ObjectWindows for C++ Version 2.0 Programmer's Guide*, Borland International Inc., 1993.

[43] R.C. Booton, Jr., *Computational Methods for Electromagnetics and Microwaves*, ISBN 0-471-52804-8 John Wiley & Sons, 1992.

[44] W.J.R. Hoefer, *ELEC621 Course Notes*, University of Victoria, September 1992.

[45] J.S. Nielsen, *TLM Analysis of Microwave and Millimeter Wave Structures with Embedded Nonlinear Devices*, Ph. D. Dissertation, University of Ottawa, 1992.

[46] B. Stroustrup, *The C++ Programming Language*, second edition, ISBN 0-201-53992-6 Addison-Wiley, 1992.

[47] *DECmpp Sx V1.1 Programming Language Reference Manual*, Digital Equipment Corporation, 1992.

[48] W.J.R. Hoefer and P.P.M. So, *The Electromagnetic Wave Simulator — A Dynamic Visual Electromagnetics Laboratory based on the Two-Dimensional TLM Method*, John Wiley & Sons Inc., 1991.

[49] J.J. Valley, *UNIX Programmer's Reference*, Chapter 11, pp. 243-264, 1991, Que., ISBN 0-88022-536-X

[50] C. Huygens, *Treatise of Light*, Pierre Vander Aa, Leiden, 1690, reprinted by Epsilon Uitgaven, Utrecht, 1990.

[51] W.J.R. Hoefer, *The Transmission Line Matrix (TLM) Method*, Chapter 8 of Numerical Techniques for Microwave and Millimeter-Wave Passive Structures, edited by Tatsuo Itoh, John Wiley & Sons Inc., 1989.

[52] N. Marcuvitz, *Waveguide Handbook*, pp. 66-72, IEE, 1986 (first published in 1951 by the McGraw-Hill Inc.).