

1994

Computational aspects of electromagnetic NDE phenomena

Ibrahim Mohamed Elshafiey
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Electrical and Electronics Commons](#)

Recommended Citation

Elshafiey, Ibrahim Mohamed, "Computational aspects of electromagnetic NDE phenomena " (1994). *Retrospective Theses and Dissertations*. 10697.

<https://lib.dr.iastate.edu/rtd/10697>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600



Order Number 9503550

Computational aspects of electromagnetic NDE phenomena

Elshafey, Ibrahim Mohamed, Ph.D.

Iowa State University, 1994

U·M·I

300 N. Zeeb Rd.
Ann Arbor, MI 48106



Computational aspects of electromagnetic NDE phenomena

by

Ibrahim Mohamed Elshafey

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Department: Electrical Engineering and Computer Engineering
Major: Electrical Engineering (Electromagnetics)

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa
1994

DEDICATION

To the ones I have not seen in this world, and I hope that God will gather me with them in the abode of His mercy. *To my grandfathers:*

Ibrahim Elshafiey & Fouad Khalil

TABLE OF CONTENTS

ABSTRACT	xiii
CHAPTER 1. INTRODUCTION	1
CHAPTER 2. DIFFERENTIAL EQUATION APPROACH FOR SOLVING ELECTROMAGNETIC NDE APPLICATIONS	11
Finite difference method	12
Finite difference time domain modeling	12
Absorbing boundary conditions	14
Finite element method	19
Weighted residuals	21
Variational principle	23
Finite element analysis techniques for 2D problems in EM NDE	24
Finite element methods for solving 3D problems	31
CHAPTER 3. INTEGRAL EQUATION APPROACH FOR SOLV- ING ELECTROMAGNETIC NDE APPLICATIONS	36
Surface integral representation	37
T-matrix method	38
Volume integral representation	39
Green's function	42

CHAPTER 4. APPLICATION OF HYBRID DIFFERENTIAL	
METHODS FOR HANDLING OPEN BOUNDARY PROBLEMS	47
Finite element hybrid model for 2D geometries	48
Electromagnetic scattering from objects embedded in multilayered media .	52
Stationary phase method	65
CHAPTER 5. APPLICATION OF INTEGRAL MOMENT METHOD	
WITH REDUCED STORAGE	68
Conjugate gradient method	74
Conjugate gradient fast Fourier transform	76
A scatterer above a perfect conducting surface	79
A scatterer embedded in multilayered media	82
Secondary Green's function	82
CHAPTER 6. PARALLEL COMPUTING FOR LARGE SIZE PROB-	
LEMS	90
Parallel computers architecture	94
Hypercube architecture	97
Communication in NCUBE2	100
CHAPTER 7. IMPLEMENTATION OF 3D FINITE ELEMENT	
MODEL ON PARALLEL COMPUTERS FOR EDDY CURRENT	
NDE POD CALCULATIONS	102
Finite element modeling of eddy current NDE of 3D objects	103
Probability of detection (POD) models	105
POD models for eddy current NDE	106
Implementation of finite element eddy current model on a hypercube . . .	108

Incomplete Cholesky Conjugate Gradient Method (ICCG)	109
Gaussian elimination	111
LU decomposition	114
CHAPTER 8. NEURAL COMPUTING APPROACH FOR SOLV-	
ING INVERSE PROBLEM IN ELECTROMAGNETIC NDE . .	117
Solution of inverse problems in electromagnetics using Hopfield neural net-	
works	118
Iterative inversion techniques	120
Hopfield neural network	122
Inverse problem in electromagnetics	131
Problem formulation	134
Simulation algorithm	136
Application of weight space mapping and modular architectures for incre-	
mental learning networks	138
Modified steepest descent and error functions	142
Back-propagation algorithm using modified steepest descent procedure	151
Modular network architecture for incremental learning	152
Network architecture	153
Simulation results	155
CHAPTER 9. CONCLUSIONS	173
BIBLIOGRAPHY	180
ACKNOWLEDGMENTS	190

LIST OF FIGURES

Figure 1.1: A typical NDE system	3
Figure 2.1: Finite difference lattice truncation	15
Figure 2.2: Finite element solution domain	21
Figure 2.3: Triangular element	27
Figure 2.4: A simple mesh	30
Figure 2.5: Edge element	32
Figure 3.1: Volume equivalent current	40
Figure 3.2: Green's function using current derivatives	44
Figure 4.1: Interior and exterior regions	49
Figure 4.2: Discretization of the internal region	50
Figure 4.3: Bistatic width for conducting cylinders	53
Figure 4.4: Bistatic width for dielectric cylinder	53
Figure 4.5: Bistatic width for a dielectric shell with a TM polarized inci- dent plane wave	54
Figure 4.6: Bistatic width for a dielectric shell TE polarization	55

Figure 4.7:	Comparing the bistatic width of a dielectric shell on $\lambda/2$ distance from a line source with the bistatic width with a plane wave excitation	56
Figure 4.8:	Comparing the bistatic width for a dielectric shell on 2λ distance from a line source with the bistatic width with a plane wave excitation	57
Figure 4.9:	Half a shell with a plane TM polarized wave	58
Figure 4.10:	Bistatic width for a solid cylinder with a TM polarized plane wave	59
Figure 4.11:	Bistatic width for half a solid cylinder with a TM polarized plane wave. Incident angle is π	60
Figure 4.12:	Bistatic width for half a solid cylinder with a TM polarized plane wave. Incident angle is $3\pi/4$	61
Figure 4.13:	Propagation of a plane wave in multilayered media	63
Figure 4.14:	Getting the field at point P_0 due to a line source at P_1	66
Figure 4.15:	Amplitude of normalized surface current vs. angle ϕ	67
Figure 5.1:	Volume discretization of the scatterer	69
Figure 5.2:	One dimensional illustration of Green's function G and electric equivalent current J sampled at the centers of elements	72
Figure 5.3:	Repeated form of one dimensional Green's function G and electric equivalent current J	73
Figure 5.4:	Comparison of the numerical and exact solutions of the bistatic radar cross section of a sphere ($ka=0.9$ and refractive index $N=2.105$)	78

Figure 5.5:	An object on top of a perfectly conducting surface, showing images of a vertical and a horizontal current components . . .	80
Figure 5.6:	One dimensional illustration of the object equivalent current and Green's function of the image elements	83
Figure 5.7:	Repeated form of one dimensional equivalent current and Green's function of the image elements	84
Figure 5.8:	Bistatic radar cross section of a sphere ($ka=0.9, N=2.105$) on a 1λ distance from a perfectly conducting surface.	85
Figure 5.9:	Saddle point	87
Figure 5.10:	Error reduction in integration	89
Figure 6.1:	Von Neumann machine	91
Figure 6.2:	Typical speedup curve	93
Figure 6.3:	Parallelizing the β part of a canonical scalable problem . . .	95
Figure 6.4:	Effect of the sequential part β on the speedup for fixed size problem	96
Figure 6.5:	Effect of the sequential part β on the speedup for scalable problem	96
Figure 6.6:	Hypercube architecture	99
Figure 7.1:	Probability density function for a signal in the presence of a flaw and without a flaw	106
Figure 7.2:	Sources of random variation in eddy current testing	107
Figure 7.3:	Speedup for LU decomposition algorithm	116
Figure 8.1:	Hopfield neural network	124

Figure 8.2:	Linear programming network	128
Figure 8.3:	Neural network with feedback circuit	130
Figure 8.4:	Coaxial multilayered media	132
Figure 8.5:	True distribution of relative permittivity and loss tangent . .	138
Figure 8.6:	First, and second iterations	139
Figure 8.7:	Third iteration	139
Figure 8.8:	First, and second iterations using amplitude of measurements	140
Figure 8.9:	Third iteration using amplitude of measurements	140
Figure 8.10:	Multilayer perceptron	143
Figure 8.11:	Near circular energy function	145
Figure 8.12:	Trough energy function	145
Figure 8.13:	Error function in (8.68) [$\delta_i^r = [O_i^r(1 - O_i^r) + \zeta](d_i - O_i^r)$], $\zeta = 0$	149
Figure 8.14:	Error function in (8.68), $\zeta = 0.2$	149
Figure 8.15:	Error function in (8.64) [$\delta_i^r = \tanh^{-1}(d_i - O_i^r)$]	149
Figure 8.16:	Error function in (8.69) [$\delta_i^r = O_i^r(1 - O_i^r) \tanh^{-1}(d_i - O_i^r) +$ $\zeta(d_i - O_i^r)$], $\zeta = 0$	150
Figure 8.17:	Error function in (8.69), $\zeta = 0.2$	150
Figure 8.18:	Modular network	154
Figure 8.19:	Typical eddy current trajectories for the four classes under consideration	157
Figure 8.20:	Progress of error with iteration until full recognition of pat- terns for regular back-propagation and MSD algorithms using error function in (8.68).	159

Figure 8.21: Progress of error with iteration until full recognition of patterns for regular back-propagation and MSD algorithms using error function in (8.69).	160
Figure 8.22: Progress of error with iteration until full recognition of patterns for regular back-propagation and MSD algorithms using error function in (8.68).	161
Figure 8.23: Progress of error with iteration until full recognition of patterns for regular back-propagation and MSD algorithms using error function in (8.69).	162
Figure 8.24: Progress of error with iteration until full recognition of patterns for regular back-propagation and MSD algorithms using error function in (8.68).	163
Figure 8.25: Progress of error with iteration until full recognition of patterns for regular back-propagation and MSD algorithms using error function in (8.69).	164
Figure 8.26: Comparing performance of a modular architecture network with a regular network for error function in (8.68).	166
Figure 8.27: Comparing performance of a modular architecture network with a regular network for error function in (8.69).	167
Figure 8.28: Comparing performance of a modular architecture network with a regular network using MSD algorithm.	168
Figure 8.29: Comparing performance of a modular architecture network with a regular network using error function in (8.68).	169

Figure 8.30: Comparing performance of a modular architecture network
with a regular network using error function in (8.69). 170

Figure 8.31: Comparing performance of a modular architecture network
with a regular network using MSD algorithm. 171

Figure 8.32: Comparing performance of a modular architecture network
with a regular network for error function in (8.68). 172

LIST OF TABLES

Table 7.1:	Run time with varying number of processors	115
Table 8.1:	Performance of training using regular back-propagation for various choice of error functions for Example 3	165
Table 8.2:	Performance of training using modified steepest descent for various choice of error functions for Example 3	165

ABSTRACT

The development of theoretical models that characterize various physical phenomena is extremely crucial in all engineering disciplines. In nondestructive evaluation (NDE), theoretical models are used extensively to understand the physics of material/energy interaction, optimize experimental design parameters and solve the inverse problem of defect characterization. This dissertation describes methods for developing computational models for electromagnetic NDE applications. Two broad classes of issues that are addressed in this dissertation are related to (i) problem formulation and (ii) implementation on computers.

The two main approaches for solving physical problems in NDE are the differential and integral equations. The relative advantages and disadvantages of the two approaches are illustrated and models are developed to simulate electromagnetic scattering from objects or inhomogeneities embedded in multilayered media which is applicable in many NDE problems. The low storage advantage of the differential approach and the finite solution domain feature of the integral approach are exploited. Hybrid techniques and other efficient modeling techniques are presented to minimize the storage requirements for both approaches.

The second issue of computational models is the computational resources required for implementation. Implementations on conventional sequential computers,

parallel architecture machines and more recent neural computers are presented. An example which requires the use of massive parallel computing is given where a probability of detection model is built for eddy current testing of 3D objects. The POD model based on the finite element formulation is implemented on an NCUBE parallel computer . The linear system of equations is solved using direct and iterative methods. The implementations are designed to minimize the interprocessor communication and optimize the number of simultaneous model runs to obtain a maximum effective speedup.

Another form of parallel computing is the more recent neurocomputer which depends on building an artificial neural network composed of numerous simple neurons. Two classes of neural networks have been used to solve electromagnetic NDE inverse problems. The first approach depends on a direct solution of the governing integral equation and is done using a Hopfield type neural network. Design of the network structure and parameters is presented. The second approach depends on developing a mathematical transform between the input and output space of the problem. A multilayered perceptron type neural network is invoked for this implementation. The network is augmented to build an incremental learning network which is motivated by the dynamic and modular features of the human brain.

CHAPTER 1. INTRODUCTION

Electromagnetic fields constitute a source of energy that is perhaps the most extensively used in a wide range of applications in everyday life. Examples of such applications range from electric motors and communication systems to geophysical exploration, medical diagnosis, military surveillance and nondestructive evaluation (NDE). Classical electromagnetic field theory was the result of the pioneering efforts of Faraday, Ampere, Gauss, Lenz, Coulomb, Volta in the 19'th century [1]. However, advances in more sophisticated electromagnetic modeling and applications have relied largely on the developments of computational models for electromagnetics.

Since the 1950s the power and availability of computers have been growing rapidly. This exhorted the research in computational methods in electromagnetics. Various computational techniques such as finite difference, finite element and integral equations and moment methods have been applied for different applications.

The focus of this dissertation is related to the development of computational models for electromagnetic NDE. A major motivation for focusing on NDE is its multidisciplinary nature. Computational models developed for NDE applications will therefore be applicable to a variety of disciplines such as geophysics, remote sensing etc. The field of NDE is of crucial importance in many industrial applications for determining material properties and evaluating uniformity and homogeneity of test

samples. NDE is also important in increasing the reliability of industrial products and ensuring the safety while these products are used. Another compelling reason for the focus of attention on NDE technology is its economical impact. For example, the total value of bridges in the United States is estimated to be \$500 billion, and bridges valued about \$1.6 billion need to be replaced [2]. The inspection of aging aircraft and nuclear power plants are other multibillion dollar industries where NDE plays a critical role.

An illustration of a typical NDE system is shown in Figure 1.1. An appropriate source of energy is launched through a transmitting transducer into the specimen under test. The interaction between the source energy and the specimen is recorded and the received signals are processed and then fed into an inverse model to predict the parameters and condition of the test specimen. The results of the inverse model enables industries to make decisions about either acceptance or rejection of the test specimen. Decision making is undertaken by specialists or using automated models based on established comprehensive laboratory correlation between specimen characteristics and nondestructive test results.

The most challenging component in the design an NDE system is the inverse model. This is related to the difficulties associated with the solution of inverse problems in general, such as nonuniqueness and ill-posedness of the solution. A variety of approaches has been developed over the decades for addressing these problems by restoration of stability of the solution through regularization.

One of the most important tools in developing an inverse model and designing an optimal nondestructive test is the availability of an accurate and efficient forward model. The forward model is required to understand the nature of the problem,

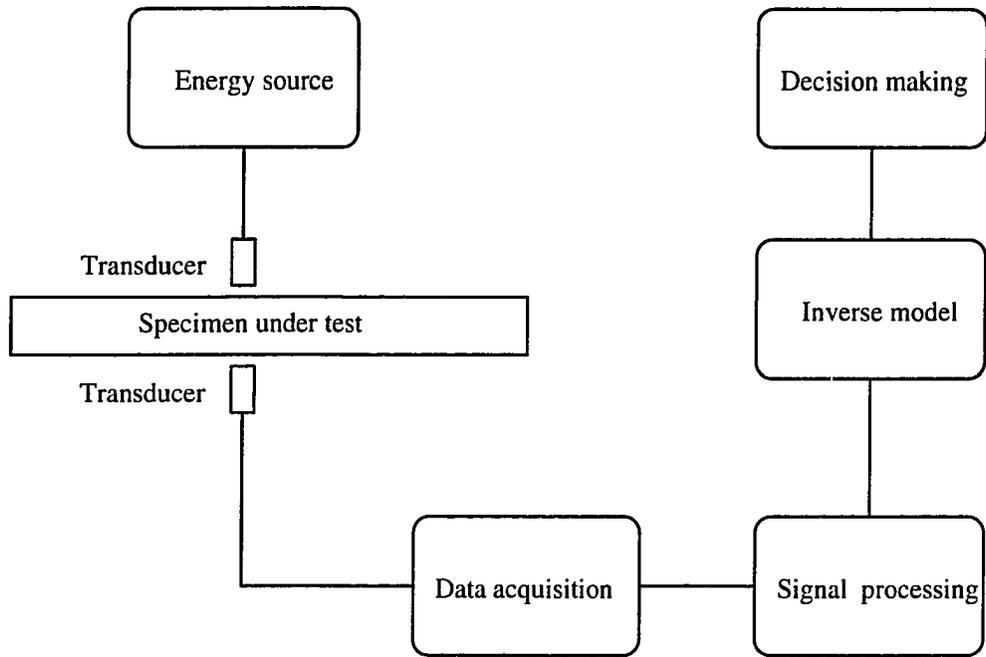


Figure 1.1: A typical NDE system

and determine the constraints that need to be imposed to regularize the solution of the inverse model. The forward model can also be used to create a database to be searched for obtaining the solution of the inverse problem. The development a forward model involves the solution of Maxwell's equations subject to appropriate boundary conditions. However, these equations, spanning the entire range of electromagnetic spectrum, are too general and require constraints to be added according to the problem type and the operating frequency range.

Operating frequency ranges of NDE applications cover the entire electromagnetic spectrum. Techniques employing the static ($\omega = 0$) operation, such as the magnetic flux leakage, and the quasi-static frequency range such as eddy current methods are more commonly used in industry than higher frequency wave phenomenon methods [3, 4, 5]. However, the attention is being focused on the other end of the spectrum. Examples include application of microwave imaging techniques in inspecting civil structures [7]. At even higher frequency ranges, thermal waves are being used in characterizing coating adhesion [8], optical methods are used in evaluating concrete and composite materials [9, 10], and x-ray techniques are used in tomographical reconstruction of defects and in assessing residual stresses and crystalline texture of metal alloys [11]. Hybrid techniques that operate at both high and low frequency ranges are also used as in the case of magneto-optic/Eddy current imaging method [12].

The wide range of EM NDE applications is also reflected in the computational techniques used in modeling these applications. Practitioners tend to specialize in specific applications depending on the problem type, the operating frequency range and more importantly on the available computing resources. An obvious question

that arises is: what is the best computational method for a particular problem? [13]. This was clear in panel discussions of the most recent COMPUMAG conference which focuses on the subject of computation of electromagnetic fields. One of the questions forwarded in the supercomputing panel discussion was addressed toward the direction of the future. The question reflects some fears that rapid developments in computer resources might render some of the modeling techniques obsolete, after the investment of large amounts of time and money in the development of these techniques.

The motivation of this research is to explore the capabilities of the state of the art computing machines as well as computing algorithms for the development of a class of forward and inverse models for problems encountered in EM NDE applications. This objective is accomplished by considering several factors such as

- optimizing the computational algorithms to obtain minimum possible memory storage and time requirements.
- developing hybrid techniques that exploit the advantages of several techniques.
- enhancing the performance by using the massive computing power of parallel architecture machines for achieving reduction in time requirements and increase in efficiency.
- investigating the feasibility of using neural computing techniques for solving inverse problems.

It is essential that these factors be considered since conventional strategies for solving such problems are clearly inadequate.

Applications considered in this dissertation are in the quasi-static and microwave range of operation for modeling eddy current and microwave scattering phenomena.

Various computing resources are also considered in developing the models, including sequential computers, parallel computers and neural computers.

The role of parallel computers is now rapidly increasing due to three main reasons: a decrease in hardware cost, continuous increase in problems size and complexity, and a decline in the growth rate of sequential computing power. It has been estimated that the growth rate per decade of computing power of a single processor has declined from a factor of 100 in the 1960s to 10 in 1970s [14]. At present, the design of computer hardware is reaching some basic physical barriers such as the speed of communication being limited by the speed of light. The dominance of the von Neumann machine with a single processor is therefore predicted to decline and parallel architectures are expected to be the future of computing [15]. Parallel computers have proved to be more economical for handling large size problems in terms of both cost and time requirements.

A novel form of parallel computer that is becoming increasingly popular and may represent an inexpensive alternative to conventional parallel architectures is the neurocomputer. A neurocomputer depends on building an artificial neural network composed of numerous simple building blocks or *neurons*. This architecture has emerged as an outcome of the desire to mimic the human brain. Neural networks are characterized by their parallel distributed architecture with a high degree of interconnectivity and feedback [16]. A significant advantage of neural networks is that, with a large number of neurons, a neural network will continue also to function properly even in the case of failure of a few processing elements, in contrast to digital computers which are typically not fault tolerant.

In addition to choosing an appropriate computing environment, EM modeling

requires the optimization of the computing techniques. The choice of a specific technique depends on the governing differential equation such as the elliptic, parabolic and hyperbolic partial differential equations that describe the static, quasi-static and wave ranges of operation, respectively. Formulation of these equations and their inversion techniques are found in [17, 18]. In general, approaches for formulation are categorized into two major categories: differential equation approaches and integral equation approaches. The differences between these two approaches lie in the choice of the problem unknowns and hence the domain of the solution. This, in turn, is reflected in the computing resource requirements. A survey of numerical methods in field calculation is found in [6, 5]. A comparative study of a differential approach that depends on the finite element method and an integral approach which depends on the boundary element method are given in [19].

In the differential equation approach, the formulation is given in terms of the fields which are defined on the whole space. This approach is in general more suitable for problems with natural boundaries such as waveguides. In the case of low frequency applications governed by diffusion equations, such as eddy current NDE, the values of the field vanish in the far field and discretization can be restricted to a finite domain around the source.

The advantage of the differential approach lies essentially in its low storage requirements. This advantage is however counteracted by the need for a large solution domain in applications governed by the wave equation where the Sommerfeld radiation condition has to be satisfied at the infinite boundary. This dissertation presents appropriate techniques for keeping the discretization domain finite and coupling this domain to the exterior infinite space.

The integral equation formulation is based on the moment method. The unknowns —mostly electric and magnetic currents — are restricted to a small domain of the source, such as the surface or the volume of a scatterer. However, the formulation leads to a linear system of equations characterized by a dense matrix. In most realistic applications the required number of discretization elements is usually high which results in large storage requirements. Efficient modeling techniques for minimizing storage requirements to limits of commonly available computers are presented.

Chapter 2 presents the differential approach for modeling electromagnetic problems. Two different techniques are presented, namely the finite difference and the finite element methods. Issues regarding the general formulation and techniques for confining the infinite solution space and handling other difficulties that might arise with such a formulation are discussed.

Chapter 3 presents an integral equation approach based on the use of the moment method. Integral equations are formulated in terms of the unknown current and the Green's function. A variety of integral equation formulations are discussed with an added emphasis on techniques for handling the singularities associated with the Green's function.

Chapter 4 presents an application of a hybrid differential approach, namely the finite element method, for modeling the electromagnetic scattering problem. The solution is obtained using a hybrid model where the scatterer is enclosed by a hypothetical surface. The region interior to the surface is modeled using finite element analysis for obtaining the equivalent sources representing the scatterer, and the modal expansion technique is used in the exterior region, for computing the scattered fields.

A model for scattering from objects or inhomogeneities embedded in multilayered media is also presented. This model can be used for simulating scattering from re-bars or defects embedded in masonry or concrete. Such problems are of interest in the NDE of civil structures.

Chapter 5 handles issues regarding the use of moment method and the corresponding integral equations. A volume integral equation formulation for solving three dimensional problems is presented. The computational complexity related to the resultant dense matrix is reduced using conjugate gradient and fast Fourier transform methods. Efficient techniques are used to incorporate the effect of the multilayered media in the Green's function which is usually a major numerical problem.

Another application in NDE that needs prohibitively large computing resource is the development of probability of detection (POD) models which involve several executions of three dimensional measurement model. Implementation of such models are impractical on sequential computers. The solution to this type of problem is to resort to parallel computers. Parallel computer environments are characterized by a large number of processors together with large primary storage. Developing a model in such an environment is characterized by challenges different from those associated with sequential computers. The main issues involve load balancing on the processors and reduction of communication time among processors.

The topic of parallel computing is discussed in Chapter 6. Parallel computer architectures with particular emphasis on the hypercube architecture are presented. Chapter 7 discusses model implementation on a hypercube architecture. The application addresses the development of probability of detection (POD) models for assessing the applicability and reliability of NDE methods. The time requirements for running

POD models involving several executions of the measurement model is typically not practical in a sequential computer environment. Solutions to this problem obtained using parallel computers are presented.

Chapter 8 introduces solutions of the electromagnetic NDE inverse problem using another form of parallel computing architecture, namely the neurocomputer. The neural network is used to solve inverse problems using two different approaches. The first approach depends on a direct solution of the governing integral equation and is done using a Hopfield type neural network. The design of the network structure and parameters are presented. The second approach depends on building a map between the input and output space of the problem. A multilayered perceptron type neural network is used for this implementation. The basic network is augmented to build an incremental learning network whose architecture is inspired by the dynamic and modular features of the human brain.

Chapter 9 presents some concluding remarks on the subject of computational methods for electromagnetic problems and directions for future research.

CHAPTER 2. DIFFERENTIAL EQUATION APPROACH FOR SOLVING ELECTROMAGNETIC NDE APPLICATIONS

In the differential electromagnetic formulation the unknowns are typically the field variables or appropriate vector and/or scalar potentials. After the solution is obtained, all other electromagnetic quantities can be derived using the set of coupled differential equations established by James Clerk Maxwell in the 19th century. Maxwell's equations are represented as

$$\nabla \times \underline{E}(\underline{r}, t) = -\frac{\partial}{\partial t} \underline{B}(\underline{r}, t) \quad (2.1)$$

$$\nabla \times \underline{H}(\underline{r}, t) = \frac{\partial}{\partial t} \underline{D}(\underline{r}, t) + \underline{J}(\underline{r}, t) \quad (2.2)$$

$$\nabla \cdot \underline{B}(\underline{r}, t) = 0 \quad (2.3)$$

$$\nabla \cdot \underline{D}(\underline{r}, t) = \rho(\underline{r}, t) \quad (2.4)$$

where \underline{E} is the electric field intensity, \underline{D} is the displacement field or electric flux density, \underline{H} is the magnetic field intensity, and \underline{B} is the magnetic flux density.

Numerical solution generally involving a complex system of equations that requires the discretization of the domain into small components or elements. Formulation proceeds by assuming usually valid simplifications of the unknown variables and their derivatives over each element and relating the values of these unknowns to their correspondent over the surrounding elements. Two major techniques used

in the solution of differential equations are the finite difference and finite element methods. A discussion of computational aspects of the finite difference and finite element methods for modeling electromagnetic NDE phenomena is presented next.

Finite difference method

The finite difference method is the oldest technique used to deal with a system of differential equations. It has been applied for solving various EM problems in frequency as well as time domains. However, the finite difference method is well suited for time domain solutions and is known as FD-TD modeling. The evolution of parallel computers is increasingly in support of this type of modeling [20]. Time-domain transient problems are spatially confined in nature because of the time delay of the effects from the sources. For problems that do not have a natural boundary, the solution domain needs to be kept finite without drastically affecting the accuracy of the solution. This is achieved by using absorbing boundary conditions on the boundary of the solution region. The application of the finite difference method in modeling microwave time domain problems is first presented followed by a description of the concept of an absorbing boundary.

Finite difference time domain modeling

Techniques employing time-domain measurements and data processing are of important practical interests in microwave NDE, when the excitation is an impulse. Solutions can be obtained either in exact form using Cagniard-de Hoop method, for simple dispersionless problems, or numerically using Fourier Integrals for the frequency domain solution [21]. In applying numerical integration, careful choice of the

integration path is required to avoid singularities which can result in high values of integration errors. On the other hand, FD-TD is simple in both concept and execution. Moreover, with the advance of supercomputing technology, most practitioners of the FD-TD method claim that FD-TD techniques represent the method of choice for modeling problems that involve complex, electrically large, three-dimensional structures, such as aerospace systems [20].

A simple formulation for the FD-TD method is described as follows. From Maxwell's equations, the Cartesian values of the derivative of the electric field \underline{E} and the magnetic field \underline{H} at time $n\Delta t$ are given by [20]

$$\frac{\partial H_x^n}{\partial t} = \frac{1}{\mu} \left(\frac{\partial E_y^n}{\partial z} - \frac{\partial E_z^n}{\partial y} \right) \quad (2.5a)$$

$$\frac{\partial H_y^n}{\partial t} = \frac{1}{\mu} \left(\frac{\partial E_z^n}{\partial x} - \frac{\partial E_x^n}{\partial z} \right) \quad (2.5b)$$

$$\frac{\partial H_z^n}{\partial t} = \frac{1}{\mu} \left(\frac{\partial E_x^n}{\partial y} - \frac{\partial E_y^n}{\partial x} \right) \quad (2.5c)$$

$$\frac{\partial E_x^n}{\partial t} = \frac{1}{\epsilon} \left(\frac{\partial H_z^n}{\partial y} - \frac{\partial H_y^n}{\partial z} - \sigma E_x \right) \quad (2.5d)$$

$$\frac{\partial E_y^n}{\partial t} = \frac{1}{\epsilon} \left(\frac{\partial H_x^n}{\partial z} - \frac{\partial H_z^n}{\partial x} - \sigma E_y \right) \quad (2.5e)$$

$$\frac{\partial E_z^n}{\partial t} = \frac{1}{\epsilon} \left(\frac{\partial H_y^n}{\partial x} - \frac{\partial H_x^n}{\partial y} - \sigma E_z \right) \quad (2.5f)$$

Using the central difference approximation, the derivatives in the above equation can be approximated as

$$\frac{\partial F^n(i, j, k)}{\partial x} = \frac{F^n(i + \frac{1}{2}, j, k) - F^n(i - \frac{1}{2}, j, k)}{\Delta x}$$

$$\frac{\partial F^n(i, j, k)}{\partial t} = \frac{F^{n+\frac{1}{2}}(i, j, k) - F^{n-\frac{1}{2}}(i, j, k)}{\Delta t}$$

where F^n denotes either E^n or H^n , and

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

For stability of the solution, the time step is chosen such that [20, 21]

$$\Delta t \leq \left(\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2} \right)^{-\frac{1}{2}} \frac{1}{C_{\max}} \quad (2.6)$$

where C_{\max} is the maximum phase velocity.

The value of the field is dependent only on its nearest neighbors. This makes the implementation easy on simple parallel computing platforms. The solution scheme of the FD-TD method is known as the explicit scheme and it does not require matrix inversion. The order of complexity for such a scheme is shown to be $O(N)$ where N is the number of grid points. This is a significant advantage relative to other methods such as moment method where, as we discuss in Chapter 7, the matrix inversion is of order $O(N^3)$.

A critical issue in the differential approach to electromagnetic modeling of applications governed by the wave equation, where the field is defined on an infinite domain, is the need for eliminating the effect of the artificial boundary introduced during discretization of the finite domain of the space. This is achieved by using absorbing boundary conditions as discussed next.

Absorbing boundary conditions

Appropriate boundary conditions should be chosen on the outer boundary of the solution region to obtain the unbounded space solution. The choice of the boundary conditions can be exact as in the case of the radiation integral method where the Kirchoff formula is used on the boundary [22]. However, approximate boundary conditions are usually used to reduce the computational complexity by choosing absorbing boundary conditions at the outer boundary. This boundary condition eliminates the effect of the discontinuity created by confining the space to a finite domain

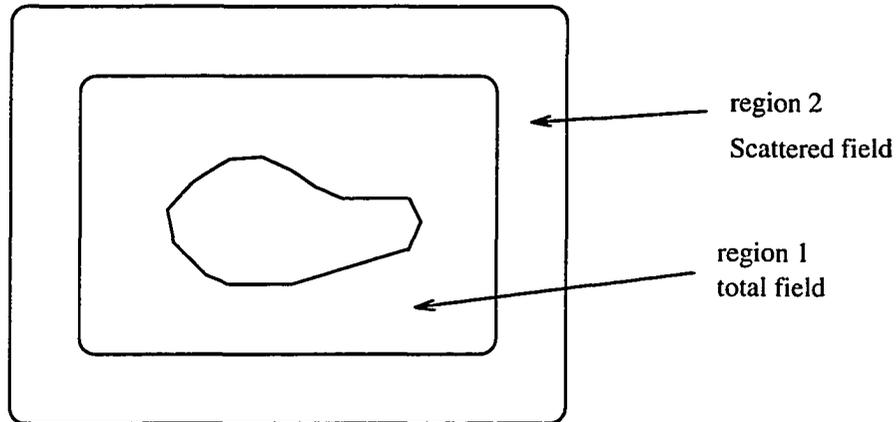


Figure 2.1: Finite difference lattice truncation

in the numerical solution. Figure 2.1 shows the lattice truncation in the finite difference method where the total field is used in region 1 that embeds the scatterer, and the scattered field is found in region 2. The estimated values of the scattered field in region 2 is then used to find the far scattered-field using an appropriate integral equation. However, the use of this zoning arrangement cannot eliminate completely the reflection from the boundaries. Appropriate boundary conditions are required to be enforced to satisfy the following requirements [23]:

- The reflections decrease rapidly as the position of the artificial boundary goes to infinity.
- The reflections decrease for shorter wavelengths.
- the reflections decrease as the incident wave approaches in a direction more parallel to some preferred direction.

Different approaches for absorbing boundary conditions (ABC) are discussed in [21]. We concentrate here on two basic approaches. A simple formulation of the first ABC approach, proposed by Engquist-Majda [24], is derived by considering a wave traveling in the negative x direction, such that the field ϕ satisfies

$$\left[\frac{\partial}{\partial x} - \frac{1}{C} \frac{\partial}{\partial t} \right] \phi(x, t) = 0 \quad (2.7)$$

where C is the wave speed.

The solution of the above equation is a wave propagating in the negative x-direction. To obtain a reflection free boundary at $x = 0$, a second order approximation is given for equation (2.7) by imposing the above equation at $x = \frac{\Delta x}{2}$. At $x = \frac{\Delta x}{2}$ and $t = (n + \frac{1}{2})\Delta t$, the approximations of the field derivatives are given by

$$\frac{\partial}{\partial x} \phi \left[\frac{\Delta x}{2}, (n + \frac{1}{2})\Delta t \right] \approx \frac{1}{\Delta x} \left[\phi^{n+\frac{1}{2}}(1) - \phi^{n+\frac{1}{2}}(0) \right] \quad (2.8a)$$

$$\frac{1}{C} \frac{\partial}{\partial t} \phi \left[\frac{\Delta x}{2}, (n + \frac{1}{2})\Delta t \right] \approx \frac{1}{c} \frac{1}{\Delta t} \left[\phi^{n+1}(\frac{1}{2}) - \phi^n(\frac{1}{2}) \right] \quad (2.8b)$$

The final form of the approximation of equation (2.7) is written as [21]

$$\phi^{n+1}(0) = \phi^n(1) + \left(\frac{c\Delta t - \Delta x}{c\Delta t + \Delta x} \right) (\phi^{n+1}(1) - \phi^n(0)) \quad (2.9)$$

Equation (2.9) is valid only for normal incident waves. For oblique incidence waves, the governing equation in the frequency domain is given as

$$\frac{d}{dx} \Phi(\underline{r}, \omega) - jk_x(\omega) \Phi(\underline{r}, \omega) = 0 \quad (2.10)$$

where $k_x = \sqrt{\omega^2/C^2 - k_y^2 - k_z^2}$. For angles of incidence close to normal, k_x can be represented as

$$k_x \approx \frac{\omega}{C} \left[1 - \frac{1}{2} \frac{C^2}{\omega^2} (k_y^2 + k_z^2) \right] \quad (2.11)$$

which is used in equation (2.10) to obtain

$$j\omega \frac{d}{dx} \Phi(\underline{r}, \omega) + \frac{\omega^2}{C} \Phi(\underline{r}, \omega) - \frac{1}{2} C (k_y^2 + k_z^2) \Phi(\underline{r}, \omega) = 0 \quad (2.12)$$

and, in the time domain

$$\left\{ \frac{\partial^2}{\partial x \partial t} - \frac{1}{C} \frac{\partial^2}{\partial t^2} + \frac{1}{2} \left(\frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) \right\} \phi(\underline{r}, t) = 0 \quad (2.13)$$

The above equation is used as an absorbing boundary condition approximation. The finite difference approximation for this equation is derived in [25].

An alternate ABC formulation, known as the Bayliss-Turkel ABC, is more accurate than the Engquist-Majda formulation. However, the Engquist-Majda formulation has the advantage of lower computational time requirements [26]. In the Bayliss-Turkel [23] formulation, the wave is represented as an expansion of the form

$$\phi(r, \theta, \phi, t) = \sum_{n=1}^{\infty} \frac{f_n(ct - r, \theta, \phi)}{r^n} \quad (2.14)$$

An operator L is defined such that

$$L = \frac{1}{c} \frac{\partial}{\partial t} + \frac{\partial}{\partial r}$$

Applying the operator L^m to the function $r^m \phi$ produces a result of order $O(\frac{1}{r^{m+1}})$, i.e.

$$L^m(r^m \phi) = O(r^{-m-1}) \quad (2.15)$$

From equation (2.15), a new recursive operator B_m is defined which has the characteristics

$$B_m(\phi) = O(1/r^{2m+1}) \quad m \geq 1 \quad (2.16)$$

where B_m is defined recursively as

$$B_1 = L + \frac{1}{r} \quad (2.17a)$$

$$B_m = \left(L + \frac{2m-1}{r}\right)B_{m-1} \quad (2.17b)$$

which can also be expressed as

$$B_m = \prod_{l=1}^m \left(L + \frac{2l-1}{r}\right) \quad (2.18)$$

Neglecting higher order terms, equation (2.16) can be written as

$$B_m(\phi) = 0 \quad (2.19)$$

which gives the ABC approximation.

Choosing, for example, $m = 2$, we obtain B_2 , such that

$$B_2 = \left(\frac{1}{C} \frac{\partial}{\partial t} + \frac{1}{\partial r} + \frac{1}{r}\right) \left(\frac{1}{C} \frac{\partial}{\partial t} + \frac{1}{\partial r} + \frac{3}{r}\right) \quad (2.20)$$

and the boundary condition is given as

$$\left(\frac{1}{C^2} \frac{\partial^2}{\partial t^2} + \frac{2}{C} \frac{\partial^2}{\partial t \partial r} + \frac{4}{rC} \frac{\partial}{\partial t} + \frac{\partial^2}{\partial r^2} + \frac{4}{r} \frac{\partial}{\partial r} + \frac{3}{r^2}\right) \phi(r, \theta, \phi, t) = 0 \quad (2.21)$$

For waves satisfying the Helmholtz equation, equation (2.13) can be written as

$$\phi(r, \theta, \phi, t) = e^{i\omega - kr} \sum_{n=1}^{\infty} \frac{f_n(\theta, \phi)}{r^n} \quad (2.22)$$

The operator B_m is written as

$$B_m = \prod_{l=1}^m \left(-jk + \frac{\partial}{\partial r} + \frac{2l-1}{r}\right) \quad (2.23)$$

and the boundary condition is given by (2.21). The term $\frac{\partial^2}{\partial r^2}$ is substituted by $\frac{\partial}{\partial \theta^2}$ and $\frac{\partial}{\partial \varphi^2}$ from the wave equation. The resulting boundary condition has only a first

order derivative with respect to r and the values of the boundary field depend only on the immediate neighborhood of the boundary [21].

For the 2D case, the expansion is given

$$\phi(r, \phi) = \frac{e^{-jk r}}{\sqrt{r}} \sum_{n=0}^{\infty} \frac{\phi_n(\phi)}{r^n} \quad (2.24)$$

and the operator B_m is defined as

$$B_m = \prod_{l=1}^m \left(-jk + \frac{\partial}{\partial r} + \frac{4l-1}{2r} \right) \quad (2.25)$$

For example, $B_2(\phi)$ is given as

$$B_2 = \left\{ \frac{\partial^2}{\partial r^2} + \left(2jk + \frac{3}{r} \right) \frac{\partial}{\partial r} + \left(-k^2 + \frac{3jk}{r} + \frac{3}{4r^2} \right) \right\} \phi = 0 \quad (2.26)$$

After removing the term $\frac{\partial^2}{\partial r^2}$ using the Helmholtz wave equation in cylindrical coordinates, the absorbing boundary condition can be expressed by the constraint equation [27]

$$\frac{\partial}{\partial r} = jk - \frac{1}{2r} + \text{other terms} \quad (2.27)$$

The above equation shows that Bayliss-Turkel ABC essentially satisfies Sommerfeld's radiation condition.

The technique of enforcing ABC on the truncated domain surface can also be used in conjunction with the finite element method presented next.

Finite element method

The finite element method is an engineering approach for solving partial differential equations using a set of discrete elements which approximates the continuous domain. The term finite element was first used in 1960 by Clough [28]. In contrast

to finite difference approach which solves continuum problems by merely discretizing the governing differential equations, the finite element method is based more on an engineering intuition. This is achieved by creating analogy between a system consisting of discrete elements and the original continuous system. Although the finite element method is used to solve the problem by considering a real discrete system, the methodology of the approach is not application oriented. Actually, a universal standard methodology has been developed over the years that is used in various engineering disciplines such as civil engineering, hydraulic engineering, electrical engineering, etc. [29, 30].

A general formulation of the FE method can be derived for a system defined in the domain Ω (Figure 2.2), such that the unknown u is governed by a system of differential equations $A_i(u) = 0, i = 1, 2, \dots$. This can be written as

$$\underline{A}(u) = \begin{Bmatrix} A_1(u) \\ A_2(u) \\ \cdot \\ \cdot \end{Bmatrix} = 0 \quad (2.28)$$

in Ω . In addition, let the system be governed by

$$\underline{B}(u) = \begin{Bmatrix} B_1(u) \\ B_2(u) \\ \cdot \\ \cdot \end{Bmatrix} = 0 \quad (2.29)$$

on the domain surface Γ .

The formulation proceeds by using any of two distinct methods based on weighted residuals and variational calculus as explained next.

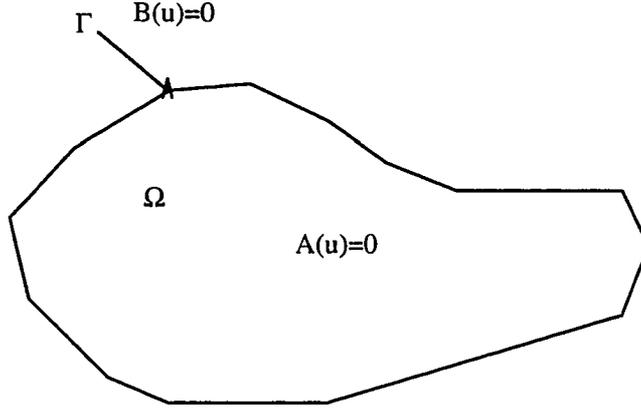


Figure 2.2: Finite element solution domain

Weighted residuals

In the weighted residual approach, equations (2.28) and (2.29) are enforced at all points of Ω and Γ by solving the following equation

$$\int_{\Omega} \underline{w}^T \underline{A}(u) d\Omega + \int_{\Gamma} \hat{\underline{w}}^T \underline{B}(u) d\Gamma = 0 \quad (2.30)$$

where w and \hat{w} are arbitrary weights.

Constraints on the function u depend on the order of differentiation in $A(u)$ and $B(u)$. If the highest order of a derivative in any of the operators A_i or B_i is k , then the u has to be continuous of order $k - 1$ (C_{k-1}). However, a *weak form* of the problem can be written using integration by parts in terms of new operators in the form

$$\int_{\Omega} C(\underline{w})^T \underline{D}(u) d\Omega + \int_{\Gamma} E(\hat{\underline{w}})^T \underline{F}(u) d\Gamma = 0 \quad (2.31)$$

where the operators C,D,E, and F have lower order derivatives, and this relaxes the required order of continuity of u .

The finite element formulation is obtained from the above equation by approximating the unknown u as an expansion in terms of the discrete values of the unknown $u_i, i = 1, 2, ..n$ which can be chosen to be the element nodal values or edge values. In other words

$$u \approx \hat{u} = \sum_{i=1}^n N_i u_i \quad \text{in D} \quad (2.32)$$

where the functions N_i are known as shape functions.

The trial function \hat{u} is given as a summation of element values \hat{u}^e , and thus for l elements, we have

$$\hat{u} = \sum_{e=1}^l \hat{u}^e \quad (2.33)$$

where the unknown function \hat{u}^e in element e is given as

$$\hat{u}^e = \sum_{k=1}^m N_k u_k^e \quad (2.34)$$

where m is the number of elements.

Equations (2.32)-(2.34) are substituted in equations (2.30) or (2.31) to obtain a global linear system of equations. Different choices of w in equations (2.30) or (2.31) can be used. For example, choosing $w_i = \delta_i$ gives the point collocation method whereas the choice of $w_j = N_j$ gives the Galerkin method.

As will be shown for a simple 2D case, the problem can finally be cast in the form of linear system

$$K \underline{u} = F \quad (2.35)$$

that can be solved.

The system matrix K (commonly referred to as the stiffness matrix) is a sparse, banded and symmetric matrix for self-adjoint systems. Even in the case of problems with non-self-adjoint operators, symmetry can be achieved by restoring the self-adjointness without altering the basic equation [30]. Computational techniques have been developed to automate the minimization of the matrix bandwidth and obtain the lowest storage requirements [31].

Variational principle

A general characteristic of the FE method is that the solution satisfies minimization of a global scalar function known as the energy function. The Euler equations of this energy function are identical to the governing system equations. For an energy functional in the form

$$\chi = \int_{\Omega} F(u, \frac{\partial}{\partial x}u, ..)d\Omega + \int_{\Gamma} E(u + \frac{\partial}{\partial x}u, ...)d\Gamma \quad (2.36)$$

The solution is the stationary point of χ defined by

$$\delta\chi = 0 \quad (2.37)$$

From equation (2.32), we get

$$\delta\chi = \frac{\partial\chi}{\partial u_1}\delta u_1 + \frac{\partial\chi}{\partial u_2}\delta u_2 + \dots = 0 \quad (2.38)$$

For (2.38) to be true for any δu_i , we require for a linear self-adjoint system

$$\begin{bmatrix} \frac{\partial\chi}{\partial u_1} \\ \cdot \\ \cdot \\ \frac{\partial\chi}{\partial u_n} \end{bmatrix} = 0 \quad (2.39)$$

The finite element formulation for a selected set of electromagnetic NDE problems is discussed next.

Finite element analysis techniques for 2D problems in EM NDE

In this subsection we give the basic formulation for the FE approach for two problems in EM NDE. First, we consider the microwave scattering problem where the governing differential equation is the Helmholtz wave equation. The second case is the eddy current NDE problem which is governed by a diffusion equation. The weighted residual Galerkin method is chosen for deriving the formulation for the first case, while the variational principle for minimizing a global energy function is used for the formulation in the second case.

FE techniques for modeling EM wave scattering Consider a 2D problem in the xy plane where the wave equation is of the form

$$(\nabla_{xy}^2 + k^2)\psi = 0 \quad (2.40)$$

where ψ , representing the plane wave traveling in the xy plane, is the scattered magnetic field H_z^s for TE^z case, or the scattered electric field E_z^s for TM^z case.

As in the finite difference formulation, the finite element method is applied in a closed domain with finite boundaries. An ABC or a coupling scheme can be introduced between the solution region and the exterior space. This is discussed in detail in Chapter 4 which casts the problem as a Dirichlet boundary problem in the domain Ω . Equation (2.30) can be written as

$$\int \int_{\Omega} [(\nabla_{xy}^2 + k^2)\psi(x, y)]w(x, y)dx dy = 0 \quad (2.41)$$

By using Green's first identity, (2.41) can be written as

$$\int \int_{\Omega} [\nabla\psi \cdot \nabla w - k^2\psi w] d\Omega - \int_{\Gamma} w \hat{n} \cdot \nabla\psi d\Gamma = 0 \quad (2.42)$$

where \hat{n} is the outward unit normal to Γ

In applying (2.42) for different elements inside the domain, the line integral contribution is nullified in the internal boundaries between the elements if the integrand is continuous. From Maxwell's equations we get for TM^z waves:

$$\begin{aligned} \hat{n} \cdot \nabla\psi &= \hat{n} \cdot (\hat{z} \times (-j\omega\mu\mathbf{H})) && \text{for } TM^z \\ &= -j\omega\mu H_t && \end{aligned} \quad (2.43)$$

and for TE^z waves:

$$\begin{aligned} \hat{n} \cdot \nabla\psi &= \hat{n} \cdot (\hat{z} \times (j\omega\epsilon\mathbf{E})) && \text{for } TE^z \\ &= j\omega\epsilon E_t && \end{aligned} \quad (2.44)$$

where H_t and E_t are the tangential components of the field at the element boundary. From equations (2.43) and (2.44), continuous functions can be obtained as $\frac{\psi}{\mu}$ for TM^z case and $\frac{\psi}{\epsilon}$ for TE^z case. Also by choosing the arbitrary weight function w to be 0 on the outer boundary Γ we obtain the FE formulation as

$$\int \int_{\Omega} \frac{1}{\mu} [\nabla\psi \cdot \nabla w - k^2\psi w] d\Omega = 0 \quad \text{for } TM^z \quad (2.45)$$

$$\int \int_{\Omega} \frac{1}{\epsilon} [\nabla\psi \cdot \nabla w - k^2\psi w] d\Omega = 0 \quad \text{for } TE^z \quad (2.46)$$

Denoting the integral in (2.45) or (2.46) as X , each integral can be written in terms of the element contributions as

$$X = \sum_{e=1}^l X^e = 0 \quad (2.47)$$

The field within each element is written in an expansion in terms of the nodal values of the field. For an element with m nodes

$$\psi^e = \sum_{p=1}^m N_p \psi_p \quad (2.48)$$

$$= \underline{N} \cdot \underline{\psi} \quad (2.49)$$

Using Galerkin method where the weight function w is chosen to be the same as the shape function N , we have a set of equations for each $p, p = 1 \dots m$. For TM^z , we have

$$X_p^e = \int \int_{\Omega^e} \frac{1}{\mu} \left[\frac{\partial N_p^e}{\partial x} \frac{\partial \underline{N}^e}{\partial x} + \frac{\partial N_p^e}{\partial y} \frac{\partial \underline{N}^e}{\partial y} - k^2 N_p \underline{N} \right] \cdot \underline{\psi} d\Omega = 0 \quad (2.50)$$

and similarly for TE^z case.

If, for instance, we consider the formulation using triangular shape elements (Figure 2.3) with linear shape function (C_0 elements), the field within an element can be expressed as

$$\psi^e = \alpha_1 + \alpha_2 x + \alpha_3 y \quad (2.51)$$

$$\begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

α_1 , α_2 , and α_3 can be given in terms of coordinates (x_i, y_i) of the three vertices as follows:

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \frac{1}{2\Delta} \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{bmatrix}$$

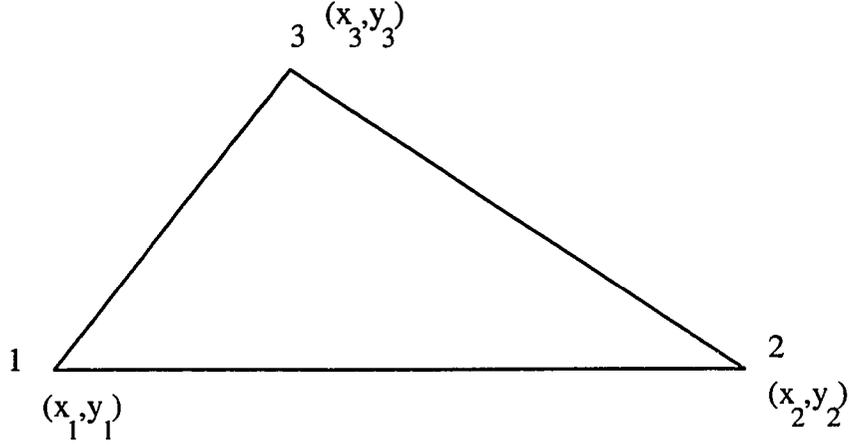


Figure 2.3: Triangular element

where: Δ is the area of the element, and

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} = \begin{bmatrix} (x_2 \cdot y_3 - x_3 \cdot y_2) & (x_3 \cdot y_1 - x_1 \cdot y_3) & (x_1 \cdot y_2 - x_2 \cdot y_1) \\ (y_2 - y_3) & (y_3 - y_1) & (y_1 - y_2) \\ (x_3 - x_2) & (x_1 - x_3) & (x_2 - x_1) \end{bmatrix}$$

The field ψ inside the element is given in terms of the three nodal fields as

$$\psi = N_1\psi_1 + N_2\psi_2 + N_3\psi_3 \quad (2.52)$$

$$= \frac{1}{2\Delta} [(a_1 + b_1x + c_1y)\psi_1 + (a_2 + b_2x + c_2y)\psi_2 + (a_3 + b_3x + c_3y)\psi_3] \quad (2.53)$$

The above expansion is used in (2.50) to obtain the matrix equations

TM:

$$[[S^e] - k^2[R^e]] [\psi] = [Q^e] \quad (2.54)$$

TE:

$$\left[\frac{1}{\epsilon_r} [S^e] - k_0^2 [R^e] \right] [\psi] = [Q^e] \quad (2.55)$$

where the element matrices S^e and R^e are given as

$$S^e = \frac{1}{4\Delta} \begin{bmatrix} (b_1.b_1 + c_1.c_1) & (b_1.b_2 + c_1.c_2) & (b_1.b_3 + c_1.c_3) \\ (b_2.b_1 + c_2.c_1) & (b_2.b_2 + c_2.c_2) & (b_2.b_3 + c_2.c_3) \\ (b_3.b_1 + c_3.c_1) & (b_3.b_2 + c_3.c_2) & (b_3.b_3 + c_3.c_3) \end{bmatrix} \quad (2.56)$$

$$R^e = \frac{\Delta}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad (2.57)$$

The three element matrices $[S^e]$, $[R^e]$ and $[Q^e]$ are used to form the global system of equations. If the global matrix were not compacted, then it would be of size $P \times P$ where P is the total number of nodes. However, the nodes are numbered chosen in a manner as to minimize the bandwidth of the global matrix.

As an example [32], consider the simple mesh shown in Figure 2.4 with corresponding local elemental matrix equations

$$K^a = \begin{bmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{41} & a_{42} & a_{44} \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ A_4 \end{bmatrix} = \begin{bmatrix} Q_{a1} \\ Q_{a2} \\ Q_{a4} \end{bmatrix} \quad (2.58)$$

$$K^b = \begin{bmatrix} b_{22} & b_{25} & b_{24} \\ b_{52} & b_{55} & b_{54} \\ b_{42} & b_{45} & b_{44} \end{bmatrix} \begin{bmatrix} A_2 \\ A_5 \\ A_4 \end{bmatrix} = \begin{bmatrix} Q_{b2} \\ Q_{b5} \\ Q_{b4} \end{bmatrix} \quad (2.59)$$

$$K^c = \begin{bmatrix} c_{22} & c_{23} & c_{25} \\ c_{32} & c_{33} & c_{35} \\ c_{52} & c_{53} & c_{55} \end{bmatrix} \begin{bmatrix} A_2 \\ A_3 \\ A_5 \end{bmatrix} = \begin{bmatrix} Q_{c2} \\ Q_{c3} \\ Q_{c5} \end{bmatrix} \quad (2.60)$$

$$K^d = \begin{bmatrix} d_{33} & d_{36} & d_{35} \\ d_{63} & d_{66} & d_{65} \\ d_{53} & d_{56} & d_{55} \end{bmatrix} \begin{bmatrix} A_3 \\ A_6 \\ A_5 \end{bmatrix} = \begin{bmatrix} Q_{d3} \\ Q_{d6} \\ Q_{d5} \end{bmatrix} \quad (2.61)$$

In forming the global matrix, after incorporating element a , we have

$$K = \begin{bmatrix} a_{11} & a_{12} & \circ & a_{14} & \circ & \circ \\ a_{21} & a_{22} & \circ & a_{24} & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ & \circ \\ a_{41} & a_{42} & \circ & a_{44} & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ & \circ \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \end{bmatrix} = \begin{bmatrix} Q_{a1} \\ Q_{a2} \\ \circ \\ Q_{a4} \\ \circ \\ \circ \end{bmatrix} \quad (2.62)$$

After incorporating element b :

$$K = \begin{bmatrix} a_{11} & a_{12} & \circ & a_{14} & \circ & \circ \\ a_{21} & a_{22} + b_{22} & \circ & a_{24} + b_{24} & b_{25} & \circ \\ \circ & \circ & \circ & \circ & \circ & \circ \\ a_{41} & a_{42} + b_{42} & \circ & a_{44} + b_{44} & b_{45} & \circ \\ \circ & b_{52} & \circ & b_{54} & b_{55} & \circ \\ \circ & \circ & \circ & \circ & \circ & \circ \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \end{bmatrix} = \begin{bmatrix} Q_{a1} \\ Q_{a2} + Q_{b2} \\ \circ \\ Q_{a4} + Q_{b4} \\ Q_{b5} \\ \circ \end{bmatrix} \quad (2.63)$$

Elements c and d can be incorporated similarly. It is clear that the global matrix is banded and symmetric.

FE methods for eddy current modeling Electromagnetic fields in eddy current phenomenon are described by the quasi-static form of Maxwell's equations. The resulting governing equation is a parabolic partial differential equation or the *diffusion equation*:

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \underline{A} \right) = -\underline{J}_s + \sigma \frac{d\underline{A}}{dt} \quad (2.64)$$

where \underline{A} is the magnetic vector potential defined by $\nabla \times \underline{A} = \underline{B}$.

If the operation is a single frequency operation and the material is isotropic with no spatial variations in μ , then the governing equations relating to eddy current

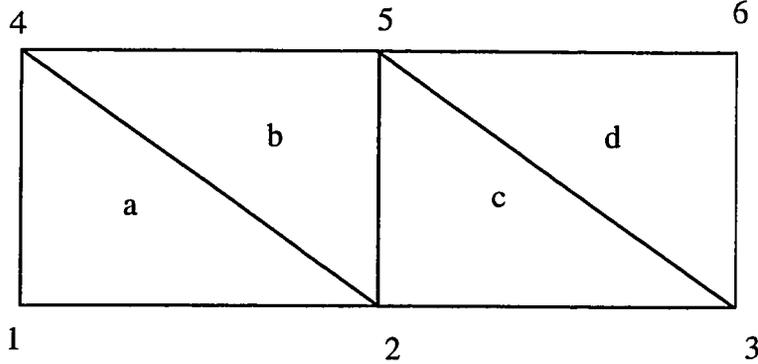


Figure 2.4: A simple mesh

problem reduces to the form

$$\frac{1}{\mu} \nabla^2 \vec{A} = j\omega\sigma \vec{A} - \vec{J}_s \quad (2.65)$$

The two-dimensional diffusion equation (2.64) can be expressed in variational terms by the energy function

$$F = \int_{\Omega} \left[\frac{1}{2\mu} \left(\left| \frac{d\vec{A}}{dx} \right|^2 + \left| \frac{d\vec{A}}{dy} \right|^2 \right) + j\frac{\omega\sigma}{2} |\vec{A}|^2 - \vec{J}_s \vec{A} \right] d\Omega \quad (2.66)$$

where v is the volume enclosing the region to be modeled.

The first, second and third terms inside the integrand represent the energy terms corresponding to magnetic field, eddy currents, and source current, respectively.

Element matrix equations are formed separately for all the elements on the finite element region, such that

$$\frac{\partial F}{\partial A_i} = 0 \quad (2.67)$$

The individual element equations are then combined according to the node numbers into a single global matrix equation.

Finite element methods for solving 3D problems

In the two dimensional problem, the unknown function is the axial field component which is continuous. However, for 3D problems, this is not the case since the normal component of the electric or magnetic fields has a discontinuity at the surface between two materials. A second difficulty that also arises in 3D problems is the possible occurrence of *spurious modes* that contaminate the true solution. Spurious modes are nonphysical solutions which are not divergence-free. One approach to avoid these difficulties is the use of *edge elements*. These are described next.

Edge-based finite element In using edge elements, the degrees of freedom of the unknown function are chosen to be associated with the edges of the element [33, 34]. For example, with a cubic edge element with local coordinates (ξ, η, ζ) (Figure 2.5), the field in the element ψ^e is given as

$$\begin{aligned}\psi_{\xi}^e &= \sum_{i=1}^8 N_i \psi_{\xi i} \\ \psi_{\eta}^e &= \sum_{i=1}^8 N_i \psi_{\eta i} \\ \psi_{\zeta}^e &= \sum_{i=1}^8 N_i \psi_{\zeta i}\end{aligned}\tag{2.68}$$

where, i is the node number, and N_i is the shape function given by

$$N_i = \frac{1}{8}(1 + \xi_i \xi)(1 + \eta_i \eta)(1 + \zeta_i \zeta)$$

The two field components along an edge are forced to be equal, and this can be written as

$\psi_{\xi_1} = \psi_{\xi_2} = \psi_1$	$\psi_{\eta_1} = \psi_{\eta_4} = \psi_5$	$\psi_{\zeta_1} = \psi_{\zeta_5} = \psi_9$
$\psi_{\xi_3} = \psi_{\xi_4} = \psi_2$	$\psi_{\eta_5} = \psi_{\eta_8} = \psi_6$	$\psi_{\zeta_2} = \psi_{\zeta_6} = \psi_{10}$
$\psi_{\xi_5} = \psi_{\xi_6} = \psi_3$	$\psi_{\eta_6} = \psi_{\eta_7} = \psi_7$	$\psi_{\zeta_3} = \psi_{\zeta_7} = \psi_{11}$
$\psi_{\xi_7} = \psi_{\xi_8} = \psi_4$	$\psi_{\eta_2} = \psi_{\eta_3} = \psi_8$	$\psi_{\zeta_4} = \psi_{\zeta_8} = \psi_{12}$

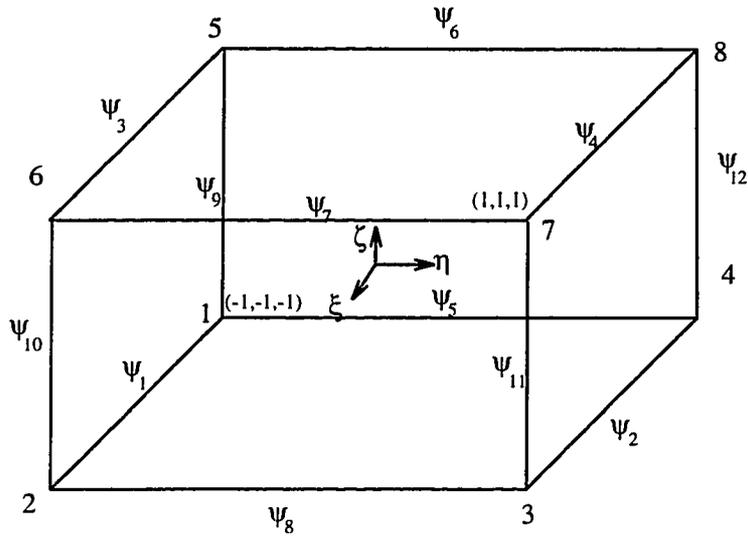


Figure 2.5: Edge element

And the field is written as

$$\psi^e = \sum_{i=1}^{12} \psi_i \quad (2.69)$$

This choice for the field guarantees that

1. $\nabla \cdot \psi = 0$ (divergence free solution)
2. tangential components of the field are continuous while normal components are discontinuous.

Nodal-Based Elements Conventional nodal-based elements have some advantages since they can be used easily in generating a mesh for a complex object with curved surfaces. However, using this type of elements may lead to spurious modes in the solution [35]. One way to eliminate the spurious modes is to introduce

a penalty term to the energy functional. Another method is to solve the problem using the vector potential \underline{A} and the scalar potentials ϕ as the unknowns. The electric field is then obtained from these potentials as

$$\underline{E} = -j\omega\underline{A} + \nabla\phi \quad (2.70)$$

A gauge has to be chosen relating the vector and scalar potentials. If a Lorentz gauge is used such that

$$\nabla \cdot \underline{A} = -j\omega\epsilon\mu\phi \quad (2.71)$$

both \underline{A} and ϕ will be continuous on material discontinuities and the problem can be solved with 4 degrees of freedom for A and ϕ .

Another gauge is in the form

$$\nabla \cdot \epsilon \underline{A} = -j\omega\epsilon^2\mu\phi \quad (2.72)$$

which results in a problem with 3 degrees of freedom corresponding to \underline{A} , while the scalar potential ϕ is identically 0. However, the normal component A_n of \underline{A} is not continuous over material discontinuities. This can be handled by introducing a cut in the mesh along any material discontinuity.

As in the finite difference formulation, the finite element method is applied in a finite domain and a coupling scheme introduced between the solution region and the exterior space. A commonly used technique is the unimoment method [36], where the field ϕ on a surface S that encloses the finite element region is given as a summation of basis functions ϕ_n ,

$$\phi[\text{on } S] = \sum_{n=1}^N a_n \phi_n \quad (2.73)$$

The finite element solution is obtained by solving the Dirichlet problem for the region interior to the surface S with each ϕ_n as the boundary condition. Correspondingly, this gives a solution $\Lambda_n(x, y)$ for the internal region.

In the external region, the field is given as $\phi^i + \phi^{sc}$, where ϕ^i is the incident field and ϕ^{sc} is the scattered field external to the surface S . The scattered field is expanded in terms of appropriate basis functions ψ_n

$$\phi^{SC} = \sum_{n=1}^N b_n \psi_n \quad (2.74)$$

The total solution is found by determining the coefficients a_n and b_n in equations (2.73) and (2.74). This is achieved by matching the boundary conditions on S .

Another technique to couple the interior and exterior regions is the bymoment method [37]. Again the field in the interior is represented as in equation (2.73). The Green's theorem is used on a new surface S' that lie within the surface S . From Green's theorem we get an integral relation

$$\int_{S'} \left[\xi_j \frac{\partial \phi^{sc}}{\partial n} - \phi^{sc} \frac{\partial \xi_j}{\partial n} \right] ds = 0 \quad (2.75)$$

where ξ_j are testing functions chosen to satisfy Helmholtz equation. The scattered field ϕ_{sc} in the above equation is represented as $(\phi - \phi^i)$, where ϕ is given by summation of solutions corresponding to basis functions ϕ_n on S , i.e.

$$\phi[S'] = \sum_{n=1}^N a_n \Lambda_n(S') \quad (2.76)$$

Equation (2.75) is represented as a system of linear equations with the coefficients a_n as unknowns.

A third formulation using finite element analysis is given in [38]. The approach uses the Extended Boundary Condition Method EBCM which is described later. The

incident and scattered field are expanded in terms of spherical harmonics. The finite element is used to find what is known as the *informed* basis functions which contain adequate information about the scatterer. This numerical solution is found to be stable as long as the scatterer is not very elongated.

Another technique used in problems related to viscous fluid flow modeling enables the extension of the FE mesh to cover an infinite domain using semi-infinite elements along the edge [39, 40]. Infinite elements for time-harmonic problems for modeling fluid surface waves were developed by Bettess and Zienkiewicz [41]. The shape function for these infinite elements is chosen such that it satisfies the Sommerfeld's radiation condition at infinity and numerical integration is performed along a semi-infinite range to form the stiffness matrix. Beer and Meek [42] used another technique to develop infinite elements for elasticity problems using a conformal mapping of the elements to finite nondimensional square or cubic shaped elements (for 2D and 3D cases). Applications using the infinite elements techniques in electromagnetics are also given in [43, 44] for modeling unbounded guided wave problems.

CHAPTER 3. INTEGRAL EQUATION APPROACH FOR SOLVING ELECTROMAGNETIC NDE APPLICATIONS

Integral equation representation for electromagnetic NDE applications can be written in a general form as

$$L(f) = g \tag{3.1}$$

where L is a linear integral operator on a domain Γ that can be a line, surface or volume. The functions f and g are complex functions defined in the space Γ .

The solution to the integral equation is obtained using the moment method MM whose formulation was established by Harrington in 1968 [45]. The method transforms the integral equation into a system of linear equations which can be handled easily on computers. In general, for an integral operator L in (3.1), the unknown function f is expanded as a series of basis functions f_n in the domain of L as follows:

$$f = \sum_n \alpha_n f_n \tag{3.2}$$

Taking the inner product of Equation (3.1) with a weight function w_m [46], equations (3.1) and (3.2) give

$$\sum_n \alpha_n \langle w_m, Lf_n \rangle = \langle w_m, g \rangle \tag{3.3}$$

Equation (3.3) is the linear system of equations to be solved to obtain the approximate solution of f .

There are three basic formulations of the MM that are used in scattering problems and are presented next.

Surface integral representation

In this formulation, the scatterer is assumed to be of homogeneous characteristic. Formulation is derived from the vector Green's theorem

$$\int_V (\underline{Q} \cdot \nabla \times \nabla \times \underline{P} - \underline{P} \cdot \nabla \times \nabla \times \underline{Q}) dV = \int_S (\underline{P} \times \nabla \times \underline{Q} - \underline{Q} \times \nabla \times \underline{P}) \cdot d\mathbf{S} \quad (3.4)$$

Let the functions P and Q be given as

$$\underline{P} = \hat{a}g(\underline{r}, \underline{r}') \quad (3.5)$$

$$\underline{Q} = \underline{E}(\underline{r}) \quad (3.6)$$

where \hat{a} is a unit vector of arbitrary orientation, and g is the scalar Green's function given by

$$g(\underline{r}, \underline{r}') = \frac{e^{-jk|\underline{r}-\underline{r}'|}}{4\pi|\underline{r}-\underline{r}'|}$$

The electric field $E(\underline{r})$, at any point \underline{r} on the scatterer surface is given in terms of the incident field $\underline{E}^i(\underline{r})$ and integration of a function E_s over the scatterer surface S

$$\frac{1}{2}\underline{E}(\underline{r}) = \underline{E}^i(\underline{r}) + \int_S \underline{E}^s(\underline{r}') ds' \quad (3.7)$$

where \underline{E}^s is given in terms of the surface electric current \underline{J}_s and magnetic current \underline{J}_{ms} , the surface electric charge ρ as

$$\underline{E}^s = -j\omega\mu\underline{J}_s g - \underline{J}_{ms} \times \nabla' g + \frac{\rho}{\epsilon} \nabla' g \quad (3.8)$$

The electric field integral equation EFIE is obtained from Equation (3.7) in addition to enforcing the appropriate boundary conditions between the interior and exterior regions of the surface S [47]. The electric and magnetic currents J_s and J_{ms} unknowns on S are obtained using the moment method.

The dual system of equations is the magnetic field integral equation MFIE given as

$$\frac{1}{2}\underline{H}(\underline{r}) = \underline{H}^i(\underline{r}) + \int_S \underline{H}_s(\underline{r}') ds' \quad (3.9)$$

$$\underline{H}^s = -j\omega\mu_s \underline{J}_{ms} g - \underline{J}_s \times \nabla' g + \frac{\rho_m}{\mu} \nabla' g \quad (3.10)$$

where ρ_m is the magnetic charge.

The use of EFIE or MFIE alone can lead to the problem of nonunique solutions that exist at frequencies corresponding to the resonance in the interior region. Stable solutions were found to be obtained using linear combinations of EFIE and MFIE [48].

T-matrix method

In the EFIE and MFIE, the unknowns of the formulation are the surface electric and magnetic currents. However, in the T-matrix method (or the extended boundary condition method EBCM) the formulation is given in terms of surface fields. From Green's theorem, it can be shown that the field ϕ is governed by

$$\int_S \left[\phi(\underline{r}') \frac{\partial g}{\partial n'} - g \frac{\partial \phi(\underline{r}')}{\partial n'} \right] ds' = -\phi^i \quad \underline{r} \text{ in } S \quad (3.11)$$

$$= \phi^s \quad \underline{r} \text{ outside } S \quad (3.12)$$

Equation (3.11) states that the integral of the field contribution on S nulls the incident field inside S and is known as the *extinction* theorem.

Green's function in equations (3.11) and (3.12) is represented as an expansion in terms of orthogonal basis functions. The incident field is also decomposed into its harmonics which are cylindrical harmonics for 2D problems and spherical harmonics for 3D cases. Equation (3.11) and the orthogonality of the basis functions are used to find integral formulation of the expansion coefficients a_n in terms of the field on the surface S .

The scattered field in the external region of the surface S is expanded with expansion coefficients b_n , and Equation (3.12) is used to find the integral representation for b_n .

The relation between the scattered wave and the incident wave is found by representing the relation between the coefficients a_n and b_n in terms of the transition matrix, or the T-matrix, such that

$$[b] = [T][a] \quad (3.13)$$

The method can be used to find scattering from complex objects. Since the method uses spherical (or cylindrical) harmonic expansions, numerical problems arise if the object, (or its cross section for 2D problems), is very elongated in one direction, and whenever the object has corners.

Volume integral representation

The surface integral and T-matrix approaches are used for homogeneous objects and the unknowns are taken to be surface values. On the other hand, volume integral representation is the tool to be used for inhomogeneous or anisotropic objects. In addition, the volume integral equation are free from the problem of nonuniqueness,

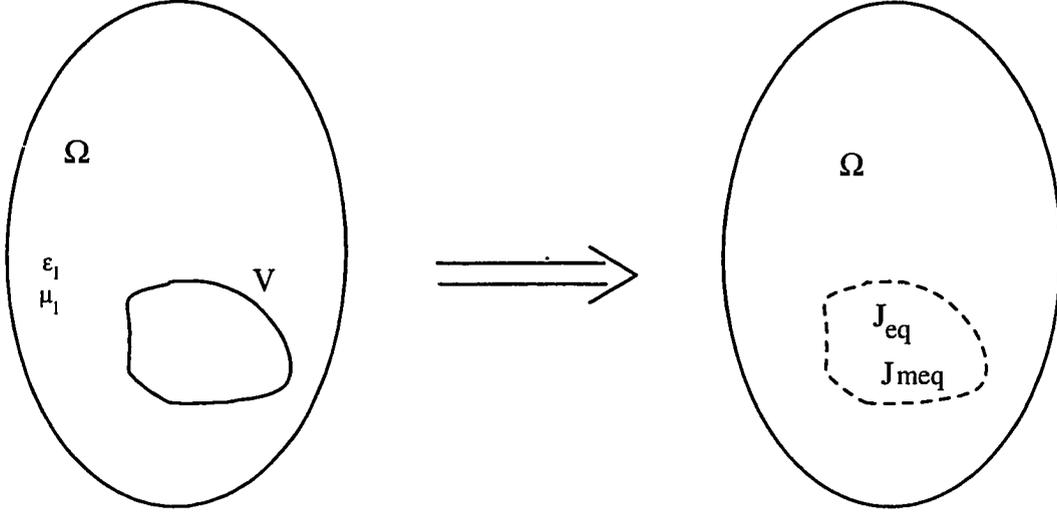


Figure 3.1: Volume equivalent current

of the solution, encountered in surface integral formulation. This section describes the volume integral approach in detail.

For a scatterer V present in the space Ω of homogeneous characteristics ϵ_1 and μ_1 , the scatterer can be replaced by its equivalent electric and magnetic currents (Figure 3.1), where

$$\underline{J}_{eq}(\underline{r}) = j\omega[\epsilon(r) - \epsilon_1]\underline{E}(\underline{r}) \quad (3.14)$$

$$\underline{J}_{m,eq}(\underline{r}) = j\omega[\mu(r) - \mu_1]\underline{H}(\underline{r}) \quad (3.15)$$

These equivalent currents induce scattered electric fields, such that

$$\underline{E}^s = -j\omega\mu_1 \int_V \underline{G}_e \cdot \underline{J}_{eq} dv' - \int_V \underline{G}_m \cdot \underline{J}_{m,eq} dv' \quad (3.16)$$

where G_e is the electric dyadic Green's function, and G_m is the magnetic dyadic

Green's function which is related to G_e as

$$G_m = \nabla \times G_e$$

For a dielectric object with constant permeability μ_1 , the scattered field \underline{E}^s is related to equivalent current \underline{J}_{eq} as

$$\underline{E}^s = -j\omega\mu \int_{v'} \underline{J}_{eq}(\underline{r}') G(\underline{r}, \underline{r}') dv' \quad (3.17)$$

where G is the dyadic Green's function

The general form of G can be derived as follows:

The equivalent current \underline{J}_{eq} induces a vector potential \underline{A} and scalar potential ϕ , such that

$$\underline{E}^s = -j\omega\mu\underline{A} - \nabla\phi \quad (3.18)$$

$$= -j\omega\mu\underline{A} + \frac{1}{j\omega\epsilon} \nabla(\nabla \cdot \underline{A}) \quad (3.19)$$

The magnetic vector potential \underline{A} is given as

$$\underline{A} = \int_{v'} \underline{J}_{eq}(\underline{r}') g(\underline{r}, \underline{r}') dv' \quad (3.20)$$

where $g = \frac{e^{-jkR}}{4\pi R}$ and $R = |\underline{r} - \underline{r}'|$.

From equations (3.17), (3.19) and (3.20), G can be expressed as

$$G = (I + \frac{1}{k^2} \nabla \nabla) g \quad (3.21)$$

where I is the unit dyad.

Equation (3.17) is represented in terms of one unknown \underline{J}_{eq} by writing \underline{E}^s in terms of the total field \underline{E} and the incident field \underline{E}^i to obtain

$$\underline{E}^s = \underline{E} - \underline{E}^i \quad (3.22)$$

$$= \frac{\underline{J}_{eq}}{j\omega(\epsilon - \epsilon_1)} - \underline{E}^i \quad (3.23)$$

Then (3.17) is solved for \underline{J}_{eq} , which can be used to obtain solutions for different fields.

The integration of Green's function is the most crucial issue in volume integral equations because of the singularity at the source point. Methods to handle this singularity are presented next.

Green's function

It can be shown [49] that integrals of the form

$$\int_v \frac{dv}{R^\beta} \quad R = 0 \text{ in the domain of } v$$

are convergent for $0 < \beta < 3$.

However, the second term of G in (3.21) has $1/R^3$ singularity, and has to be regularized. Two approaches can be used to deal with this singularity. The first approach involves the use of current derivatives, or electric charges, and the second approach is based on regularizing the singularity.

Using current derivatives The second term in (3.18) is in terms of the potential ϕ , which is induced from charges ρ as

$$\phi = \int_{v'} \frac{\rho(\underline{r}')}{\epsilon} g(\underline{r}, \underline{r}') dv' \quad (3.24)$$

The field due to this potential E_ϕ^s is written as

$$\underline{E}_\phi^s = \frac{1}{j\omega\epsilon} \nabla \int \nabla' \cdot \underline{J}(\underline{r}') g(\underline{r}, \underline{r}') dv' \quad (3.25)$$

The choice of the basis functions of the current determines the basis functions for charge. A common choice is a rooftop basis function for the current that leads to a doublet of pulse functions for ρ [50].

We consider here only the x-component of the field E_x^s . Assuming the discretization element width in the x direction is equal to Δ (Figure 3.2), the field E_x^s at element m due to element n is given by

$$E_x^{s(mn)} = (\phi_{x+}^{mn} - \phi_{x-}^{mn})/\Delta \quad (3.26)$$

where

$$\begin{aligned} \phi_{x+}^{mn} = \frac{J_x^n}{j\omega\epsilon\Delta} & \left[\int_{n-} g(\mathbf{r}_m + \frac{\Delta}{2}\hat{x}, \mathbf{r}') dv' \right. \\ & \left. - \int_{n+} g(\mathbf{r}_m + \frac{\Delta}{2}\hat{x}, \mathbf{r}') dv' \right] \end{aligned} \quad (3.27)$$

$$\begin{aligned} \phi_{x-}^{mn} = \frac{J_x^n}{j\omega\epsilon\Delta} & \left[\int_{n-} g(\mathbf{r}_m - \frac{\Delta}{2}\hat{x}, \mathbf{r}') dv' \right. \\ & \left. - \int_{n+} g(\mathbf{r}_m - \frac{\Delta}{2}\hat{x}, \mathbf{r}') dv' \right] \end{aligned} \quad (3.28)$$

In the above equations, ($n+$, and $n-$) correspond to the element n shifted $\Delta/2$ in the positive and negative x-direction, respectively.

For $m = n$, the relation [51]

$$\int_{v'} \nabla' g dv' = \int_{S'} g ds' \quad (3.29)$$

is used to transform the volume integral into a surface integral over the element surface S' and, thus, avoid the singularity point [52].

Using Green's function derivatives The integral in (3.17) can be approximated by summing over all elements $l = 1 \dots N$

$$\underline{E}^s(\mathbf{r}_m) = \sum_{l=1}^N \underline{V}^l(\mathbf{r}_m) \quad (3.30)$$

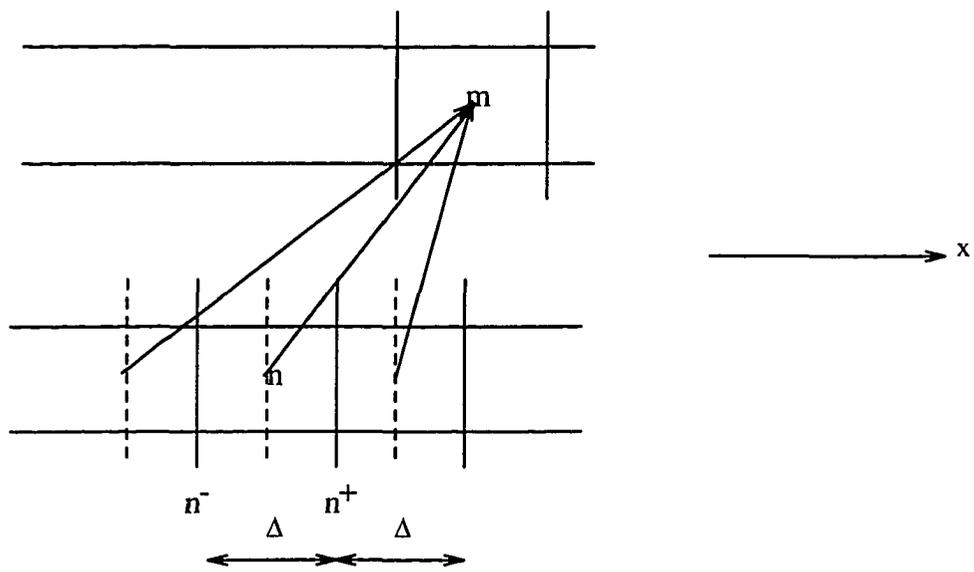


Figure 3.2: Green's function using current derivatives

where

$$\underline{V}'(\underline{r}_m) = -j\omega\mu \int_{v'_i} G(\underline{r}_m, \underline{r}') \cdot \underline{J}'(\underline{r}') dv' \quad (3.31)$$

This integral is handled by expressing it as a sum of 3 terms ($\underline{A}_{ml} + \underline{B}_{ml} + \underline{C}_{ml}$), where

$$\underline{A}_{ml} = -j\omega\mu \int_{v'_i-v'_e} G(\underline{r}_m, \underline{r}') \cdot \underline{J}'(\underline{r}') dv' \quad (3.32)$$

$$\underline{B}_{ml} = -j\omega\mu \int_{v'_e} [G \cdot \underline{J}'(\underline{r}') - G_0 \cdot \underline{J}'(\underline{r}_m)] dv' \quad (3.33)$$

$$\underline{C}_{ml} = -j\omega\mu \underline{J}'(\underline{r}_m) \int_{v'_e} \frac{-\hat{n}' \cdot \hat{R}'}{4\pi R^2} dv' \cdot \mathcal{I} \quad (3.34)$$

$G_0 = \frac{1}{k^2} \nabla \nabla \cdot \frac{1}{4\pi R}$, \hat{n}' is the unit outward directed normal, \hat{R}' is the unit vector of $(\underline{r}' - \underline{r})$, and \mathcal{I} is the unit dyad. The importance of this formulation is that v'_e does not need to be infinitesimal. This makes the numerical estimation of A_{ml} stable.

From Equation (3.21), the dyadic Green's function is written as

$$G = \frac{1}{k^2} \begin{bmatrix} k^2 + \frac{\partial^2}{\partial x_1'^2} & \frac{\partial^2}{\partial x_1' \partial x_2'} & \frac{\partial^2}{\partial x_1' \partial x_3'} \\ \frac{\partial^2}{\partial x_2' \partial x_1'} & k^2 + \frac{\partial^2}{\partial x_2'^2} & \frac{\partial^2}{\partial x_2' \partial x_3'} \\ \frac{\partial^2}{\partial x_3' \partial x_1'} & \frac{\partial^2}{\partial x_3' \partial x_2'} & k^2 + \frac{\partial^2}{\partial x_3'^2} \end{bmatrix} g \quad (3.35)$$

The second derivatives for G are given as [53]

$$\frac{\partial^2 g}{\partial x_m'^2} = k^2 \left[-\cos^2 \theta_m + \frac{j}{kR} \left(1 - \frac{j}{kR}\right) (3 \cos^2 \theta_m - 1) \right] g \quad m = 1, 2, 3 \quad (3.36)$$

$$\frac{\partial^2 g}{\partial x_m' \partial x_n'} = k^2 \cos \theta'_m \cos^2 \theta_n \left[-1 + \frac{3j}{kR} \left(1 - \frac{j}{kR}\right) \right] g \quad m \neq n \quad (3.37)$$

where

$$\cos \theta_m = (x'_m - x_m)/R$$

For two different elements, (3.30) can be estimated numerically using equations (3.36) and (3.37). For $l = m$ case, we have the component i of \underline{V} as

$$V_i^l(\underline{r}_l) = -j\omega\mu \left[\int_{v'_i} g J_i(\underline{r}') dv' + \frac{1}{k^2} \int_{v'_i} J_i \frac{\partial^2}{\partial x_i'^2} g dv' \right] \quad i = 1, 2, 3 \quad (3.38)$$

$$= -j\omega\mu I_1 + \frac{1}{k^2} I_2 \quad (3.39)$$

The singularity is encountered in the second integral I_2 .

Using equations (3.32), (3.33) and (3.34), the second integral I_2 in (3.38) can be written as

$$I_2 = B_{ii} + C_{ii} \quad (3.40)$$

where

$$C_{ii} = \frac{-1}{3} J_i(\underline{r}_l) \quad (3.41)$$

$$B_{ii} = \int_{v'_i} [J(\underline{r}') - J(\underline{r}_l)] \frac{\partial^2 g}{\partial x_i'^2} dv' + \int_{v'_i} J(\underline{r}) \frac{\partial^2 (g - g_0)}{\partial x_i'^2} dv' \quad (3.42)$$

$$= \frac{1}{3} [1 - e^{-jka} - jkae^{-jka}] \quad (3.43)$$

If $\underline{J}(\underline{r}')$ is expressed as a summation of pulse basis functions, we have

$$I_1 = \frac{1}{k^2} [jkae^{-jka} + e^{-jka} - 1] \quad (3.44)$$

$$I_2 = \frac{1}{3} [1 - e^{-jka} - jkae^{-jka}] - \frac{1}{3} J_i(\underline{r}_l) \quad (3.45)$$

These results are used to form the matrix equation that can be solved for the unknown equivalent currents. Applications of the computational methods described so far are presented next.

CHAPTER 4. APPLICATION OF HYBRID DIFFERENTIAL METHODS FOR HANDLING OPEN BOUNDARY PROBLEMS

The objective of applying numerical methods to differential equations is to obtain a solution with high accuracy within a realistic time frame. With the increase of the problem size, memory storage requirements and execution time are increased. The time increase is usually not a linear function of problem complexity. For example, in the case of the finite element method, the problem is recast to obtain a linear system of equations. Since the inversion of a matrix has a time complexity of order $O(N^3)$, a mesh with twice the number of unknowns should increase the time by a factor of 8. However, this problem becomes more serious as the matrix size becomes very large and cannot be stored in the main memory of the computer. Off-core solution methods have to be employed, in which case the solution time becomes affected by several factors such as, the type of the secondary storage, the physical distance between the computer and the storage, the traffic rate in the computer network, etc.

In the case of hybrid methods, we transform the original problem into solving multiple problems of smaller size. In some cases, the use of hybrid methods not only decreases the size of a big problem, but also reduces infinite size domains to a finite size that can be dealt with numerically. For example, electromagnetic scattering problems can have natural boundaries as in waveguide or cavity problems, whereas

in microwave scattering from an object, the field is defined in the infinite space and the size of the problem has to be reduced.

One solution of this problem is to use absorbing boundary conditions as discussed in Chapter 2 [34, 54, 55]. However, this technique requires the medium, external to the scatterer, to be completely homogeneous. In some important applications, the external medium is not homogeneous as in the case of scattering from an object embedded in multilayered media.

A hybrid finite element method for solving scattering problems can be achieved by dividing the space into two regions. The scatterer is enclosed by a hypothetical surface. The region interior to the surface is modeled using finite element analysis techniques to solve for the equivalent sources representing the scatterer. A transmission line model is then used for propagating the waves of the equivalent sources in the multilayered media.

Finite element hybrid model for 2D geometries

In this formulation, the problem domain is divided into interior and exterior regions using an arbitrary cylindrical surface enclosing the 2D object as shown in Figure 4.1. In the exterior region, the field is expressed in terms of the incident field ψ^{inc} and scattered field ψ^s , where the scattered field is represented as a weighted sum of orthogonal basis functions B_n such that

$$\psi^s(\rho, \phi) = \sum_{n=-N}^N A_n B_n(\rho, \phi) \quad (4.1)$$

One form of basis functions B_n is the scattered field of a line source that lies in the interior region as shown in [37]. Another form is to use Hankel functions as

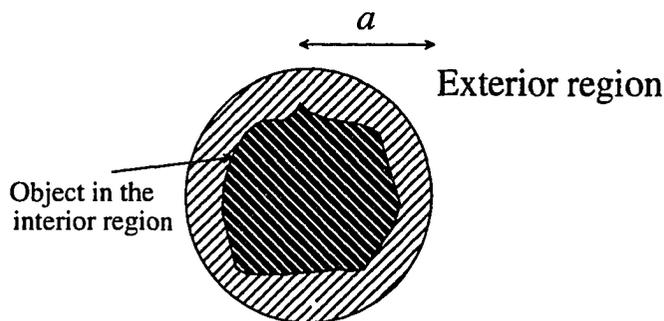


Figure 4.1: Interior and exterior regions

shown in [36, 56] for scattering of an object in free space. Free space basis functions B_n^f are given as

$$B_n^f = e^{jn\phi} H_n(k\rho) \quad (4.2)$$

In the interior region, the field is represented in terms of basis components Λ_n such that

$$\psi(\rho, \phi) = \sum_{n=-N}^N a_n \Lambda_n(\rho, \phi) \quad (4.3)$$

Denoting the surface value of the field corresponding to Λ_n as ψ_n , we have the surface value of the interior field given by

$$\psi[\text{on } S] = \sum_{n=1}^N a_n \psi_n \quad (4.4)$$

The basis functions ψ_n are chosen to be in the form $[\exp(jn\phi)]$ for circular cross section surfaces. The region interior to the surface is discretized into small subregions or finite elements as shown in Figure 4.2. The finite element method is used to model the interior region, as shown in Chapter 2, to obtain the solution of Λ_n corresponding to each ψ_n .

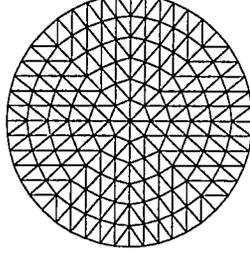


Figure 4.2: Discretization of the internal region

After solving the Dirichlet problem for the interior region for each ψ_n , the boundary conditions are applied between the interior and exterior regions. This can be written as

$$\int_0^{2\pi} \sum_{-N}^N a_n \Lambda_n(\rho(S), \phi) \zeta_m(\phi) d\phi = \int_0^{2\pi} \left\{ \sum_{-N}^N A_n e^{jn\phi} H_n^{(2)}(k_0 \rho(S)) + \psi^{inc}(\rho(S), \phi) \right\} \zeta_m(\phi) d\phi \quad (4.5)$$

$$\int_0^{2\pi} \sum_{-N}^N a_n \frac{\partial \Lambda_n}{\partial \rho}(\rho(S), \phi) \zeta_m(\phi) d\phi = \int_0^{2\pi} \left\{ \sum_{-N}^N A_n e^{jn\phi} \frac{\partial H_n^{(2)}}{\partial \rho}(k_0 \rho(S)) + \frac{\partial \psi^{inc}}{\partial \rho}(\rho(S), \phi) \right\} \zeta_m(\phi) d\phi \quad (4.6)$$

where the weighting functions ζ_m are chosen to be $\exp(jm\phi)$, $m = -N, \dots, N$

Equations (4.5) and (4.6) can be expressed as a matrix equation with the coefficients of the expansions in the interior and exterior regions as the unknowns.

$$\left[\begin{array}{c} G \end{array} \right] \left[\begin{array}{c} a_{-N} \\ \cdot \\ a_N \\ \text{---} \\ A_{-N} \\ \cdot \\ A_N \end{array} \right] = \left[\begin{array}{c} \cdot \\ f(a) \\ \cdot \\ \text{---} \\ \cdot \\ F(a) \\ \cdot \end{array} \right] \quad (4.7)$$

where:

$$[G] = \left[\begin{array}{cccc|cccc} 2\pi & & & & 2\pi H_{-N}^{(2)}(k_0 a) & & & & \\ & \cdot & & & & & \cdot & & \\ & & \cdot & & & & & \cdot & \\ & & & 2\pi & & & & & 2\pi H_N^{(2)}(k_0 a) \\ \text{---} & \text{---} \\ \cdot & \cdot & \cdot & \cdot & 2\pi H_{-N}^{(2)'}(k_0 a) & & & & \\ \cdot & \cdot & \cdot & \cdot & & & \cdot & & \\ \cdot & \Lambda'_{nm} & \cdot & \cdot & & & & 2\pi H_n^{(2)'}(k_0 a) & \\ \cdot & \cdot & \cdot & \cdot & & & & & \cdot \end{array} \right]$$

$$f_n(a) = \int_0^{2\pi} \psi^{inc} e^{jn\phi} d\phi$$

$$F_n(a) = \int_0^{2\pi} \frac{\partial \psi^{inc}}{\partial \rho} e^{jn\phi} d\phi$$

$$\Lambda'_{nm} = \int_0^{2\pi} \frac{\partial \Lambda_n}{\partial \rho}(a, \phi) e^{jm\phi} d\phi$$

The derivative $\frac{\partial \Lambda_n}{\partial \rho}(a, \phi)$ is obtained from the boundary elements in terms of the nodal values of the field.

The values for the coefficients obtained by solving equation (4.7) are used in equations (4.1) and (4.3) to obtain the field in the interior and exterior regions.

To validate the model, results are obtained for the bistatic scattering width $W(\phi)$ of conducting and dielectric cylinders with a TM^y incident plane wave, where y is along the axial direction. The bistatic scattering width is defined as

$$W(\phi) = \lim_{\rho \rightarrow \infty} 2\pi\rho \left| \frac{\psi^s(\rho, \phi)}{\psi^{inc}} \right|^2 \quad (4.8)$$

Comparisons with analytical results are shown in Figures 4.3 and 4.4. In Figure 4.3, a bistatic scattering width is compared with exact values [1], for the case of conducting cylinders with radii $R=0.158\lambda$, 0.4λ , and 0.6λ , respectively. In Figure 4.3, the comparison is done for a dielectric shell with an inner radius of 0.25λ and outer radius of 0.3λ [45]. Figures 4.5-4.12 show the polar plots of the model bistatic width for different cylindrical cross sections.

A direct application of this model to the geometry of a scatterer in multilayered media is described next.

Electromagnetic scattering from objects embedded in multilayered media

The modeling of electromagnetic scattering from objects or inhomogeneities embedded in multilayered media is a subject of significant research interest in a number of disciplines. Such models have a wide range of applications ranging from geophysical exploration to radar inspection of masonry structures. In NDE applications, the embedded object can simulate a crack or a reinforcement bar.

For the case of an object embedded in multilayered media, the incident field has to be propagated through multiple layers. For example, for stratified media in the xy

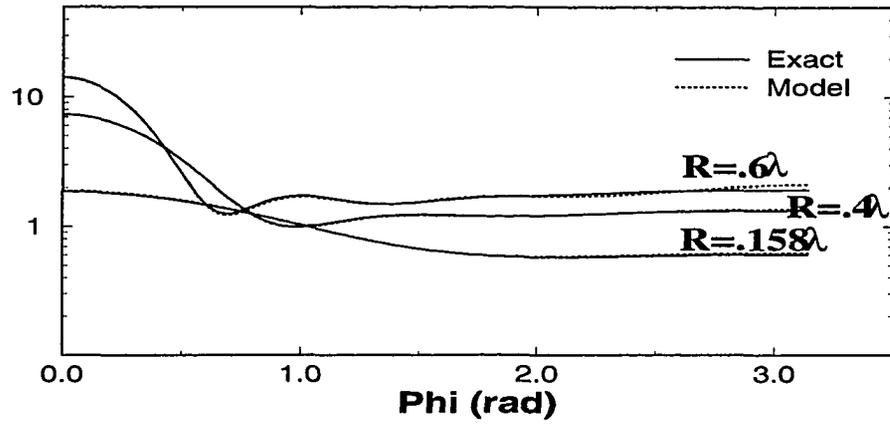


Figure 4.3: Bistatic width for conducting cylinders

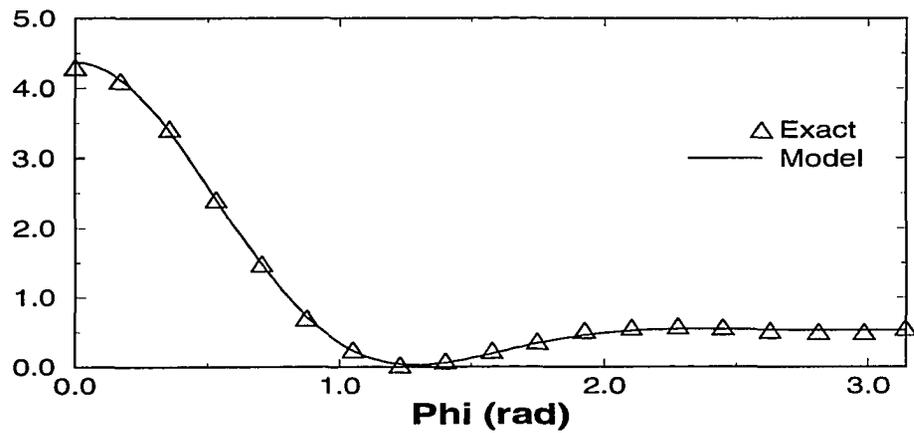


Figure 4.4: Bistatic width for dielectric cylinder

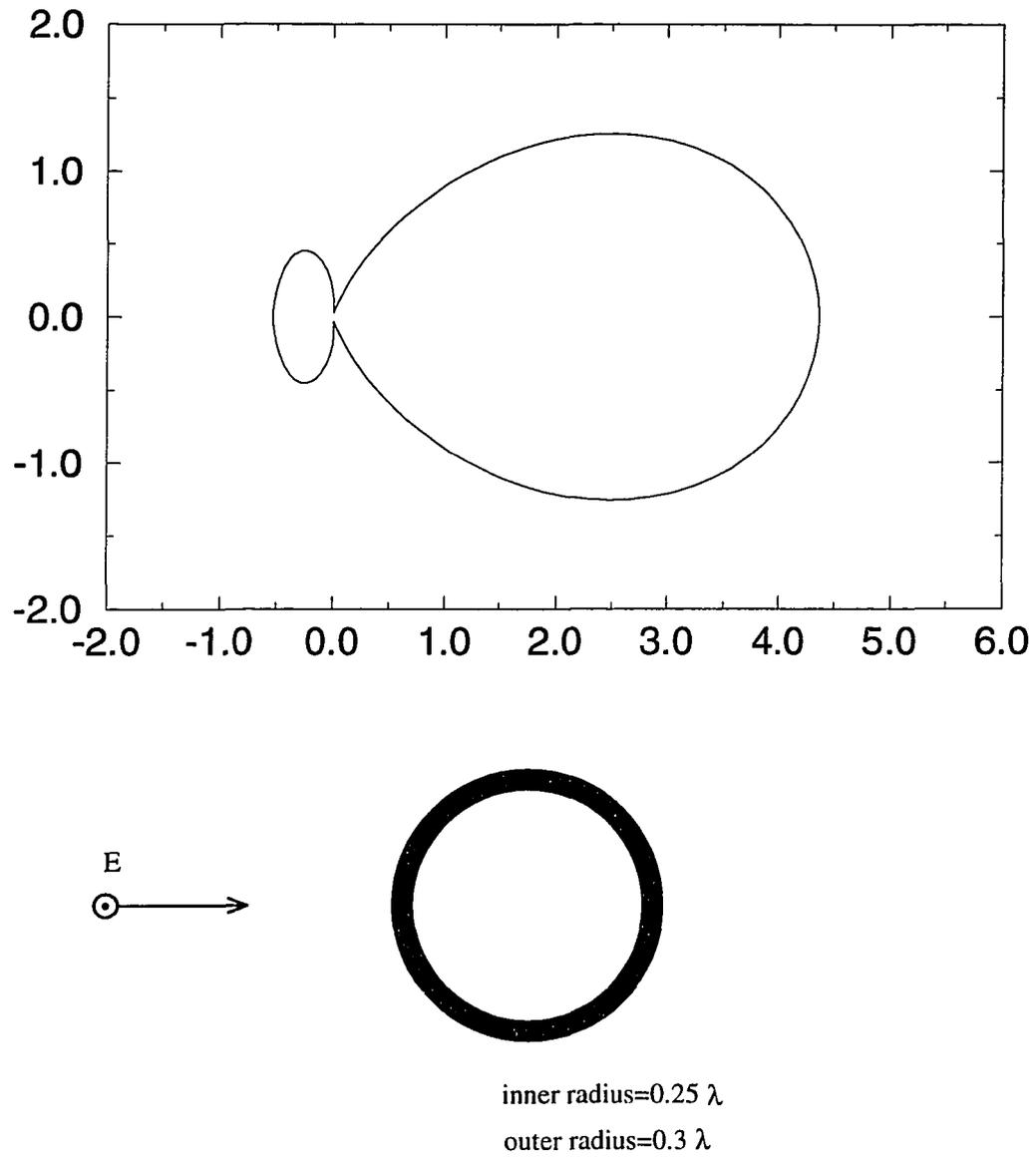


Figure 4.5: Bistatic width for a dielectric shell with a TM polarized incident plane wave

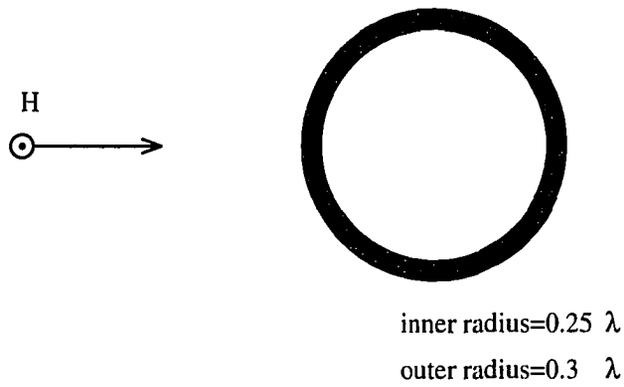
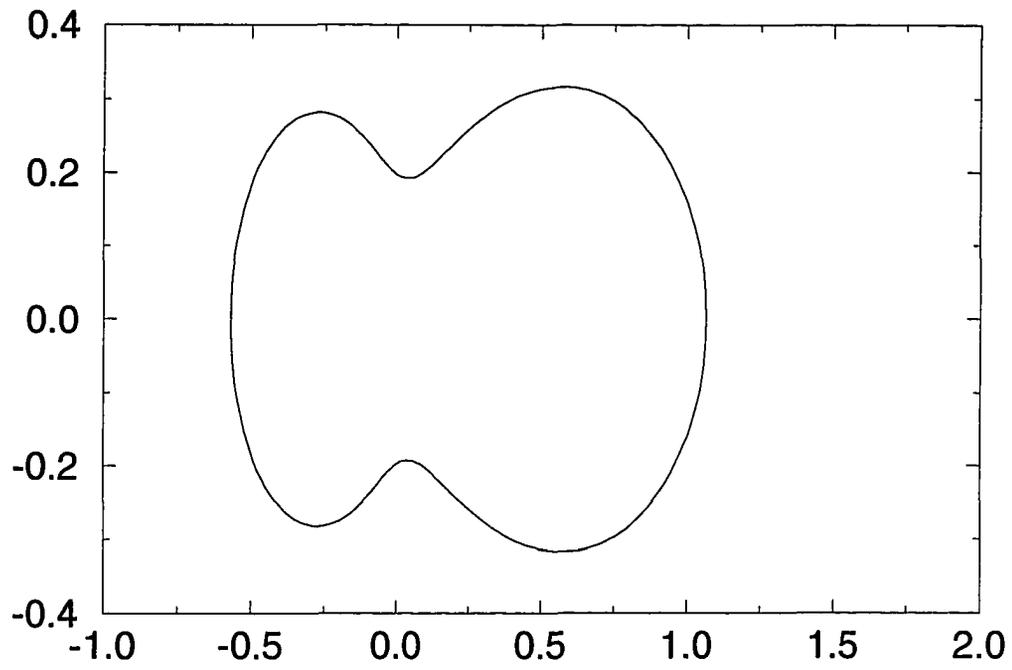


Figure 4.6: Bistatic width for a dielectric shell TE polarization

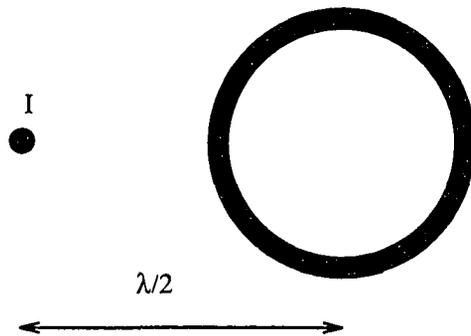
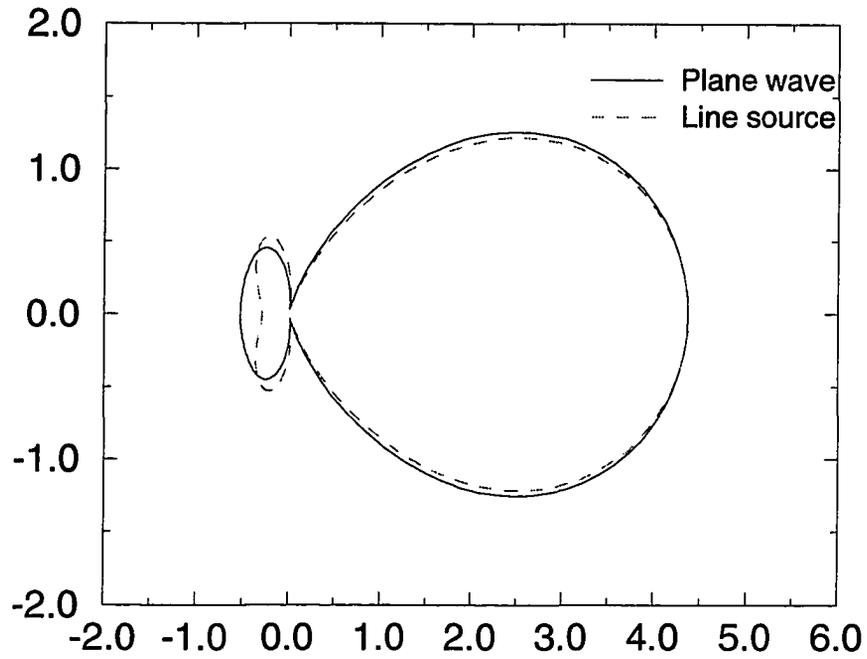


Figure 4.7: Comparing the bistatic width of a dielectric shell on $\lambda/2$ distance from a line source with the bistatic width with a plane wave excitation

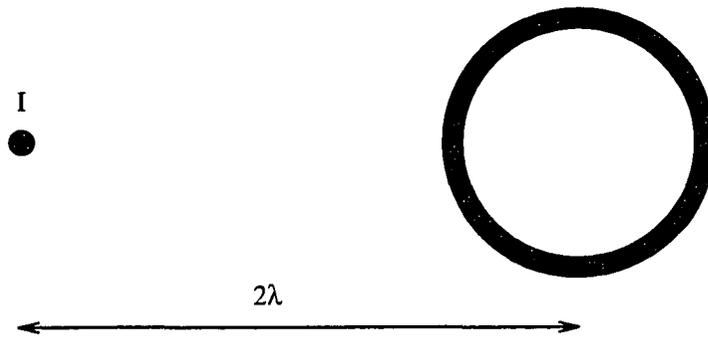
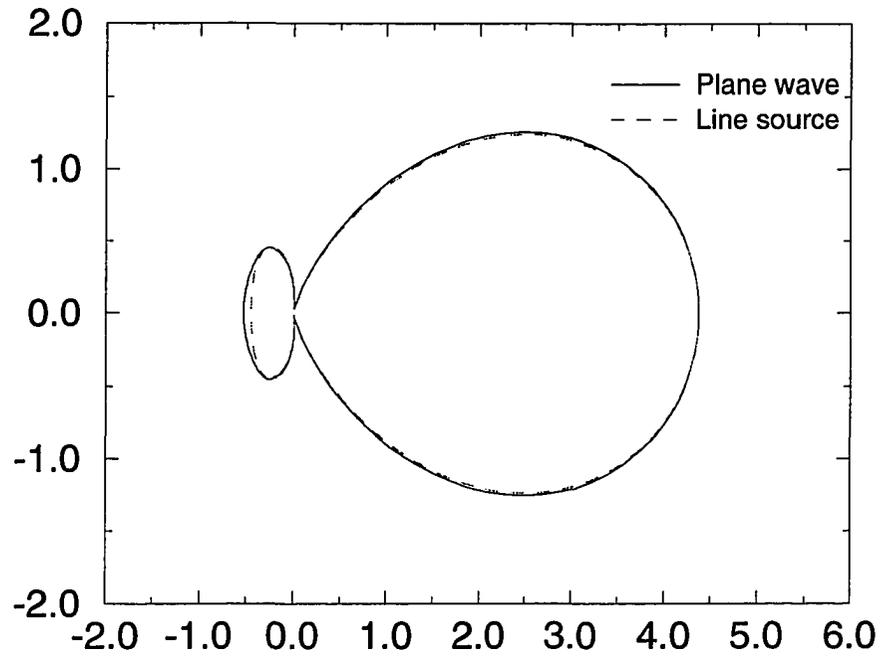


Figure 4.8: Comparing the bistatic width for a dielectric shell on 2λ distance from a line source with the bistatic width with a plane wave excitation

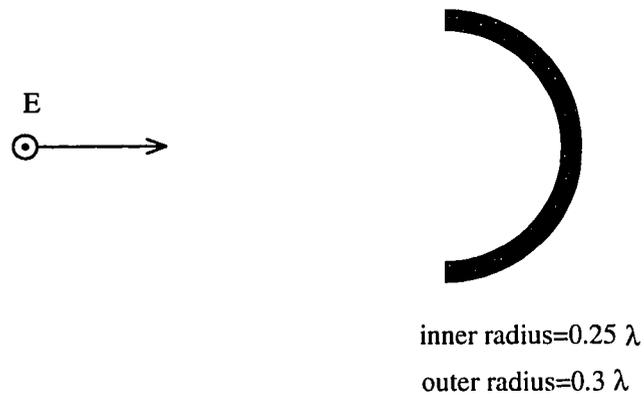
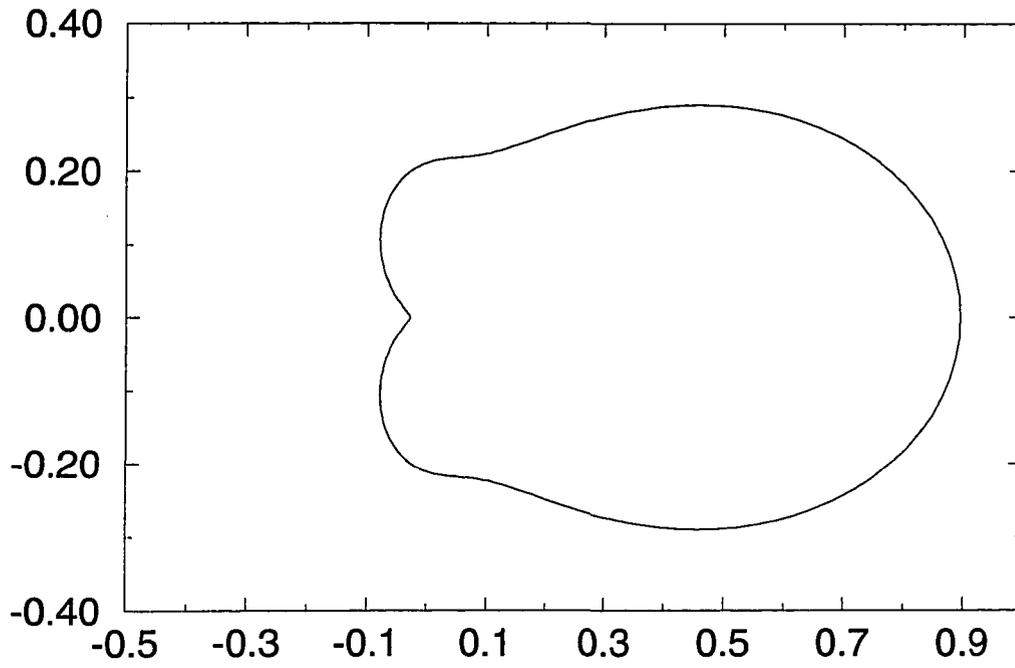


Figure 4.9: Half a shell with a plane TM polarized wave

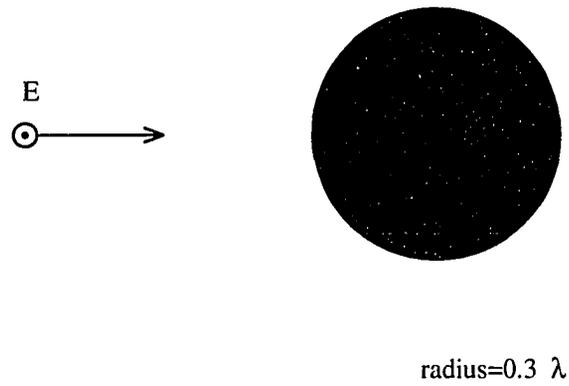
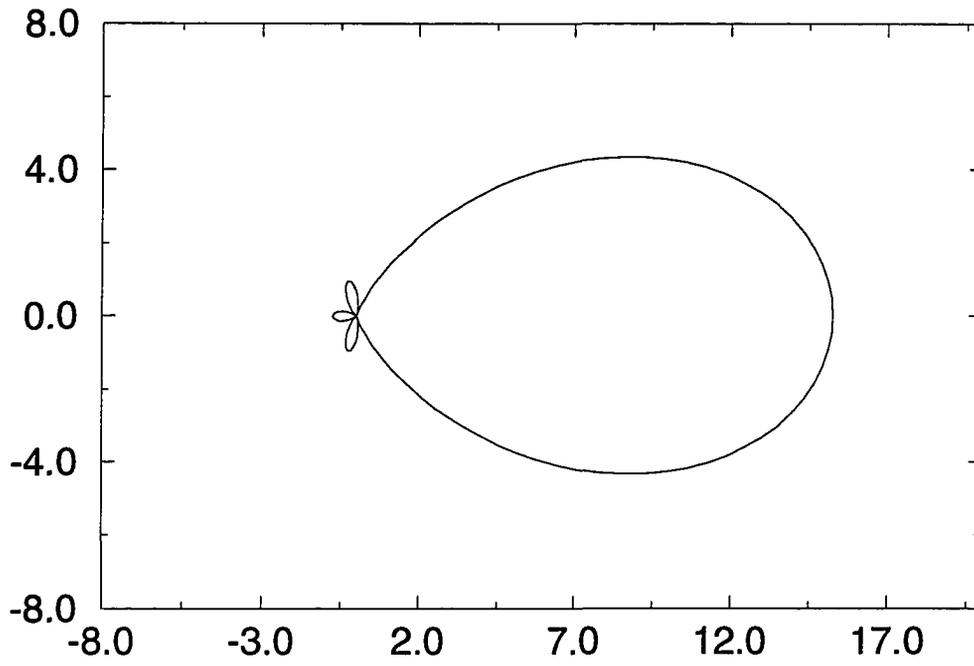
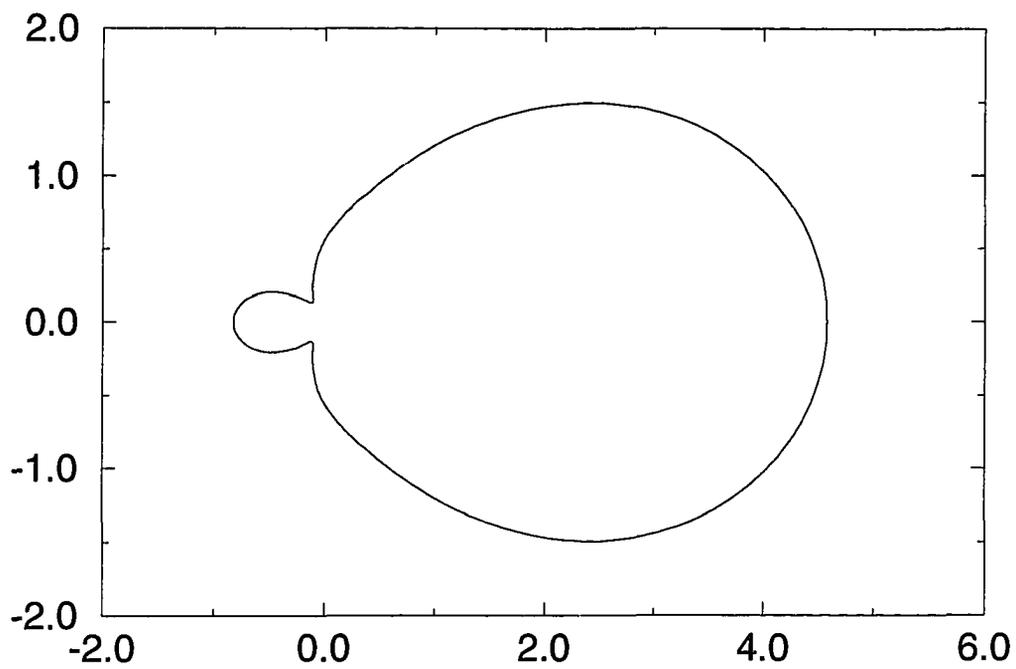


Figure 4.10: Bistatic width for a solid cylinder with a TM polarized plane wave



radius=0.3 λ

Figure 4.11: Bistatic width for half a solid cylinder with a TM polarized plane wave. Incident angle is π .

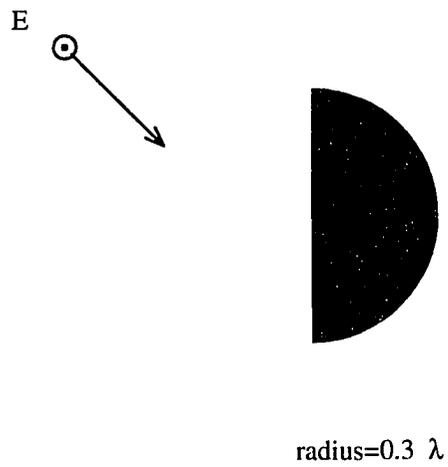
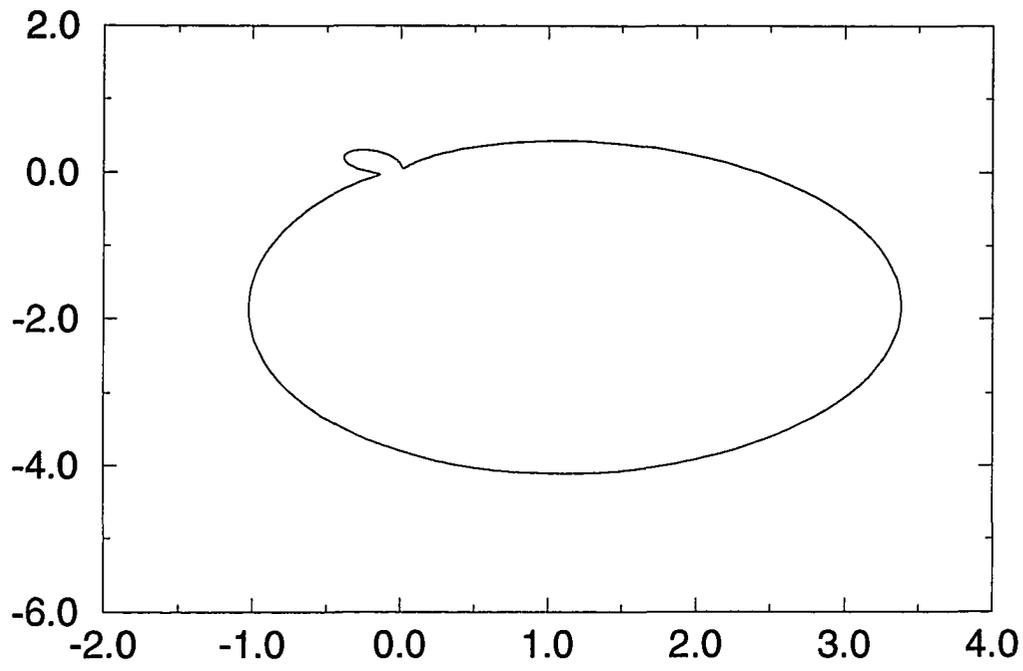


Figure 4.12: Bistatic width for half a solid cylinder with a TM polarized plane wave. Incident angle is $3\pi/4$

plane, assuming a plane wave incident in the positive z -direction as shown in Figure 4.13, there will be reflection from the layered media in the negative z -direction, and transmission in the positive z direction. For example for a TM^y polarized plane wave, the electric field in the m th layer can be described by

$$E_{m,y}(z, x) = [a_m e^{-u_m z} + b_m e^{u_m z}] e^{-j\lambda x} \quad (4.9)$$

where $u_m^2 = (\lambda^2 - k_m^2)$

The boundary conditions at $z = z_m$ are given by

$$E_{m-1,y} = E_{m,y} \quad (4.10)$$

$$\hat{Y}_{m-1} \frac{\partial E_{m-1,y}}{\partial z} = \hat{Y}_m \frac{\partial E_{m,y}}{\partial z} \quad (4.11)$$

where $\hat{Y}_m = j\omega\mu_m$

The solutions for the unknowns which are the wave amplitudes a_m along the positive z direction, and the amplitudes b_m along the negative z direction are obtained using a transmission line analogy. The reflection coefficient $\mathcal{R}_m = \frac{b_m}{a_m}$ is obtained from the characteristic impedance of the m th layer and the input impedance to the next layer. For TM waves, we have [57]

$$\mathcal{R}_m = \frac{b_m}{a_m} = \frac{N_m - Y_{m+1}}{N_m + Y_{m+1}} \quad (4.12)$$

where

$$Y_m = N_m \frac{Y_{m+1} + N_m \tanh(u_m h_m)}{N_m + Y_{m+1} \tanh(u_m h_m)} \quad (4.13)$$

$$N_m = \frac{u_m}{\hat{Y}_m} \quad (4.14)$$

$$Y_M = N_M \quad (4.15)$$

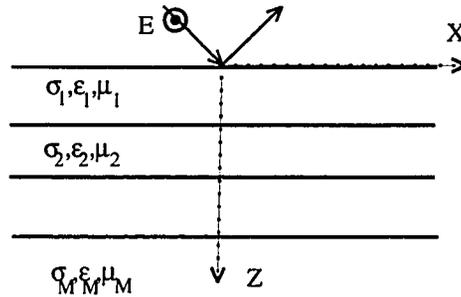


Figure 4.13: Propagation of a plane wave in multilayered media

and h_m is the thickness of layer m .

This is analogous to a transmission line that can be used to find the a_m and b_m coefficients [57].

For a scatterer in the M th layer, the effect of the multilayered media can be included by considering the basis functions in equation (4.1) as

$$B_n = B_n^f + \int_{-\infty}^{\infty} \mathcal{R}_M(\lambda) \hat{B}_n^f e^{-u_M z} e^{-j\lambda x} d\lambda \quad (4.16)$$

where B_n^f is given in equation (4.2), and

$$\hat{B}_n^f = \int_{-\infty}^{\infty} B_n(x, z_M) e^{j\lambda x} dx$$

For the general case when the incident field is not a plane wave, the incident field is numerically decomposed into a sum of plane waves. Each plane wave is propagated into the multilayered media, to find a field component in the layer of the inhomogeneity. The total incident field on the scatterer is found by summing the

different components. For example, for a field distribution given by

$$E_y(z = -h) = \begin{cases} \cos \frac{\pi x}{a} & |x| < \frac{a}{2} \\ 0 & |x| > \frac{a}{2} \end{cases} \quad (4.17)$$

the incident field due to this source can be expressed as [46]

$$E_y(x, z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(\lambda) e^{-j\lambda x} e^{-jk_z z} d\lambda \quad (4.18)$$

where f is found to be

$$f(\lambda) = \frac{2\pi a \cos(\lambda a/2)}{\pi^2 - (\lambda a)^2} \quad (4.19)$$

and

$$k_z = \begin{cases} -j\sqrt{\lambda^2 - k^2} & k < |\lambda| \\ \sqrt{k^2 - \lambda^2} & k > |\lambda| \end{cases} \quad (4.20)$$

The next step is to obtain the scattered field.

The total field in each element of the scatterer is first used to determine the current in each element, and consequently the scattered field in the exterior region is obtained as a sum of line sources embedded in multilayered media. A line source field is decomposed into plane waves. For example, for a line source at $z = h$, the primary field can be written as

$$E^p = \int_{-\infty}^{\infty} A(\lambda) e^{\pm jk_z(z-h)} e^{-j\lambda x} dx \quad (4.21)$$

$A(\lambda)$ represents the plane waves decomposition and it is found to be

$$A(\lambda) = \frac{-\omega \mu I}{2\pi k_z} \quad (4.22)$$

where $k_z = (k^2 - \lambda^2)^{1/2}$

The total field in the m th layer is finally found by numerically evaluating the sum of integrations

$$E_{my} = \sum_{\text{elements}} \int_{-\infty}^{\infty} \left[a_m(\lambda) e^{-jk_z m z} + b_m(\lambda) e^{jk_z m z} \right] e^{-j\lambda x} d\lambda \quad (4.23)$$

where a_m and b_m correspond to plane waves propagating in the positive and negative z directions, respectively.

Computing the integral of equation (4.23) is usually difficult because of the large oscillations of the integrand for large values of x and z . However, the problem can be alleviated by using the stationary phase method to approximate the integral.

Stationary phase method

Generally for an integral of the form [21, 58]

$$I(x) = \int_a^b f(t) e^{jx\psi(t)} dt \quad (4.24)$$

the largest contribution to the integral for large values of x comes from around the stationary phase point t_{sp} corresponding to

$$\psi'(t_{sp}) = 0. \quad (4.25)$$

Thus, we have

$$\begin{aligned} I(x) &= \int_{t_{sp}-\epsilon}^{t_{sp}+\epsilon} f(t) e^{jx\psi(t)} dt \\ &\sim f(t_{sp}) \int_{t_{sp}-\epsilon}^{t_{sp}+\epsilon} e^{jx[\psi(t_{sp}) + \frac{t^2}{2}\psi''(t_{sp})]} dt \\ &\sim f(t_{sp}) e^{jx\psi(t_{sp})} \int_0^{\infty} e^{\frac{\psi''(t_{sp})}{2} t^2} dt \\ &= f(t_{sp}) e^{jx\psi(t_{sp})} \sqrt{\frac{\pi}{2}} \frac{e^{\pm i\frac{\pi}{4}}}{\sqrt{|\psi''(t_{sp})|}} \end{aligned} \quad (4.26)$$

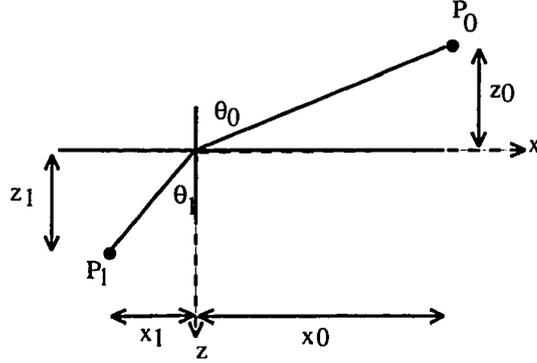


Figure 4.14: Getting the field at point P_0 due to a line source at P_1

To illustrate the use of the method, we consider a simple example of finding the field at point P_0 in media 0, due to a line source P_1 in media 1, as illustrated in Figure 4.14.

The field at P_0 can be written as

$$E = \int_{-\infty}^{\infty} A(\lambda) e^{-jk_{z1}z_1} e^{-jk_{z0}z_0} e^{-j\lambda x} dx \quad (4.27)$$

where $A(\lambda)$ is obtained from equation (4.22).

The stationary phase value of λ (λ_{sp}) is obtained from

$$\frac{d}{d\lambda} [-jk_{z0}z_0 - jk_{z1}z_1 - j\lambda_{sp}x] = 0 \quad (4.28)$$

where $k_{z0} = \sqrt{(k_0^2 - \lambda^2)}$, and $k_{z1} = \sqrt{(k_1^2 - \lambda^2)}$

λ_{sp} is found to be

$$\begin{aligned} \lambda_{sp} &= k_0 \frac{x_0}{\sqrt{x_0^2 + z_0^2}} = k_0 \sin \theta_0 \\ &= k_1 \frac{x_1}{\sqrt{x_1^2 + z_1^2}} = k_1 \sin \theta_1 \end{aligned} \quad (4.29)$$

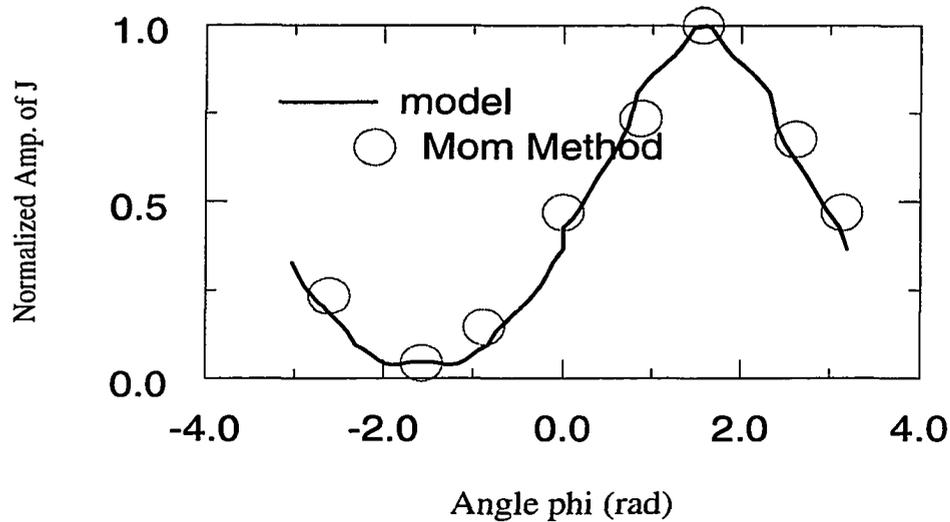


Figure 4.15: Amplitude of normalized surface current vs. angle ϕ

The value of λ_{sp} in equation (4.29) shows that when the distance between the line source and the measuring point is large, most of the contribution to the integral of equation (4.27) comes from a plane wave that has angles of propagation satisfying Snell's law. The amplitude of this plane wave is obtained from equation (4.26).

Considering a conducting cylinder of radius 0.175λ , embedded at a depth of 0.175λ from the surface of a dielectric layer of relative permittivity $\epsilon_r = 4$, the model is used to find the normalized current density J_z on the cylinder surface for an incident TM^y polarized plane wave normal to the layer's surface. Figure 4.15 gives the amplitude of the current density, normalized to its maximum as a function of the angle ϕ from the horizontal direction. Results are compared with those obtained using the moment method given in [59].

CHAPTER 5. APPLICATION OF INTEGRAL MOMENT METHOD WITH REDUCED STORAGE

In the previous chapters we have seen that in the finite element method, the symmetry, banded characteristic of the stiffness matrix are exploited to achieve low memory storage requirements while hybrid techniques are invoked to keep the FE domain finite. In this chapter we consider a different approach using integral equations. In this approach, the moment method is used to form a linear system of equations. This system of equations produces a dense matrix and this is reflected in the enormous storage requirements even for moderate size problems. For example, if the problem involves a volume discretization of $10 \times 10 \times 10$ elements, with 3 degrees of freedom for the unknown in each element, the matrix size is 3000×3000 , and the storage requirement for the system matrix with double precision complex number unknowns will be 9×16 mega bytes. This chapter focuses on techniques for solving the integral equations while limiting the storage requirements for the problem.

Consider the volume integral equation

$$\underline{E}^s = -j\omega\mu \int_{v'} G(\underline{r}, \underline{r}') \cdot \underline{J}_{eq}(\underline{r}') dv' \quad (5.1)$$

where E^s is the scattered field, G is Green's tensor described in Chapter 3, and \underline{J}_{eq} is the electric current given by

$$\underline{J}_{eq}(\underline{r}) = j\omega[\epsilon(\underline{r}) - \epsilon_1] \quad (5.2)$$

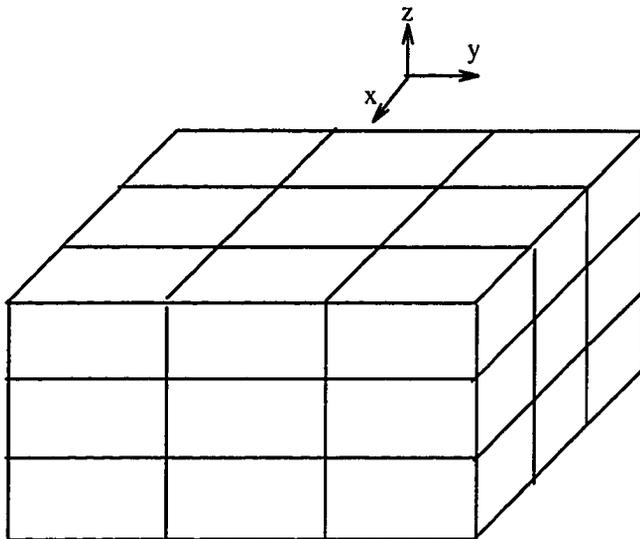


Figure 5.1: Volume discretization of the scatterer

The above integral equation is to be solved for the electric current density which is confined to the finite domain of the scatterer. The unknown \underline{J}_{eq} is discretized using the moment method. The scatterer is divided into N volume elements as illustrated in Figure 5.1, and a set of N pulse basis functions P_i is used such that

$$P_i(\underline{r}) = \begin{cases} 1 & \text{for } \underline{r} \in \text{element } i \\ 0 & \text{elsewhere} \end{cases} \quad (5.3)$$

The unknown is represented as

$$\underline{J}_{eq}(\underline{r}) = \sum_{i=1}^N \sum_{k=1}^3 J_i^k P_i(\underline{r}) \hat{u}_k \quad (5.4)$$

where \hat{u}_k , $k = 1, 2, 3$ represents a unit vector in the x, y and z directions, respectively.

Equation (5.1) is finally cast in the form

$$[Z]\underline{J} = \underline{E}^s \quad (5.5)$$

where:

The current vector \underline{J} consists of elements $J_{3i} = J_{eq_x}(\underline{r}_i)$, $J_{3i+1} = J_{eq_y}(\underline{r}_i)$, and $J_{3i+2} = J_{eq_z}(\underline{r}_i)$ for element $1 < i < N$. The point \underline{r}_i is located at the center of the i th element.

Similarly, the RHS vector consists of subvectors of the three components of the scattered field at the measurement point \underline{r}_j representing the center of the j th element.

The matrix Z consists of submatrices Z_{ij} , and

$$Z_{ij} = -j\omega\mu G(\underline{r}_i, \underline{r}_j)v_i \quad (5.6)$$

v_i is the volume of the i th element.

Since equation (5.1) is in the form of a convolution equation, the matrix $[Z]$ in equation (5.5) is a Toeplitz matrix, which means that its elements depend only on the difference between its two indices rather than their absolute value. This suggests the application of an iterative solution with the use of Fast Fourier Transform (FFT) in estimating the result of the matrix vector multiplication in each iteration. In this approach, a rectangular box is discretized uniformly in each of its three directions. An element that lies in the scatterer is given a material attribute corresponding to that of the scatterer. An element that lies outside the scatterer is given the material attribute of the surrounding medium, and from equation (5.2), its unknown equivalent currents will be zero.

The discrete Fourier transform DFT is found for the three components of the unknown current sampled at the elements centers. Considering a three dimensional

discretization of dimension $L_1 \times L_2 \times L_3$, the DFT of the x-component of current, \hat{J}_x is defined as

$$\hat{J}_x(k_1, k_2, k_3) = \sum_{n_1=0}^{2L_1-1} \sum_{n_2=0}^{2L_2-1} \sum_{n_3=0}^{2L_3-1} J_x(n_1, n_2, n_3) e^{-j2\pi(\frac{k_1 n_1}{2L_1} + \frac{k_2 n_2}{2L_2} + \frac{k_3 n_3}{2L_3})} \quad (5.7)$$

where $J_x(n_1, n_2, n_3)$ is zero padded such that it is equal to 0 in elements outside of the object, and equal to the sample of the x-component of the current at the center of each element n for $n < N$, as explained next.

The Green's tensor consists of nine vectors whose elements are first formed and stored in the spatial domain and then replaced with their DFT values. However, due to symmetry, only six vectors corresponding to the 3-diagonal and 3-off-diagonal elements are stored.

The DFT of the result vector is found by multiplying the stored values of the DFT of Green's tensor by the corresponding DFT values of the current calculated in each iteration. The result vector is found by calculating the inverse discrete Fourier transform of the three product vectors.

Since the DFT of a non-periodic signal of length N is the same as that of a periodic signal with period N , the calculation of the convolution equations will have aliasing errors if the signals are not zero-padded. In order to avoid aliasing during convolution of two signals of length N_1 and N_2 , the signals have to be zero padded to length $N_1 + N_2 - 1$ [60]. However, in the application under consideration, the unknown values of interest are at the centers of the elements, and the padding is required to extend only to twice the number of elements. Figure 5.2 shows a one dimensional illustration of the current and Green's function. The result of convolution at the centers of the four elements of this problem is the same as that obtained using the periodic signal in Figure 5.3.

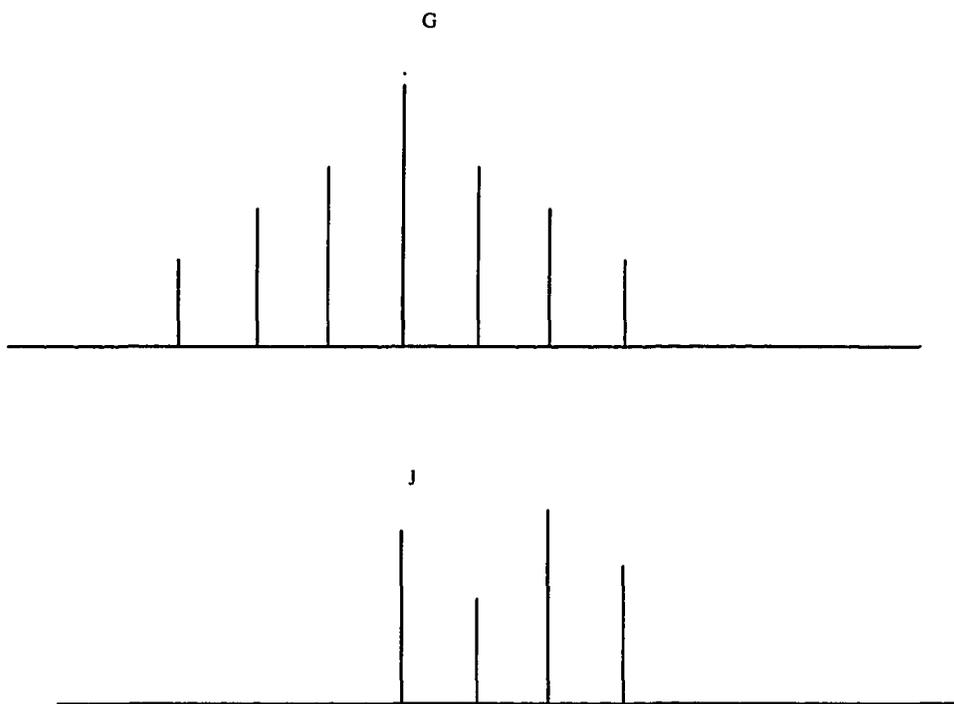


Figure 5.2: One dimensional illustration of Green's function G and electric equivalent current J sampled at the centers of elements

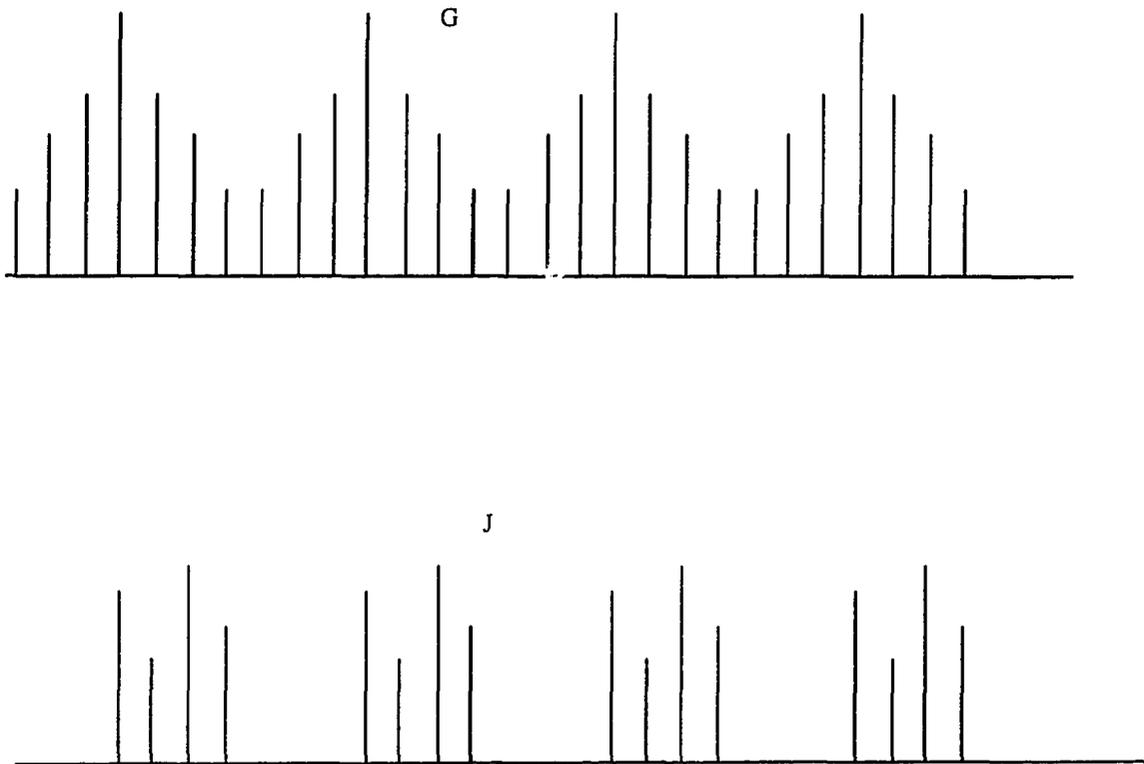


Figure 5.3: Repeated form of one dimensional Green's function G and electric equivalent current J

In the implementation, six vectors \underline{G}^{mn} , $m \leq n$ of Green's tensor are formed where

$$\underline{G}^{mn} = G^{mn}(r_1, r_l) \quad 1 \leq l \leq N \quad (5.8)$$

where $G^{mn}(r_1, r_l)$ is obtained from equation (3.35) and r_1 is the origin.

Symmetry of $G^{mn}(r_1, r_l)$ is invoked to reduce the storage requirements of the vectors \underline{G}^{mn} .

The DFT is found using the FFT technique, and the results are used to replace the values of the \underline{G}^{mn} vectors. Equation (3.35) shows that the diagonal elements of Green's tensor have even symmetry while the off-diagonal elements are of odd-symmetry. The symmetry is used to reduce the required storage to half the size of arrays.

In the iterative solution scheme, the algorithm described above finds the result of the multiplication of $[Z]$ matrix times the unknown vector. The storage requirements for an N element problem is reduced from $9N^2$ for direct solution algorithms to $6N$ complex numbers. However, an efficient technique has to be used to reduce the number of required iterations. The next section explains the conjugate gradient method used to solve the linear system of equations, in a minimal number of iterations.

Conjugate gradient method

In the conjugate gradient method, the solution of the linear system of equations $[A]_{L \times L} \underline{X}_{L \times 1} = \underline{Y}_{L \times 1}$, is obtained iteratively. An initial guess vector \underline{X}_0 is iteratively improved by minimizing the error successively in the direction of each vector of a set of *direction vectors* $\{\underline{P}_n\}$, $n=1,2,..L$. If the \underline{P} vectors are A-orthogonal or *A-conjugate*, the number of iterations will be finite and equal to the number of eigenvalues of the

matrix $[A]$, assuming there is no round-off error. The set \underline{P} is A-conjugate if the inner product $\langle [A]\underline{P}_i, \underline{P}_j^*[A] \rangle$ is 0 for all $i \neq j$.

If the vectors \underline{P} are obtained as the A-orthogonalization of the unit coordinate vectors, we obtain the popular Gaussian elimination method. However, if we have the set $\{ \underline{P}_n \}$, such that the residual vectors \underline{R}_n are mutually orthogonal, i.e.,

$$\langle \underline{R}^i, \underline{R}^j \rangle = 0 \quad i \neq j$$

we obtain the conjugate gradient method CGM [61].

The update equation is given as

$$\underline{X}_{n+1} = \underline{X}_n + t_n \underline{P}_n \quad (5.9)$$

and the residuals are given by

$$\underline{R}_{n+1} = \underline{R}_n + t_n [A] \underline{P}_n \quad (5.10)$$

where t_n is a step coefficient such that it minimizes the error E_n given by the norm of \underline{R}_{n+1} , i.e. $E_n = \|\underline{R}_{n+1}\|^2$. This gives

$$t_n = - \frac{\|[A]^a \underline{R}_n\|^2}{\|[A] \underline{P}_n\|^2} \quad (5.11)$$

where $[A]^a$ is the adjoint of matrix $[A]$.

The direction vectors are obtained iteratively as

$$\underline{P}_{n+1} = -[A]^a \underline{R}_{n+1} + q_n \underline{P}_n \quad (5.12)$$

where

$$q_n = \frac{\|[A]^a \underline{R}_{n+1}\|^2}{\|[A]^a \underline{R}_n\|^2}$$

After n iterations, the error $\|\underline{X}_n - \underline{X}_{\text{exact}}\|$ satisfies the relation [61]

$$\frac{\|\underline{X}_n - \underline{X}_{\text{exact}}\|}{\|\underline{X}_0 - \underline{X}_{\text{exact}}\|} \leq 2 \left(\frac{\sqrt{B} - \sqrt{b}}{\sqrt{B} + \sqrt{b}} \right)^n \quad (5.13)$$

where B and b are the largest and smallest eigenvalues of the matrix $[A]$, respectively.

Although the number of iterations is guaranteed not to exceed L , neglecting the effect of round-off, the required number of iterations is usually less than L . A good initial guess has a significant effect on reducing the required number of iterations. In scattering problems, we use Born approximation to provide the initial current due to the incident electric field which is usually taken as the initial guess. The rule of thumb typically gives the required number of iterations as $L/6$, [48].

Conjugate gradient fast Fourier transform

The procedure can be summarized as follows

- The object is divided into $L_1 \times L_2 \times L_3$ elements.
- Form 6 Green's function arrays $\underline{g}^i[L1][L2][L3]$, $1 < i < 6$ (corresponding to diagonal and off diagonal elements of Green's tensor) between element 1, and all other elements.
- Use symmetry to store the values of Green's function arrays $\underline{g}_1^i[2L1][2L2][2L3]$ between element 1 and all elements in an extended object of dimension $2L_1 \times 2L_2 \times 2L_3$.
- Obtain the FFT of the stored Green's function $G^i[2L1][2L2][2L3]$, $1 < i < 6$. In implementation, only half of the arrays of Green's function and its FFT are required to be stored due to symmetry.

- Start with a random value for the three components of the current, $\underline{x}_0^j[L1][L2][L3]$, $1 < j < 3$, using Born approximation.
- Zero pad the current in the extended array to avoid aliasing, $\underline{x}_0^i[2L1][2L2][2L3]$.
- Obtain the FFT of the current and get its product with the stored Green's function FFT.

$$Y^i[2L1][2L2][2L3] = \sum_{j=1}^3 X_0^j[2L1][2L2][2L3] G^m[2L1][2L2][2L3]$$

where $1 < m < 6$ depends on i and j .

- Take the IFFT to get the product arrays y^i , where

$$y^i(n_1, n_2, n_3) = \mathcal{F}^{-1}\{Y(k_1, k_2, k_3)\}$$

- Apply conjugate gradient method to update the solution and check for convergence.

The algorithm was implemented on a simple problem of scattering from a sphere for which an exact solution is given by Mie theory [62]. Figure 5.4 shows results obtained using the model for the bistatic radar cross section of a dielectric sphere of permittivity 2.105, normalized with respect to the radius a of the sphere. Results are compared with the exact solution obtained using Mie theory. For an incident plane wave polarized in the x direction, the E-plane and H-plane radar cross sections are given as $\sigma_\theta(\theta, 0)$ and $\sigma_\phi(\theta, \pi/2)$, respectively, where the p component of the radar cross section $\sigma_p(\theta, \phi)$ is defined as

$$\sigma_p(\theta, \phi) = \lim_{R \rightarrow \infty} 4\pi R^2 \frac{|E^s(\theta, \phi)|^2}{|E^{inc}|^2} \quad (5.14)$$

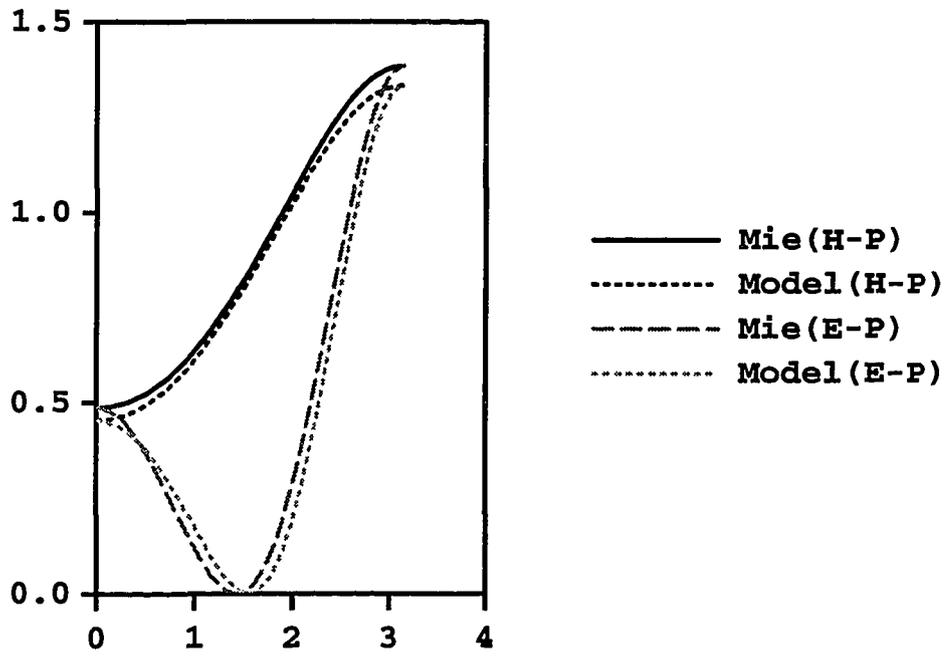


Figure 5.4: Comparison of the numerical and exact solutions of the bistatic radar cross section of a sphere ($ka=0.9$ and refractive index $N=2.105$)

The results in Figure 5.4 are shown for $\frac{\sigma_\theta(\theta,0)}{\pi a^2}$ and $\frac{\sigma_\phi(\theta,\pi/2)}{\pi a^2}$.

The study of electromagnetic scattering from a sphere has important applications in NDE of integrated circuits chips. In this case, the sphere simulates a dust particle and the problem is cast in the form of scattering from a sphere on top of an infinite half space. Detailed discussion of the problem is found in [62]. A sphere is also used to simulate a general type of 3D void or inclusion embedded in a civil structure such as a roadway or a bridge deck.

The Green's function used in the case of a 3D object embedded or on top of a multilayered media has to include a secondary term to account for the interaction with layered media. A special case of this problem is an object on top or above a perfectly conducting surface and is discussed next, followed by a discussion of the general case where the secondary Green's function has to be formed.

A scatterer above a perfect conducting surface

The scattered field of an object on top of a perfectly conducting surface (Figure 5.5) is obtained by including the effect of the surface in the integral equation and replacing the surface by a space of permittivity ϵ_1 . The governing integral equation for this problem is written as

$$\underline{E}^s = -j\omega\mu \left\{ \int_{v'} G((x-x'), (y-y'), (z-z')) \cdot \underline{J}_{eq}(\underline{r}') dv' + \int_{v'_{image}} G((x-x'), (y-y'), (z-z')) \cdot \underline{J}_{eq-image}(\underline{r}') dv' \right\} \quad (5.15)$$

where $J_{eq-image}$ is the equivalent current on the elements of the image of the object.

The unknowns $J_{eq-image}$ are actually not independent of the unknowns J_{eq} . From

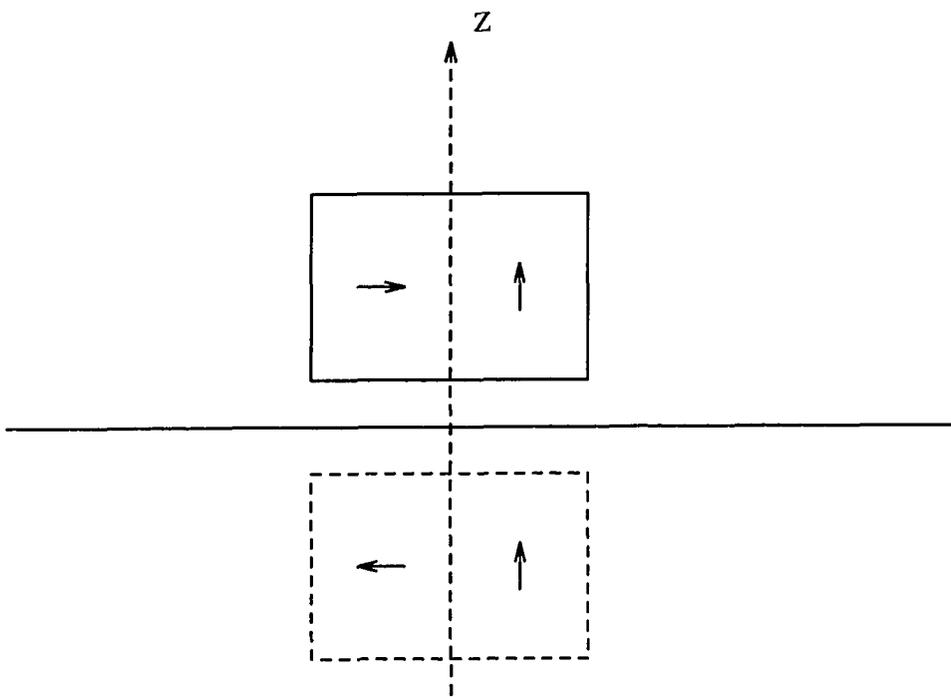


Figure 5.5: An object on top of a perfectly conducting surface, showing images of a vertical and a horizontal current components

image theory, we have

$$\begin{aligned}\text{Image}\{J_x\} &= J_x \\ \text{Image}\{J_y\} &= J_y \\ \text{Image}\{J_z\} &= -J_z\end{aligned}\tag{5.16}$$

The number of equations required to be formed is thus $3N$ for N elements which is achieved by choosing \underline{r} to reside at the center of each element for each degree of freedom. Equation (5.15) is written as

$$\begin{aligned}\underline{E}^s = -j\omega\mu \left\{ \int_{v'} G((x-x'), (y-y'), (z-z')) \cdot \underline{J}_{eq}(\underline{r}') dv' + \right. \\ \left. \int_{v'} G((x-x'), (y-y'), (z+z')) \cdot \underline{J}_{eq-image}(\underline{r}') C dv' \right\}\end{aligned}\tag{5.17}$$

where the second integration has been replaced by an integration on the scatterer itself with $(z+z')$ used in the Green's function. C is a diagonal matrix of elements $[1 \ 1 \ -1]$ which comes from equation (5.16).

The second integral term in equation (5.15) can be changed to obtain a convolution form. Equation (5.15) can be written as

$$\begin{aligned}\underline{E}^s = -j\omega\mu \left\{ \int_{v'} G((x-x'), (y-y'), (z-z')) \cdot \underline{J}_{eq}(\underline{r}') dv' + \right. \\ \left. \int_{v'_{image}} G((x-x'), (y-y'), (z-z')) \cdot \underline{J}_{eq}(x', y', -z') C dv' \right\}\end{aligned}\tag{5.18}$$

In the implementation, six vectors are used to store the Green's tensor for sources in the image and observation points in the object. Equation (5.18) is in convolution form while the negative sign of the z' in the current is reflected by a reordering of its FFT. Figure 5.6 is a one dimensional illustration for forming current and Green's function vectors for the image elements and Figure 5.7 is the repeated form of these values illustrating that there will be no aliasing in evaluating the convolution in

equation (5.18). Figure 5.8 shows the calculated bistatic radar cross section for a sphere on top of a perfectly conducting plane.

In the above formulation, image theory is used to avoid the need for performing integrations for the secondary Green's functions. If the object is on top or embedded in dielectric materials, however, the secondary Green's functions have to be obtained via integration.

A scatterer embedded in multilayered media

In this case, the Green's function due to the materials enclosing the object or *the secondary Green's function* has to be found. Formulation of the secondary Green's function is discussed next.

Secondary Green's function

Assuming we have a dipole over multilayered media, we can obtain the secondary field by decomposing the primary spherical field using Sommerfeld's identity

$$\frac{e^{-jk_r}}{r} = \frac{j}{2} \int_{-\infty}^{\infty} \frac{k_\rho}{k_z} e^{jk_z|z|} H_0^{(2)}(k_\rho \rho) dk_\rho \quad (5.19)$$

where k_z and k_ρ are the wave numbers in Z and ρ directions, and $\rho = \sqrt{x^2 + y^2}$.

Each component of the wave decomposition will have a different k_z and a corresponding reflection coefficient from the multilayered media. Thus, the secondary Green's function will be an integration of all reflected waves corresponding to different k_z or k_ρ ; these integrals are known as Sommerfeld integrals. Denoting k_ρ by λ , we have an integration for all values of λ as

$$G_s = \int_{-\infty}^{\infty} f(\lambda) H_n(b\lambda) e^{zh(\lambda)} d\lambda \quad (5.20)$$

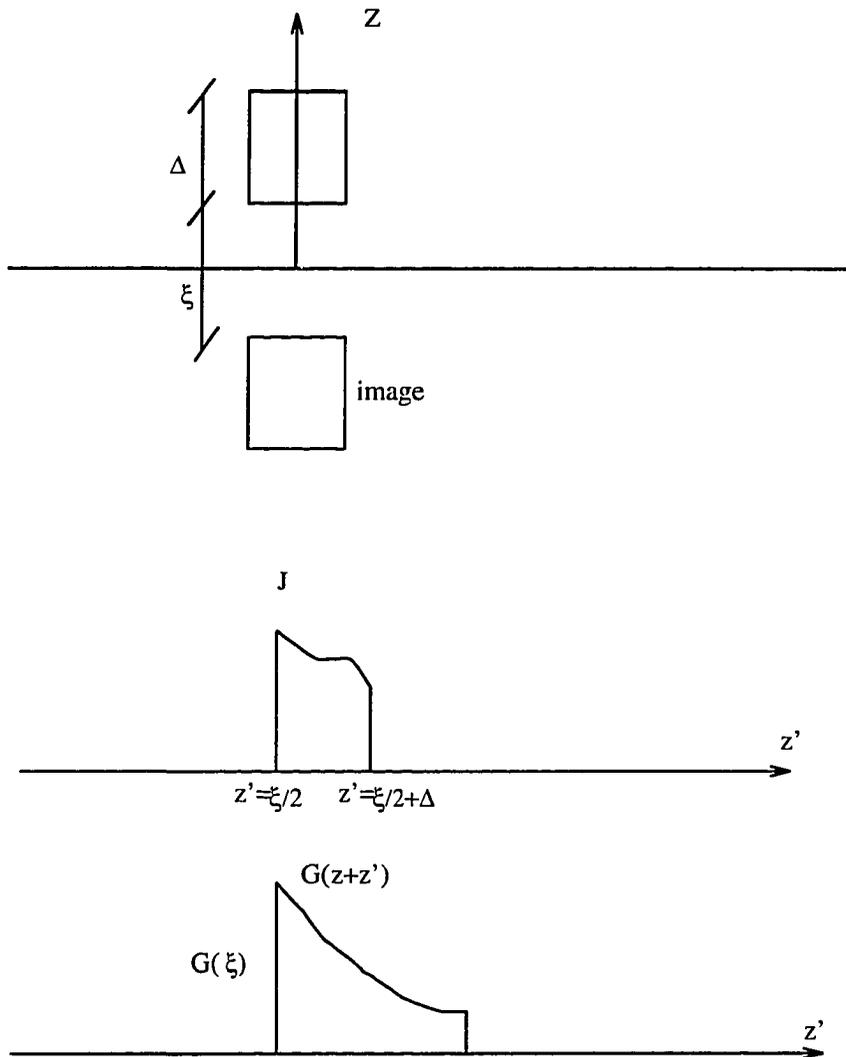


Figure 5.6: One dimensional illustration of the object equivalent current and Green's function of the image elements

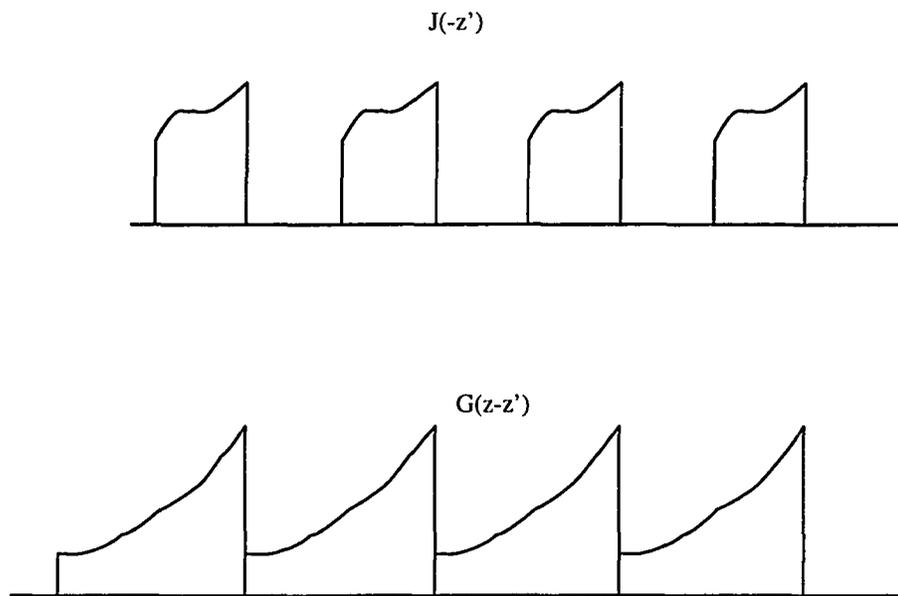


Figure 5.7: Repeated form of one dimensional equivalent current and Green's function of the image elements

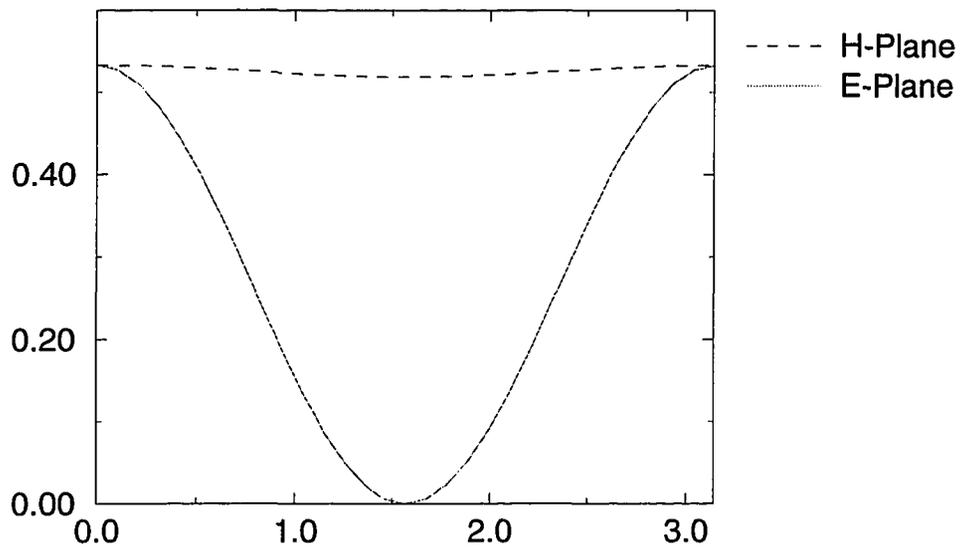


Figure 5.8: Bistatic radar cross section of a sphere ($ka=0.9, N=2.105$) on a 1λ distance from a perfectly conducting surface.

Evaluating integrals of the form of equation (5.20) is very crucial in solving problems of multilayered media. The difficulty of evaluating these integrals is related to the numerical instability due to the large oscillations of the integrand. Several methods have been used to evaluate these integrals.

Stationary phase method: The idea of this technique is to use one plane wave component with the largest contribution to the integral. This has been discussed in Chapter 4.

Saddle point technique: This method depends on deforming the contour of the integration to the steepest descent path where the imaginary part of the argument of the exponential function is constant. If the function $h(\lambda)$ in equation (5.20) is written as

$$h(\lambda) = \phi(\lambda) + j\psi(\lambda)$$

and from Cauchy-Riemann equations, $\nabla\phi \cdot \nabla\psi = 0$, thus, the lines of constant ϕ and constant ψ are orthogonal. In Figure 5.9 the path M-S-N, that passes through the saddle point S, is the steepest descent path where ψ is constant and $\phi(x, y) < \phi(x_S, y_S)$. For large distances between the source and measuring point, most of the contribution of the secondary Green's function integral comes from around the saddle point, where ϕ is maximum [58].

Digital filter technique: In this method, equation (5.20) is folded into a range from 0 to ∞ . The integral can be written in the form

$$K(b) = \int_0^\infty k(\lambda)H_n(b\lambda)d\lambda \quad b > 0 \quad (5.21)$$

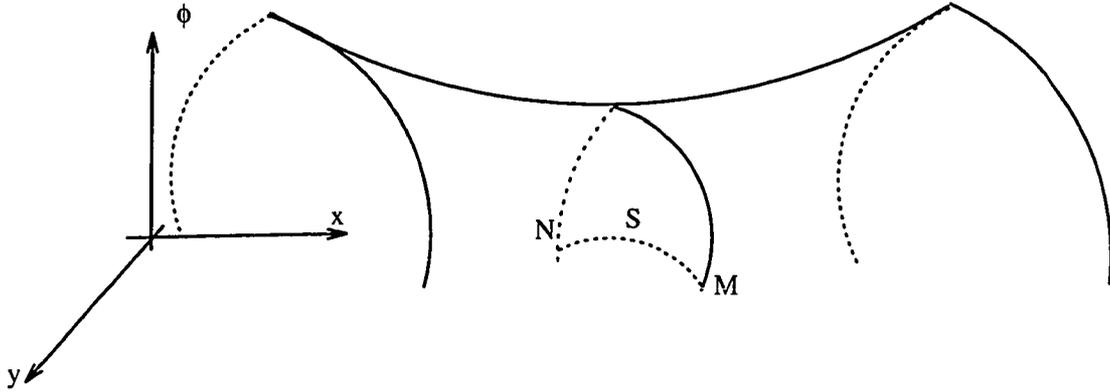


Figure 5.9: Saddle point

Using the transformations $x = \ln b$, $y = -\ln(\lambda)$ and multiplying both sides by e^x , we get

$$e^x K(e^x) = \int_{-\infty}^{\infty} k(e^{-y}) \left[e^{x-y} H_n(e^{x-y}) \right] dy \quad (5.22)$$

Equation (5.22) is in the form of a convolution integral. A digital filter is designed [63] to perform this integration where the input function is $k(e^{-y})$, and the output is $e^x K(e^x)$. This is done by obtaining known input output pairs from an analytical equation, and deriving the filter spectrum from the ratio of the DFT of the output to the DFT of the input signals.

Real axis integration: This method overcomes the problem of fast oscillation of the integrand of equation (5.20) without intensive investigation of the locations of poles and branch cuts required for the saddle point technique. The method depends on mapping the integrand with different maps corresponding to different locations on

the real axis. An asymptotic form of the integrand is obtained by approximating the exponential function in the integrand by its Taylor expansion. This term is subtracted from the integrand to stabilize the integration, and its analytical integration is added to the result [64].

The integration is performed using an adaptive mesh size. A step interval H is chosen and then spanned by a sequence of n finer substeps $h = H/n$. The final answer of the integration over the H interval is considered to be an analytic function of $F(h)$ and the correct solution is the value of $F(0)$, which is obtained by extrapolating the results corresponding to different substep sizes h_1, h_2, \dots . This technique is known as Richardson's extrapolation. The extrapolating function can be chosen to be a polynomial or rational function [65].

The number of substeps n starts with a value of two, and is then increased. At each step an extrapolation is attempted to give the extrapolation value and an estimate for the error, and the process continues as long as the error is above acceptable limits. If the number of substeps exceeds a certain value without achieving the required accuracy, the step size H is reduced and then subdivided into new substeps. For infinite integration intervals, truncation is required to be performed as described in [64].

The techniques discussed in Chapter 4 and Chapter 5 can efficiently reduce the computational resources required for a given problem and, hence, enable the use of the available computers. However, many real world problems are complex with three dimensional geometries and cannot be solved on sequential computers within a practical time frame. The need for supercomputing power is inevitable in these cases. Chapter 6 gives an overview of parallel computers, followed by a discussion of

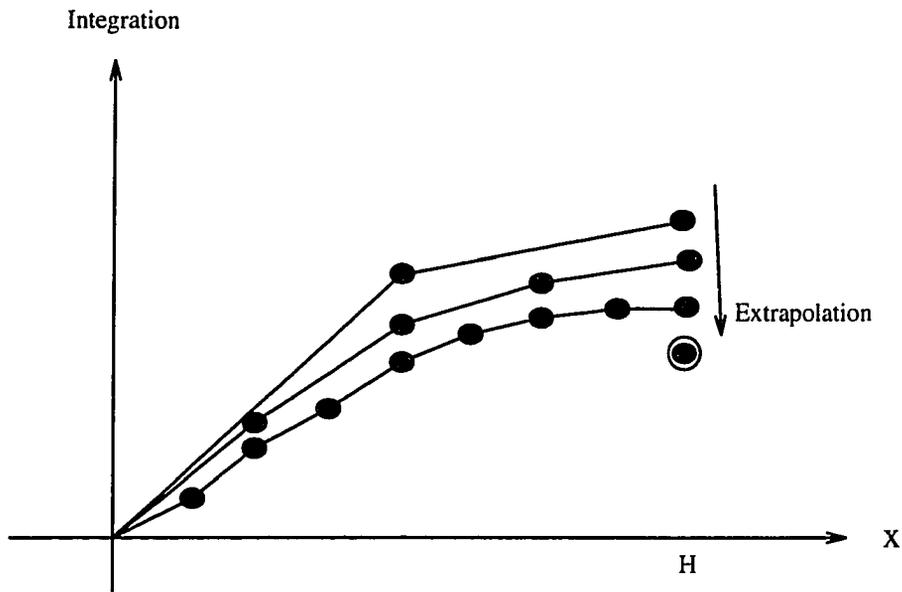


Figure 5.10: Error reduction in integration

a parallel computing implementation of electromagnetic NDE application in Chapter 7.

CHAPTER 6. PARALLEL COMPUTING FOR LARGE SIZE PROBLEMS

Since the 1940s, computing machines and algorithms have been centered around the basic sequential architecture taught by John von Neumann at Moore School. In this architecture, a single control unit, a single processor and single memory are used (Figure 6.1). Instructions and operands are fetched from the memory, and results are stored back in memory. The von Neumann machine contained one of everything, which made it conceptually simple to design and program, resulting in the popularity of such machines until recent times.

Another reason for the popularity of sequential architecture was given by Herb Grosch in the 1960s, and is known as *Grosch's Law* [66]. This law gives the proportionality between the performance of a computer and the square of its cost. It predicts a reluctance of the buyers to go to new machines at twice the cost as available machines, unless the performance of the new machine is fourfold. However, the simple view of the sellers was that, with twice as much hardware, there will only be an upgrade to twice the performance. Thus, doubling the hardware—and the cost—of new computers was not enough, according to Grosch's law, for the buyers to move to the new computers. Thus, Grosch's law kept the trend for computer companies to mostly design and build sequential computers.

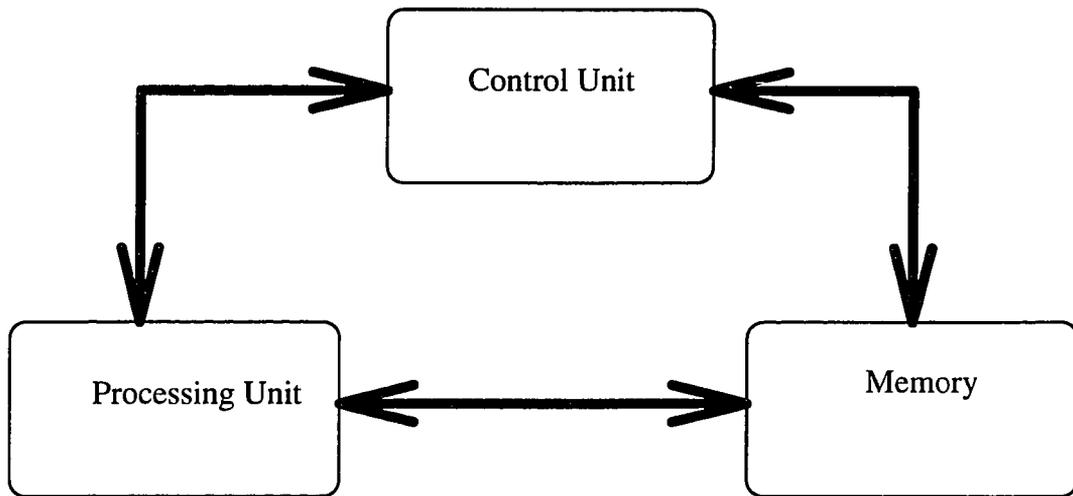


Figure 6.1: Von Neumann machine

However, with the switch of electronics components to solid state, Grosch's law started to fail. The components were getting smaller, cheaper and, at the same time, faster. In 1976, Seymour Cray outsmarted Grosch's law with the development of the first "vector" computer, the Cray-1. In a vector computer, successive operations on data vectors can be overlapped to increase the overall throughput. The machine was capable of 130 Mega Floating point operations per second (MFLOPS). In 1988, the Cray Y-MP was announced with eight processors and the peak performance of the 2.67 GFLPS.

Skepticism around parallel computers that also hindered the move to invest in buying parallel computers and implementing parallel algorithms were also strengthened by the performance analysis known as Amdahl's law. In this analysis [67], a problem is assumed to have a serial part of percentage ' β ', and parallel part ' $1 - \beta$ '. If the total execution time of the problem on a sequential computer is $C(1)$, then the time for executing the sequential part on a sequential machines is given as $[C(1) (1 - \beta)]$, and the time for the parallel part time is given as $[C(1)\beta]$.

The total time on a parallel computer with P processors $[C(P)]$ is given as

$$C(P) = C(1)(1 - \beta) + \frac{C(1)\beta}{P} \quad (6.1)$$

and the speedup $S = \frac{C(1)}{C(P)}$ is given by

$$S = \frac{1}{(1 - \beta) + \frac{\beta}{P}} \quad (6.2)$$

which is known as Amdahl's law.

According to Amdahl's law, if the number of processors P goes to infinity, the speedup will saturate at a value $\frac{1}{(1-\beta)}$ (Figure 6.2) . Thus, using a parallel computer

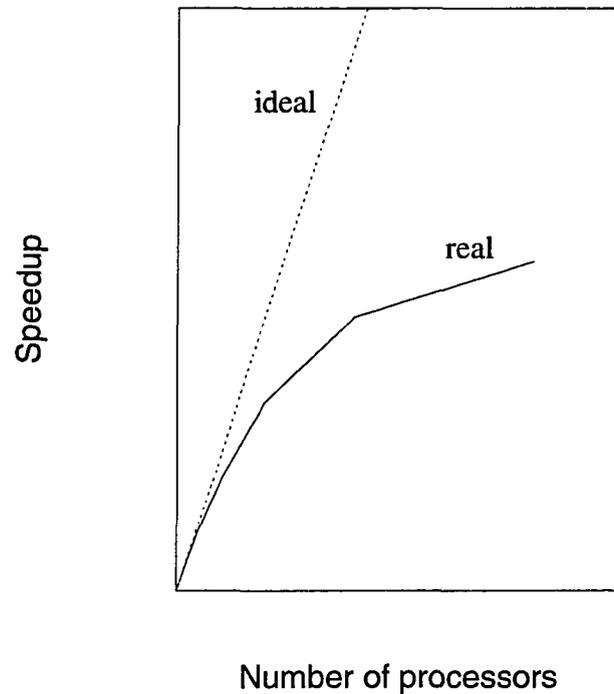


Figure 6.2: Typical speedup curve

with a large P can be a waste of resources since the performance reaches a speedup saturation level that can be a small value, irrespective of the number of processors.

In 1988, a team of researchers at Sandia Labs won a \$1000 cash prize for showing results which invalidate Amdahl's law. Amdahl's law assumes that the $(1 - \beta)$ parallel component of the problem is implicitly independent of the number of processors. This underlying assumption was questioned by Sandia's research group [68]. The results of their work, referred to as the *Gustafson-Barsis Law*, is based on the use of *scaled* speedup, which means that the problem size N increases with P . From Figure 6.3,

the hypothetical serial processor time complexity is given as

$$C(1) = \beta + (1 - \beta)N \quad (6.3)$$

and parallel processor time

$$C(P) = \beta + (1 - \beta) = 1 \quad (6.4)$$

The speedup $S(N,P)$ is given as

$$S(N, P) = \beta(1 - N) + N \quad (6.5)$$

Figures 6.4 and 6.5 show the difference of effect of the serial part β on speedup, considering the fixed size and scalable problem. The results of Gustafson-Barsis law motivated the interest in parallelism and parallel computing [66].

Parallel computers architecture

Computer architecture is generally divided, according to Flynn's taxonomy, into four major groups according to both data and instruction streams:

	Single Instruction	Multiple Instruction
Single Data	SISD (von Neumann)	MISD
Multiple Data	SIMD	MIMD

The SISD machines are the von Neumann machines, which include the commonly used sequential computers, although most machines now apply some sort of hidden parallelism as in pre-fetching [15]. There are now machines that fall into the category

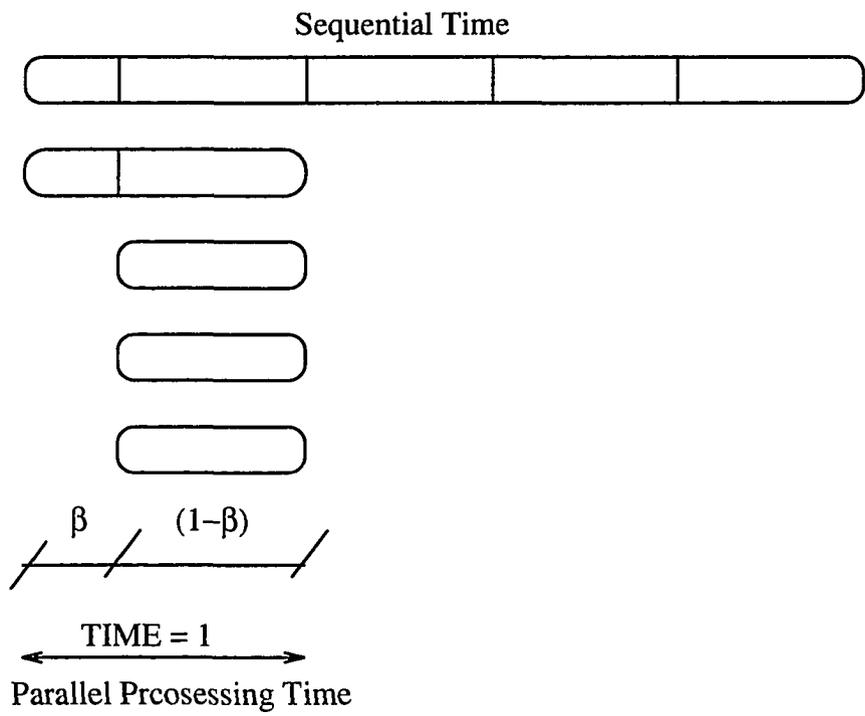


Figure 6.3: Parallelizing the β part of a canonical scalable problem

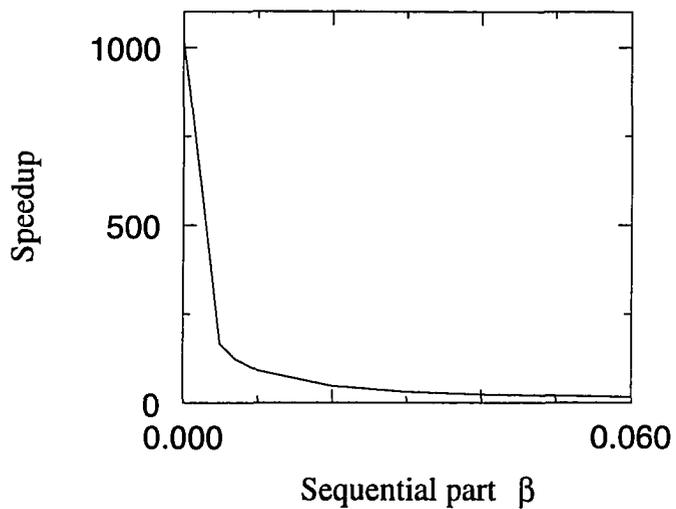


Figure 6.4: Effect of the sequential part β on the speedup for fixed size problem

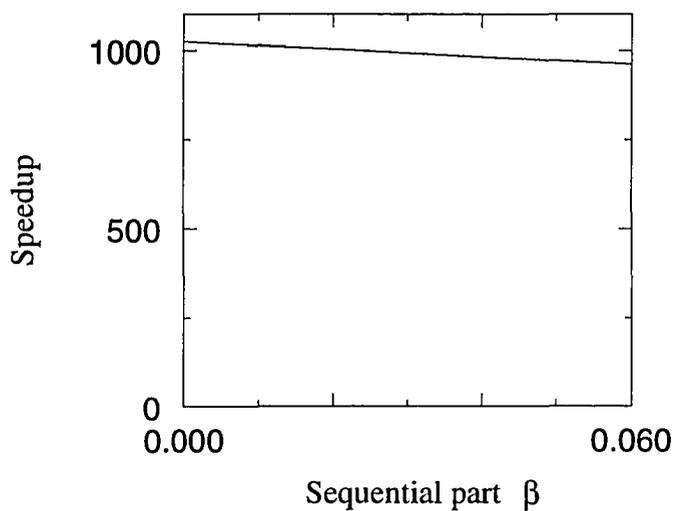


Figure 6.5: Effect of the sequential part β on the speedup for scalable problem

of MISD type. The available parallel computers fall into the two categories of SIMD and MIMD. SIMD machines are very useful for applications that require the same set of operations to be applied to large arrays of data as in the case of image processing. MIMD computers, on the other hand, are more flexible in programming than SIMD machines and can be used in a wider range of applications. The MIMD machines are further classified on the basis of the scheme used for interconnecting the processors. For a small number of processors, *shared-memory* can be used by connecting the different processors to a single memory store. This architecture, however, faces the problem of lack of scalability. The number of processors cannot be increased to a large value because of the bottleneck associated with memory access. An alternative MIMD architecture is to have distributed memory, where each processor has its own memory. Such an architecture can overcome the problem of memory access, but the performance depends on the interconnection scheme used to link the processors. For example, if all of the the processors are connected to a single bus, the same sort of bottleneck will be encountered. The solution is to have each processor connected to a few neighboring processors.

One of the most popular interconnection topologies that addresses the problem of scalability and memory access is the the hypercube architecture described next.

Hypercube architecture

The classical hypercube architecture has received widespread attention in recent years. The processors (*or nodes*) have only local memories that make the architecture scalable to accommodate larger numbers of processors relative to globally shared memory systems [69]. The architecture is accommodated in systems that have mul-

tuple processor, distributed-memory and message-passing features.

For the hypercube architecture with a processor at each corner or vertex of an n -dimensional cube, each processor (node) is connected to other processors (nodes) that only differ in one bit of the binary representation of its number. For example, a hypercube of dimension 3 consists of 8 processors. The interconnections are such that node 0 (000) is connected to nodes: 1 (001), 2 (010), and 4 (100) (Figure 6.6) . The total number of nodes of a hypercube of dimension D is 2^D , and the largest communication distance is D links. The exponential increase in the number of processors with linear increase in communication distance makes the the hypercube highly scalable.

One of the first companies to manufacture parallel computers with hypercube architecture was NCUBE. The company launched the first generation, NCUBE/ten in 1985 and the second generation, NCUBE-2 in 1989. NCUBE/10 had a maximum dimension of 10 (1024 nodes) and NCUBE-2 has a maximum dimension of 13. A fully configured version of NCUBE-2 has a theoretical peak of 27 GFLOPS.

The EM NDE parallel computing research described in the next chapter is conducted on two NCUBE machines. The first machine is at the Scalable Computing Laboratory (SCL) at Ames Lab, Ames, Iowa. This machine has 128 processors, and each processor has a peak speed of 2.8 MFLOPS.

The second machine is at the Sandia National Laboratory (SNL) Massively Parallel Research Laboratory (MPCRL). The NCUBE at SNL has the following properties:

- 1024 nodes (backplane engineered for 8192 nodes)
- custom VLSI CISC processor

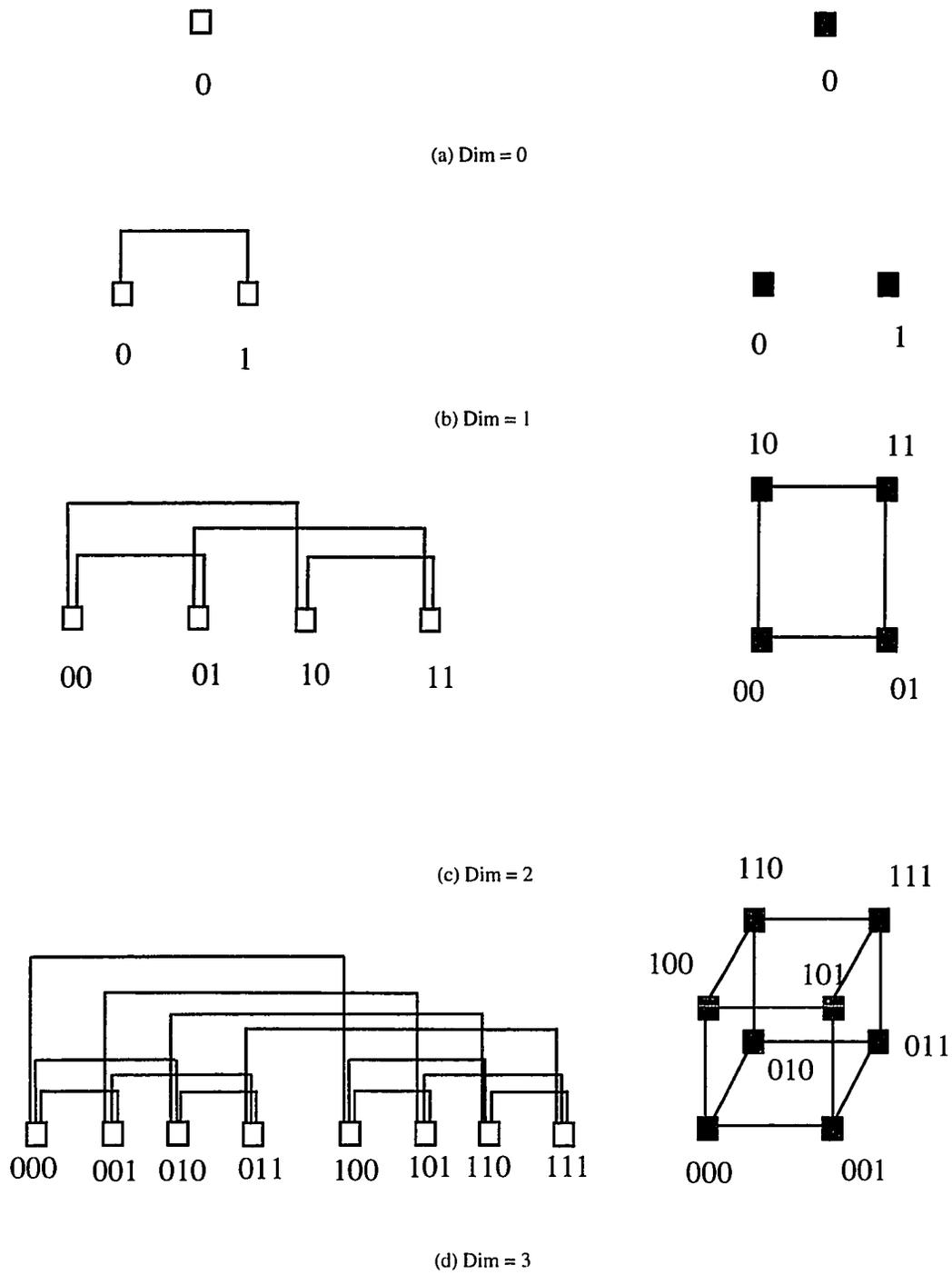


Figure 6.6: Hypercube architecture

- 64 bit registers
- rich instruction set and addressing modes
- 2.7 MFLOPS / node (1.5 - 2 actual)
- 7 MIPS / node (4 - 7 actual)
- 20 MHz clock
- 4 MB / node memory

Communication in NCUBE2

In distributed memory architectures, each processor commonly maintains its own memory, even though a shared memory configuration can be emulated by making the interprocessor communications hidden from the user. In NCUBE2, the user program specifically requests a communication of data between two (or more) nodes. This is handled by function calls. For example, a function *nread* handles the reception of data on nodes. The function *nread* reads a message from an NCUBE2 user process message buffer to a specified address. When a message arrives at its destination processor, the system stores it in the message buffer space of the process for which it is destined. The *nread* call copies the message data from the user process message buffer space to a specified address. The arguments to the *nread* call specify the length (number of bytes) and type of message to be read, the source of the message, and the location at which the message data is to be stored.

The function is described as integer function *nread* (*buffer*, *nbytes*, *source*, *type*), where:

buffer: the location at which the message data is to be stored.

nbytes: the length in bytes of the message to be read.

source: the source processor and process ID from which the message is to be read.

type: the message type used to specify the kind of message to be read.

Another function *nwrite* is used in writing data from one source to a destination. The *nbcast* function is used to broadcast data from one processor to a masked hypercube of the architecture.

With the communication system of NCUBE2, the user has full control of location and motion of the data within the system memory. The system is easy to handle, although special care has to be given to the programming logic to avoid deadlocks that arise in cases such as two processors waiting for a write from each other. The user program should optimize the load distribution on the processors to reduce the interprocessor communication and achieve a load balance.

The next chapter describes an implementation of an NDE electromagnetic application on the NCUBE2. The application problem involving Monte Carlo simulation cannot be solved within practical time limitations on a sequential computer, and resorting to parallel computing for such applications is inevitable.

**CHAPTER 7. IMPLEMENTATION OF 3D FINITE ELEMENT
MODEL ON PARALLEL COMPUTERS FOR EDDY CURRENT NDE
POD CALCULATIONS**

Eddy current NDE techniques depends on analyzing the measured values of the impedance of a coil that is brought close to the specimen under test. Finite Element Method is a natural candidate for eddy current problems since the domain of the problem is finite, due to the localized nature of the fields around the transducer.

The quantitative evaluation of an NDE modality, such as the eddy current method, requires the development of probability of detection 'POD' models for that modality. POD models are obtained using a Monte Carlo modeling of different sources of random variability that are encountered during actual testing. This demands running the numerical model several thousand times. As the time requirements for numerically solving 3D problems is usually high, POD calculations for three dimensional geometries are not practical on sequential computers. A solution to this problem, using parallel computers, is described next.

The computational bottleneck for finite element models is the solution of a linear system of equations. Iterative methods for the solution have the advantage of low storage requirements, in contrast to direct methods that usually depend on time consuming off-core methods of solution. However, iterative methods may also require

a large number of iterations for certain types of data. Parallel computer implementations using both iterative and direct methods for finite element modeling of eddy current NDE are presented.

Finite element modeling of eddy current NDE of 3D objects

In FE modeling of the eddy current test, the region of interest that includes the test object and the probe is discretized into small subregions or *finite elements* as described in Chapter 2. The elements are chosen here to be hexahedral, where each element has 8 nodes. The formulation of the problem is obtained by incorporating the governing system of differential equations into a scalar function generally known as the *energy functional*. The energy functional F is written similar to the 2D formulation as

$$F = \int_v (\text{stored energy} - \text{input energy} + \text{dissipated energy}) dv \quad (7.1)$$

$$F = \int_v \left\{ \frac{1}{2} \left[\frac{1}{\mu_x} \left(\frac{\partial A_z}{\partial y} - \frac{\partial A_y}{\partial z} \right)^2 + \frac{1}{\mu_y} \left(\frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x} \right)^2 + \frac{1}{\mu_z} \left(\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y} \right)^2 \right] \right. \\ \left. - (J_x A_x + J_y A_y + J_z A_z) + \frac{1}{2} j \omega \sigma (A_x^2 + A_y^2 + A_z^2) \right\} dx dy dz \quad (7.2)$$

The second step is to represent the unknown vector \underline{A} in (7.2) in terms of its nodal values and a shape function N for the element, as

$$A_x(x, y, z) = \sum_{i=1}^8 N_i(\xi, \eta, \zeta) A_{xi} \quad (7.3)$$

$$A_y(x, y, z) = \sum_{i=1}^8 N_i(\xi, \eta, \zeta) A_{yi} \quad (7.4)$$

$$A_z(x, y, z) = \sum_{i=1}^8 N_i(\xi, \eta, \zeta) A_{zi} \quad (7.5)$$

The third step is to differentiate the functional with respect to the nodal values for each element. This produces a local matrix for each element.

$$\partial F(A)/\partial k_i = 0 \quad (7.6)$$

where:

$$i=1,2,3\dots 8$$

$$k=x,y,z$$

A local system of equation is formed from (7.6) as

$$[S^e][A^e] = [Q^e] \quad (7.7)$$

where the RHS is obtained from source elements and boundary conditions and the local matrices $[S^e]$ are symmetric of dimension 24x24.

In the 4th Step, the local matrices are assembled into one global matrix. The global system of equations is solved for the nodal values of the magnetic vector potential \underline{A} .

The coil impedance can be easily found once the nodal values of \underline{A} are known. The inductance is related to the stored energy W , and the input current I_s . We have

$$L = 2W/I_s^2 \quad (7.8)$$

$$W(\text{Stored Energy}) = \frac{1}{2} \int_{\text{volume}} \frac{1}{\mu} |\nabla \times A|^2 dv \quad (7.9)$$

The resistive part of the impedance is given in terms of the dissipated energy P .

$$R = P/I_s^2 \quad (7.10)$$

$$P(\text{Dissipated Energy}) = \int_v \sigma \omega^2 |A|^2 dv \quad (7.11)$$

In general, the solution of the matrix equation is extremely computation intensive. As an example, a typical 3D finite element mesh for the geometry of an eddy current test consists of 2475 elements. The corresponding number of nodes is 3072, each of which has three degrees of freedom. The bandwidth of one half of the matrix is (210×3) . The memory required to store the global matrix is given as $[(3072 \times 3) \times (210 \times 3) \times 8 = 46.5 \text{ Mega Bytes (MB)}]$, if the matrix elements are stored as single precision complex numbers.

As the size of the global matrix increases, solution of the linear system of equations becomes increasingly difficult, requiring an off-line solution scheme. This makes the execution slow, with time requirements typically in the range of a few hours. The time requirements for executing POD models involving several executions of the measurement (finite element) model is, therefore, exorbitant making the implementation impractical on sequential computers. We first introduce the concept of POD models, followed by solution to the execution time problem via the use of parallel computers; in particular, the hypercube architecture.

Probability of detection (POD) models

Probability of detection (POD) models constitute a powerful tool for assessing the applicability and quantifying the reliability of a selected nondestructive evaluation (NDE) technique. In addition POD models are useful for determining the optimum test parameters for a given experimental configuration.

The probability of detection model centers around a measurement model that computes the measurement variable. Factors influencing the measurement process are modeled by a multivariate Gaussian distribution. Samples from the multivariate

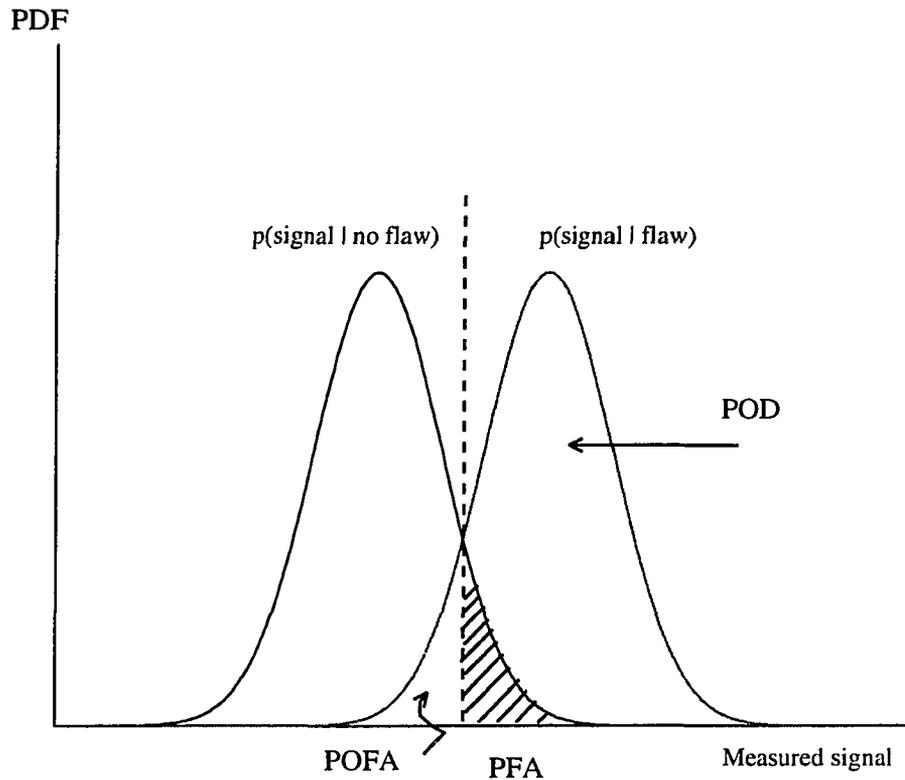


Figure 7.1: Probability density function for a signal in the presence of a flaw and without a flaw

distribution are used to perturb the measurement model, in a Monte Carlo simulation scheme, to predict the conditional probability density function (pdf) of the signal generated by a flaw. The pdf is then used for computing the probability of detection, the probability of false alarm, and the probability of false acceptance (Figure 7.1).

POD models for eddy current NDE

In eddy current techniques, the impedance of the coil probe carries information about the state of the test specimen. The POD model accounts for different random

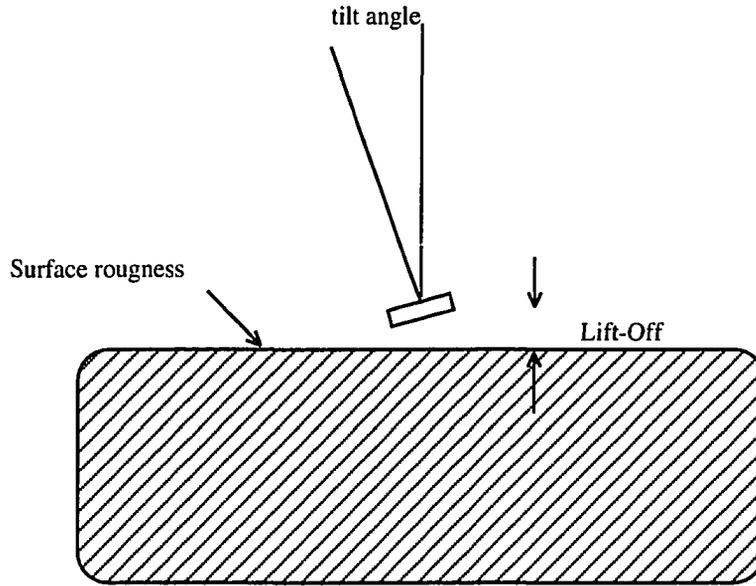


Figure 7.2: Sources of random variation in eddy current testing

sources of noise encountered in the process of testing. These random sources include variations in the coil lift-off and tilt (Figure 7.2), the material conductivity, temperature changes and surface roughness, in addition to those introduced due to human factors, and instrumentation noise.

These variations can be modeled as random perturbations to a deterministic finite element model representing the test setup.

It can be shown [70] that for estimating the probability P_k with a desired accuracy ϵ , the minimum number of runs N can be obtained from the following inequality:

$$P(|\hat{P}_k - P_k| > \epsilon) \leq \frac{1}{4N\epsilon^2} \quad (7.12)$$

where \hat{P}_k represents the estimate of P_k .

For $\epsilon = 5\%$, we require N to be greater than 2500. Thus, the computation of a single probability value can take up to a few years of computation time. An alternate strategy, that involves implementing the model on a parallel computing architecture to obtain the estimates of the probability values within a more reasonable time frame, is described next.

Implementation of finite element eddy current model on a hypercube

Examples involving implementation of finite element algorithms on a hypercube are found in [71, 72, 73]. Three different implementations of a FE model on an NCUBE2 that employs a hypercube architecture are presented.

The first scheme uses an iterative approach for solving the linear system of equations. The advantage of this approach is the low memory requirement, making it suitable for implementation on most existing computers. Implementation of the FE model, using an iterative matrix solution, was performed on the DEC stations and the CRAY Y-MP machines to avoid the need for using wavefront solution schemes.

The governing equation for eddy current phenomenon is given in terms of the unknown vector magnetic potential \underline{A} . In order to obtain the desired level of accuracy in calculating the coil impedance from \underline{A} , a large number of iterations are required. However, due to the large number of processors and memory resources available on the NCUBE, direct methods of implementation are possible, thereby avoiding the need for using iterative procedures. This results in a significant speedup, both as a consequence of parallel computing environment, as well as due to the elimination of iterative disk access which is prevalent in off-core solution methods.

The implementation of the iterative incomplete Cholesky conjugate gradient

method and direct Gaussian elimination methods on the NCUBE environment are described in the following sections.

Incomplete Cholesky Conjugate Gradient Method (ICCG)

A description of the theoretical principles of the ICCG method is found in [48, 74]. The advantage of using this iterative technique is to reduce the storage requirement. In the CG method, the global matrix ‘ S ’ is used for finding the product vectors $S^a \underline{R}^n$ and $S \underline{g}^n$, where S^a is the adjoint of S , n is the iteration number, \underline{R} is the residual vector and \underline{g} is the variational vector. In implementing the method, only the local elements are stored and used to find the product vectors. For example, the product $S \underline{g}^n$ is found as

$$\sum_{\text{elements}} S_i \underline{g}_i^n \quad (7.13)$$

where S_i is the local element matrix for element i , and \underline{g}_i is a subvector of \underline{g} corresponding to nodes of the i th element. The product $A^a R^n$ is similarly obtained.

For the 2475-element problem, the global matrix is shown to require 46.5 MB of storage. On the other hand, the total storage for local matrices in the ICCG algorithm can be shown to be

$$2475 \times 300 \times 8 \text{ bytes} = 5.94 \text{ MB}$$

where 300 represents the number of elements required to store 24x24 symmetric local matrices.

NCUBE implementation Computing on the NCUBE requires the computation as well as storage loads to be evenly distributed among the processors. In the CG

algorithm, where the bulk of the computation is the ($S^a \underline{R}^n$ and $S \underline{g}^n$) matrix-vector multiplication, the implementation makes each processor responsible for finding a subvector of the product vector. The subvectors are chosen to be of almost equal size. A code was written to find the elements contributing to each of the output subvectors. Each processor forms and stores only the elements it needs, and sums their contribution in order to find a product subvector.

Once the output subvectors are found, each node gets the rest of the product vector by communicating with the other nodes. The best algorithm in terms of communication time is to have a single node accumulator to collect the subvectors at one node and then a *single node broadcast* of the total vector to the other nodes.

The computation of the 3 norms in the conjugate gradient algorithm is done by dividing the task among the processors. Each processor finds the norm of a subvector and the results are communicated to other nodes.

Time complexity and speedup The time complexity 'TC' of ICCG algorithm implemented on a sequential computer can be written as

$$TC = 2 \times 2 \times (24)^2 \times NElements + 3 \times 2 \times NNodes + C \quad (7.14)$$

where NElements is the number of elements, NNodes is the number of nodes, and C represents latency time.

For the parallel algorithm, the time complexity TC is given as

$$TC = 2 \times 2 \times (24)^2 \times O\left(\frac{NElements}{P}\right) + 3 \times 2 \times O\left(\frac{NNodes}{P}\right) + 2 \times t_c \times \log P + C_2 \quad (7.15)$$

where t_c is communication time factor, P represents the number of processors and C_2 represents latency time.

As most of the run time is used for solving the system of equations, we time only one iteration of the ICCG. For the 2475-element example, the DEC 5000 station takes 29 seconds per iteration, while the CRAY Y-MP takes around 1 second. On the NCUBE, the time for each iteration is given as

Number of processors	1	32	64
Time for one iteration (sec)	46.6	3.9	2.5

The relative speedup for the case of 32 and 64 processors is given by

$$\text{For } P=32: \text{ Speedup} = \frac{46.6}{3.9} = 11.95$$

$$\text{For } P=64: \text{ Speedup} = \frac{46.6}{2.5} = 18.64$$

The absolute speedup, compared with DEC 5000 stations, is

$$\text{For } P=32: \text{ Speedup} = \frac{29}{3.9} = 7.44$$

$$\text{For } P=64: \text{ Speedup} = \frac{29}{2.5} = 11.6$$

Gaussian elimination

The Gaussian elimination method transforms the global matrix into a triangular matrix. The implementation on the NCUBE is done by the scatter decomposition of the matrix to allow load balancing. The matrix is stored in rows, where only the upper half band of the matrix is stored due to symmetry. Processor n stores row numbers $n+i*P$, where P is the number of processors, and $i=0,1,\dots$. In forming the global matrix, each processor checks for the mesh elements contributing to its rows, constructing only the local matrices corresponding to these elements.

There are two phases in the method. The first phase is the elimination phase followed by the second phase, the backsubstitution phase. The operations in the elimination phase are as follows:

For $i=1$ to $N-1$

(where N is the dimension of the unknown vector)

The processor that has the i th row broadcasts the elements of this row and the i th element of the RHS vector to all other processors.

Each processor works to change the rows that it stores of the global matrix, and the RHS vector, such that for rows $i+1$ to N

$$\text{jth row} = \text{jth row} - \frac{S_{ij}}{S_{ii}} * \text{ith row}$$

End

In the backsubstitution phase, we use a loop to find the elements of the unknown vector \underline{X} , from X_N to X_1 .

For $i=N$ to 1,

The processor that has the i th row finds the value of X_i and broadcasts it to other processors.

End

Time complexity and speedup In the sequential Gaussian elimination (GE) algorithm, for each row we need

$$(1 + 2 + \dots + BW) = \frac{BW}{2} BW \text{ multiplications and additions}$$

where BW is the bandwidth of one half of the global matrix.

The time complexity which accounts for elimination, backsubstitution, and latency is given by

$$TC = (2 \times \frac{BW^2}{2} + 2 \times BW) \times NNodes * 3 + C \quad (7.16)$$

In the parallel implementation scheme, the corresponding time complexity is given by

$$TC = (BW^2 + 2 \times BW) \times O(NNodes * 3/P) + NNodes * 3 \times t_c \times \log P + C_2 \quad (7.17)$$

For the 2475-element example, the sequential run time on a DEC station is 8 hours, whereas the run time using 32 processors is approximately 42 minutes, which represents an absolute speedup of ($Speedup \approx \frac{8 \times 60}{42} = 11.42$).

In testing this algorithm, it was found that the algorithm is sensitive to data values as the implementation does not perform pivoting, in order not to disturb the symmetry of the global matrix. A solution for overcoming this data sensitivity is to store the global matrix in double precision. Unfortunately, this strategy requires an increase in the number of processors in order to obtain sufficient memory. The LU Cholesky decomposition method, which is another variation of the GE technique, was found to improve the performance.

LU decomposition

The most efficient form of Gaussian elimination for solving a system of equations ($SX=Y$), when S is symmetric and positive definite, is the LU decomposition method. In this method, we write

$$S = L U \quad (7.18)$$

where, from symmetry considerations, $U = L^T$

The most efficient parallel implementation of the technique involves storing the global matrix in columns, rather than in rows. Processor i stores columns $(i, i+P, \dots)$. The elements of U are stored in place of the global matrix, thus eliminating the need for additional memory for U . The processor which stores the i th column of S computes U_{ii} from its local data. This value and the i th column of S are then broadcast to all other processors so that the rest of the elements of the i th row of U can be computed.

The backsubstitution consists of two phases. In phase one, we solve the problem ($L\underline{T} = \underline{Y}$), where \underline{T} is a temporary vector. In the second phase, we solve the equation ($U\underline{X} = \underline{T}$).

In the first phase, the elements T_i , from $i=1$ to N , are calculated. The processors store the columns of U , which are actually rows of L . After finding an element of the vector \underline{T} , the processor broadcasts this value to the other processors where it is used to update the sum of the products of the corresponding row elements and the elements of the \underline{T} vector. After receiving the previous $i-1$ elements, the processor that has the i th column of (S) finds the solution of T_i , and then broadcasts its value.

The second phase ($U\underline{X} = \underline{T}$) is different because the matrix U is stored in columns. The elements X_i are found from $i=N$ to 1 . The processor that finds the element X_j , broadcasts the product vector of X_j times the j th column of U to the

Table 7.1: Run time with varying number of processors

Number of processors	1	4	32	64
Time (hr:min:sec)	1:29:45	26:34	11:03	8:47

other processors. These product vectors are then used in finding the solutions for X_i , $i < j$. After all elements of X are found, the results are accumulated in one processor to be used for computing other quantities of interest, such as coil impedance.

Time complexity and speedup The time complexity formulation is similar to the complexity discussed for Gaussian elimination.

For the example on hand, the run time using 32 processors is approximately 40 minutes. The absolute speedup over the DEC stations, with $P=32$ is

$$\text{Speedup} \approx \frac{8 * 60}{40} = 12 \quad (7.19)$$

To get relative speedup data, a smaller size mesh whose global matrix can be stored in the memory of one node with 16 MB was considered. A mesh with 910 elements, 1232 nodes, and 128 bandwidth was created. The run time, with varying number of processors, is recorded in Table 7.1 and the speedup is shown in Figure 7.3.

From Figure 7.3, it can be seen that the speedup value saturates with increasing number of processors. Since POD models require several runs of the measurement model, speedup can be further improved by performing simultaneous runs. For example, for an NCUBE of dimension 10 and a speedup of 12, obtained using a hypercube of dimension 5, the effective speedup will be $12 \times 2^5 = 384$.

By obtaining such values of speedup, building the POD model becomes practical.

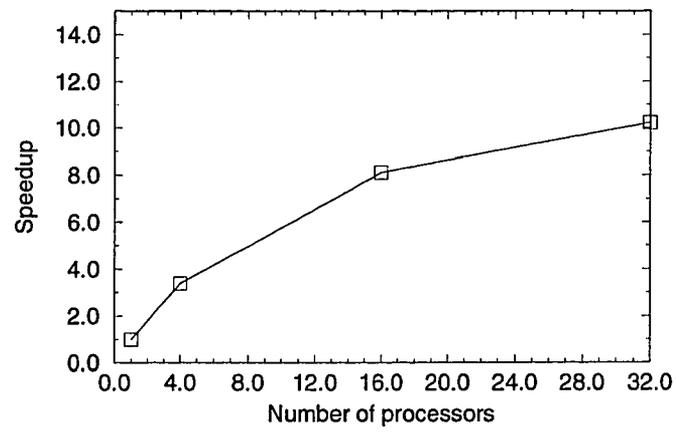


Figure 7.3: Speedup for LU decomposition algorithm

CHAPTER 8. NEURAL COMPUTING APPROACH FOR SOLVING INVERSE PROBLEM IN ELECTROMAGNETIC NDE

A novel approach to parallel computing is the more recent neurocomputer that depends on building an artificial neural network composed of numerous simple computational nodes or neurons. Two approaches are presented here to solve the inverse problem in electromagnetic NDE using neural networks. The first approach depends on a direct solution of the governing integral equation and is done using a Hopfield type neural network. The second approach develops a mathematical transform between the input and output space of the problem.

Neural networks initially emerged as an attempt to mimic the biological nervous system with respect to both architecture and information processing strategies. The somas of biological neurons are represented by a summing unit with a nonlinear transfer function, and synapses between the neurons are modeled by weighted interconnections.

The primary advantage of using neural networks is that it is potentially stable. This is due to the parallel distributed processing structure of neural networks, as well as the high degree of interconnectivity and feedback. Neural networks also offer the characteristics of generalization and error tolerance. A neural network with a large number of neurons will continue to function properly even in the case of failure of a

few neurons. A wide range of neural network architecture, neuron characteristics and training algorithm have emerged over the years to address a variety of problems. A survey of the historical background and development of the subject of neural networks can be found in several publications [75].

Solution of inverse problems in electromagnetics using Hopfield neural networks

In contrast to “forward problems”, where the system output is predicted using the model parameters, “inverse problems” involve the estimation of model parameters from the measured system response. The problem is, in general, ill-posed and regularization is required to restore stability [76, 77].

Mathematically, inverse problems can be modeled by volume integral equations. In electromagnetic problems, these equations describe the nonlinear relationship between the scattered field and the scattering object. In particular, the Fredholm integral equations of the first and second kind, commonly used for modeling a wide variety of physical processes, are expressed as [78]

$$g(x) = h(x) + \lambda \int_a^b f(y)K(x,y)dy \quad c < x < d \quad (8.1)$$

where $g(x)$, the measured data, and $K(x,y)$, the kernel or Green’s function for the geometry, are known to be bounded and continuous, λ is a possibly complex valued parameter, $f(x)$ is the unknown function, and a , b , c , and d are constants. If $h(x) = 0$, the equations are said to be homogeneous equations of the first kind. In operator form, the equation of the first kind can be written as

$$Kf = g \quad (8.2)$$

In discrete form, equation (8.2) is equivalent to a system of algebraic equations

$$A\mathbf{x} = \mathbf{b} \quad (8.3)$$

in which the matrix A is severely ill-conditioned, owing to the near linear dependence of the adjacent rows and columns. The matrix representation for the integral equation of the second kind is of the form

$$(I - \lambda A)\mathbf{x} = \mathbf{b} \quad (8.4)$$

where the near singularity of A is not a source of problem unless λ is large. As a result, these equations are relatively more tractable [79, 80] than the equation of the first kind.

The Fredholm equation of the first kind must be stated more precisely as

$$\int_a^b K(x, y)f(y)dy = g(x) + \epsilon(x) \quad (8.5)$$

where $\epsilon(x)$ represents the measurement noise.

In most cases, it can be shown that a family of solutions to satisfy equation (8.5) exists (nonuniqueness), and that small errors in the measurement g result in large oscillations in the solution (ill-posedness). These difficulties are resolved by applying regularization procedures, whereby the ill-posed inversion problem can be replaced by a stable minimization problem [81]. Instead of solving $Kf = g$ directly, one minimizes the quadratic functional

$$\| Kf - g \|^2 + \lambda \| L(f) \|^2 \quad (8.6)$$

where λ is the Lagrange multiplier and L is a linear constraint operator that has a smoothing or stabilizing effect on the solution procedure. A common choice for $L(f)$ is f'' , the second derivative of f [82, 83].

A novel strategy employing neural networks in an iterative approach for solving integral equations encountered in electromagnetic inverse problems using Hopfield network is presented. The iterative approach to the solution of nonlinear integral equations is reviewed [84] followed by a brief introduction to Hopfield networks.

Iterative inversion techniques

Consider the Helmholtz equation for electromagnetic fields

$$(\nabla^2 + k^2)\phi(\underline{r}) = -j\omega\mu\underline{J}(\underline{r}') \quad (8.7)$$

subject to generalized homogeneous boundary condition [1]

$$\alpha_1\phi(\underline{r}_s) + \alpha_2\frac{\partial\phi(\underline{r}_s)}{\partial n} = 0$$

where $\underline{J}(\underline{r})$ is the source (electric current density), $\phi(\underline{r})$ represents the field, \underline{r}_s is on the enclosing surface S, and \hat{n} is an outward directed unit vector.

The integral formulation of the problem is obtained using the Green's function, which is basically the solution to the fundamental equation

$$(\nabla^2 + k^2)G(\underline{r}, \underline{r}') = -\delta(\underline{r} - \underline{r}') \quad (8.8)$$

The free-space Green's function for the Helmholtz equation in spherical coordinates can be shown to be [85]

$$G(\underline{r}, \underline{r}') = \frac{e^{-jk|\underline{r}-\underline{r}'|}}{4\pi|\underline{r}-\underline{r}'|} \quad (8.9)$$

and, in an axially symmetric geometry [85],

$$G(\underline{r}, \underline{r}') = \frac{j}{4}H_0^{(2)}(k|\underline{r}-\underline{r}'|) \quad (8.10)$$

where $H_0^{(2)}$ is the Hankel function of the second kind.

In electromagnetic scattering problems, the scattered field, due to an inhomogeneity of complex permittivity ϵ_{cf} embedded in a homogeneous medium of conductivity ϵ_c^0 , is assumed to be produced by an equivalent source distribution. Here, the inverse problem solution involves the estimation of the spatial distribution of the source $\epsilon_{cf}(\underline{r})$. The net measured field is decomposed into a superposition of two parts

$$E(\underline{r}) = E_0(\underline{r}) + E_s(\underline{r}) \quad (8.11)$$

The first term $E_0(\underline{r})$ is produced by the source in the background while the second term represents the field scattered by the inhomogeneous medium.

For a scattered TE wave polarized in the y direction, we can explicitly show the two scalar components in the above equation to be [86]

$$E_y(\underline{r}) = E_{0y}(\underline{r}) + \int_{\text{inhomogeneity}} G(\underline{r}, \underline{r}') k_0^2 \delta\epsilon_c(\underline{r}') E_y(\underline{r}') dv' \quad (8.12)$$

where

$$\delta\epsilon_c(\underline{r}) = \epsilon_{cf}(\underline{r}) - \epsilon_c^0(\underline{r}) \quad (8.13)$$

This is an exact, nonlinear integral equation, which can be solved iteratively using a first order Born approximation. This method neglects the scattered component in the first iteration using

$$E_y^{(0)}(\underline{r}) = E_{0y}(\underline{r}) \quad (8.14)$$

thereby linearizing the inversion equation.

The iterative inversion procedure consists of the following steps:

1. Start with an initial guess for $\epsilon_{cf}^0(\underline{r})$
2. Apply Born approximation to equation (8.12) to obtain the first iteration,

$$E_y^{(1)}(\underline{r}) = E_y^{(0)}(\underline{r}) + \int_{\text{inhomogeneity}} G(\underline{r}, \underline{r}') k_0^2 \delta\epsilon_c(\underline{r}') E_y^{(0)}(\underline{r}') dv'$$

3. In the i^{th} iteration, solve the integral equation

$$E_y^i(\underline{r}) - E_y^{(i-1)}(\underline{r}) = \int_{\text{inhomogeneity}} G(\underline{r}, \underline{r}') k_0^2 \delta\epsilon_c(\underline{r}') E_y^{(i-1)}(\underline{r}') dv' \quad (8.15)$$

for $\delta\epsilon_c$

4. Update $\epsilon_{cf}(\underline{r})$ and go to Step 3.

The iterative procedure is continued until the difference between the measured field and computed field at the receiver location is smaller than a specified value. A new strategy for solving integral equations is presented in the following sections. The proposed strategy employs a Hopfield network which is described next.

Hopfield neural network

The basic organization of a Hopfield network [87] is shown in Figure 8.1 . Each neuron in the network is represented by an operational amplifier and the relation between the output v_i and input u_i of the i th amplifier is given by a transfer characteristic s with $v_i = s(u_i)$. The input of each amplifier is connected to ground through a resistor ρ_i in parallel with a capacitor C_i to simulate the delay of the response of a biological neuron. The output of neuron i is connected to the input of neuron j by a

synapse represented by a conductance $T_{ij} = 1/R_{ij}$. If the synapse is inhibitory, the conductor is connected to an inverted version of the output of the amplifier.

The differential equation describing the dynamics of amplifier i is given by

$$C_i \frac{du_i}{dt} = \sum_j \frac{v_j - u_i}{R_{ij}} - \frac{u_i}{\rho_i} + I_i \quad (8.16)$$

where

- u_i is the input potential for amplifier i
- v_i is the output potential for amplifier i
- I_i is the external input current for amplifier i

Equation (8.16) can be written in the form

$$C_i (du_i/dt) = \sum_j T_{ij} v_j - u_i/R_i + I_i \quad (8.17)$$

where: $1/R_i = 1/\rho_i + 1/\sum_j R_{ij}$

The weight T_{ij} represents a conductance connecting neurons i and j .

The output voltage v_i of neuron i is related to its input u_i by the relationship,

$$v_i = s(u_i)$$

The performance of the network is governed by the global function which is the energy function E given by [87]

$$E = -\frac{1}{2} \sum_i \sum_j T_{ij} v_i v_j + \sum_i \frac{1}{R_i} \int_0^{v_i} s_i^{-1}(\xi) d\xi - \sum_i I_i v_i \quad (8.18)$$

Hopfield showed that, as the amplifier gain goes to infinity, the continuous network with a symmetric weight matrix T will have stable states that correspond to

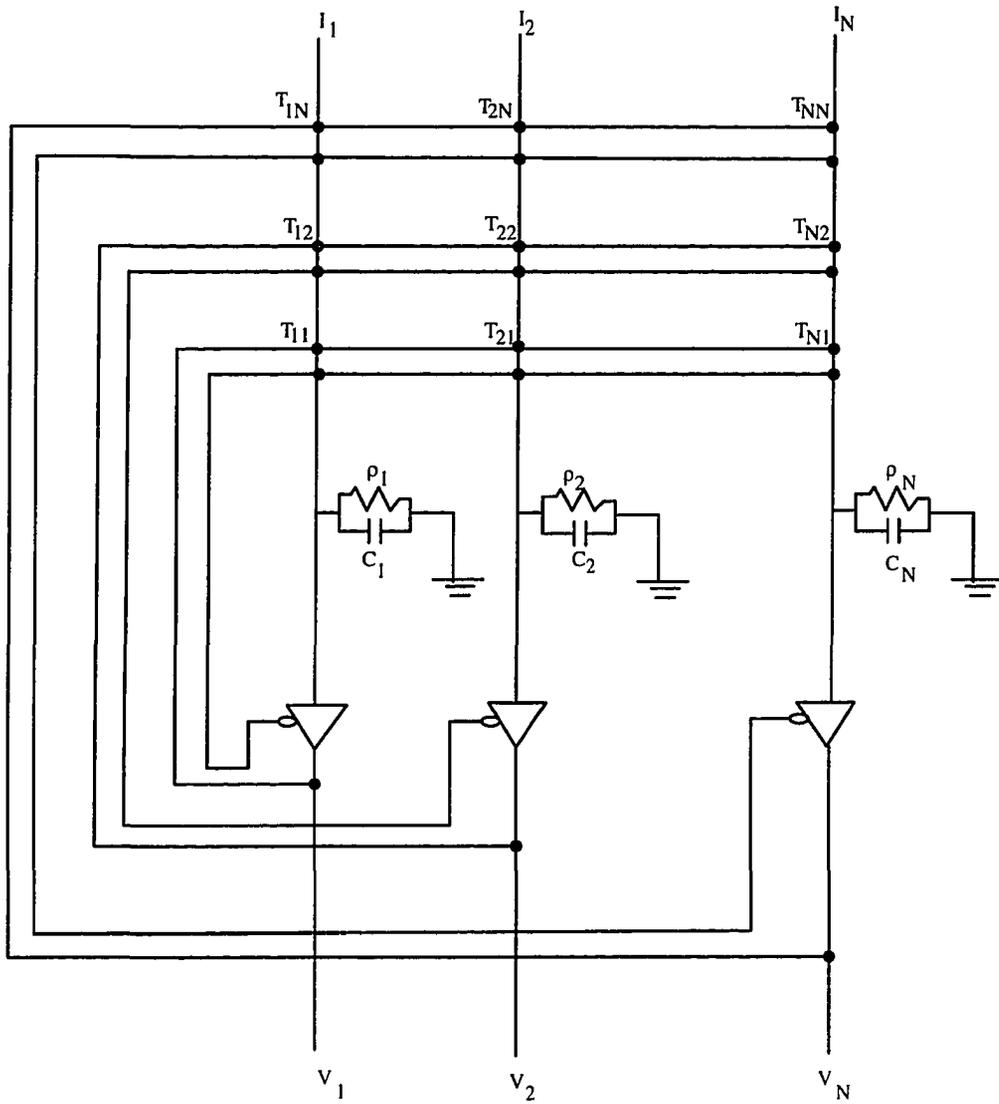


Figure 8.1: Hopfield neural network

corners of a hypercube which, in turn, correspond to the stable states of the digital-stochastic network. For finite gain, the second term in the right hand side of equation (8.18) becomes significant, and the minima are displaced from the corners towards the interior of the cube.

Equations (8.17) and (8.18) can be written in vector form as follows:

$$C \frac{d\underline{u}}{dt} = T\underline{V} - R^{-1}\underline{U} + \underline{I} \quad (8.19)$$

$$E = -\frac{1}{2}\underline{V}^T T\underline{V} + \underline{r}^T \hat{\underline{S}} - \underline{V}^T \underline{I} \quad (8.20)$$

where:

$$C := \text{diag}[C_1, C_2, \dots, C_n], R := \text{diag}[R_1, R_2, \dots, R_n]$$

$$\hat{S}_i := \int_0^{v_i} s_i^{-1}(\xi) d\xi$$

$$r_i := 1/R_i$$

For symmetric T, we get:

$$\begin{aligned} \nabla E &= -T\underline{V} + R^{-1}\underline{U} - \underline{I} \\ &= -C \frac{d\underline{U}}{dt} \end{aligned} \quad (8.21)$$

The time derivative of E is given by

$$\begin{aligned} \frac{dE}{dt} &= \dot{\underline{V}}^T \nabla E(\underline{V}) \quad (\text{chain rule}) \\ &= -C \dot{\underline{V}}^T \dot{\underline{U}} \\ &= -C \check{\underline{S}}^T(\underline{V}) \hat{\underline{V}} \end{aligned} \quad (8.22)$$

where:

$$\begin{aligned} \check{\underline{S}} &= \left[\frac{d}{dv_1} s_1^{-1}(v_1), \frac{d}{dv_2} s_2^{-1}(v_2), \dots, \frac{d}{dv_n} s_n^{-1}(v_n) \right]^T \\ \hat{v}_i &= \dot{v}_i^2 \end{aligned} \quad (8.23)$$

Since we choose $s_i^{-1}(v_i)$ to be a monotonic increasing function, from equation (8.22) we can conclude that $(d E/d t)$ will always be negative. Thus, the network will converge to a minimum of E and will not oscillate. This property of guaranteed and stable convergence of the Hopfield network to the energy minima has been exploited to solve many nonlinear optimization problems. A Hopfield network approach for the solution of an integral equation for the electromagnetic source is presented.

A simple transfer function for a neuron is chosen to be a linear transfer function with

$$s_i = \begin{cases} \alpha u_i & |u_i| \leq \gamma/\alpha \\ \gamma & u_i > \gamma/\alpha \\ -\gamma & u_i < -\gamma/\alpha \end{cases} \quad (8.24)$$

where α is the amplifier gain and γ is the saturation level.

For this amplifier transfer function, the neural network energy function can be written in matrix form as

$$E = -\frac{1}{2} \underline{V}^T T \underline{V} + \frac{1}{2} \underline{V}^T T_r \underline{V} - \underline{V}^T \underline{I} \quad (8.25)$$

where

$$\begin{aligned} T_r &:= \text{diag} [1/(\alpha_1 R_1) \ 1/(\alpha_2 R_2) \ \dots \ 1/(\alpha_n R_n)] \\ \underline{V}^T &:= [v_1 \ v_2 \ \dots \ v_N] \\ \underline{I}^T &:= [I_1 \ I_2 \ \dots \ I_N] \end{aligned}$$

The amplifiers parameters (α, γ) in equation (8.24) are obtained using theoretical results derived in [88]. The first theorem relates the norm of \underline{U} with γ , the elements

of T matrix, and the input vector \underline{I} . This result can be used to choose the value of γ [16].

Theorem: All trajectories of output vector \underline{V} converge to a bounded region in the state space.

The second theorem relates to the stability of point ($\underline{V} = 0$). We need the point ($\underline{V} = 0$) to be unstable so that the output will converge to a non-trivial solution. The point ($\underline{V} = 0$) is found to be unstable if for any i , $1 \leq i \leq n$ [16]:

$$\lambda_i > \frac{1}{\alpha R_i} \quad (8.26)$$

Equation (8.26) can be used in determining an appropriate value for α .

With the choice of network parameters presented above, the time derivative of the energy function is always less than or equal to 0 if the transfer function of the amplifiers is monotonically increasing. An implication of this property is that the network is guaranteed to converge to a minimum of the cost function which, in turn, translates into a solution to the inverse problem. As is the case with all iterative minimization algorithms, the Hopfield network does not necessarily converge to the global minimum of the cost function.

A strategy for enhancing the prospects of convergence to a global minimum is presented in Figure 8.2, where the output of the minimization network is feedback via a Hopfield linear programming network.

Linear programming network Linear programming networks are described in [89], where the minimization of cost function is performed subject to linear constraints:

$$D\underline{V} \geq \underline{B}$$

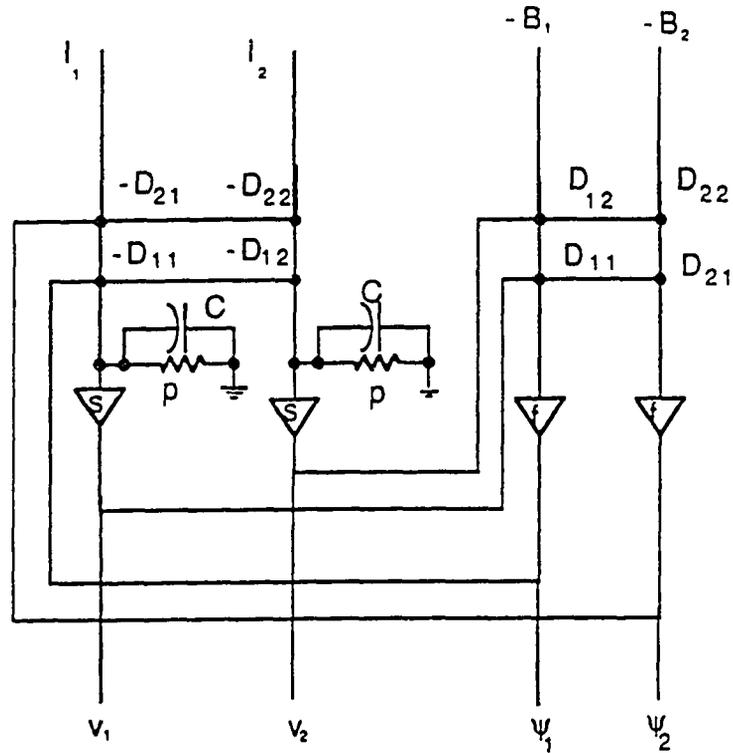


Figure 8.2: Linear programming network

In the network shown in Figure 8.2, the amplifiers marked "f" have transfer functions f such that for the i 'th amplifier, the output ψ_i is related to the input u_i by

$$\psi_i = f(u_i)$$

The response time of "f" amplifiers is negligible compared to "s" amplifiers.

The differential equation describing the dynamics of the network can be written in the form [89]:

$$C \frac{d\mathbf{U}}{dt} = D^T \mathbf{f}(\mathbf{U}) - R^{-1} \mathbf{U} + \mathbf{I}$$

$$= D^T f(D\underline{V} - \underline{B}) - R^{-1}\underline{U} + \underline{I} \quad (8.27)$$

The energy function of the network is similarly derived as:

$$E = \underline{a}^T \mathcal{F}(D\underline{V} - \underline{B}) + \underline{r}^T \hat{\underline{S}} - \underline{V}^T \underline{I} \quad (8.28)$$

where:

$$\underline{a}^T = [1 \ 1 \ \dots \ 1]$$

$$\mathcal{F}_i := \int_0^{v_i} f_i(\xi) d\xi$$

$$\hat{\underline{S}}_i := \int_0^{v_i} s_i^{-1}(\xi) d\xi$$

Therefore :

$$\nabla E = -C \frac{d\underline{U}}{dt}$$

$$\begin{aligned} \frac{dE}{dt} &= \dot{\underline{V}}^T \nabla E(\underline{V}) \\ &= -C \dot{\underline{V}}^T \dot{\underline{U}} \\ &= -C \check{\underline{S}}^T(\underline{V}) \hat{\underline{V}} \end{aligned} \quad (8.29)$$

where $\check{\underline{S}}$ and $\hat{\underline{V}}$ are given in equation (8.23).

Equation (8.29) is similar to equation (8.22), indicating that the derivative of the energy function of a linear programming network is also negative. Thus, the network will converge towards a minimum of the energy function and will not oscillate.

We use the circuit as shown in Figure 8.3. The energy minimization network is given in Figure 8.1, while the linear programming network is shown in Figure 8.2. The input \underline{U} of amplifiers of the minimization network is connected to the external input \underline{I} for the linear programming network through a switch. The output $\underline{\psi}$ of the linear programming network is connected to input \underline{I} of the minimization network.

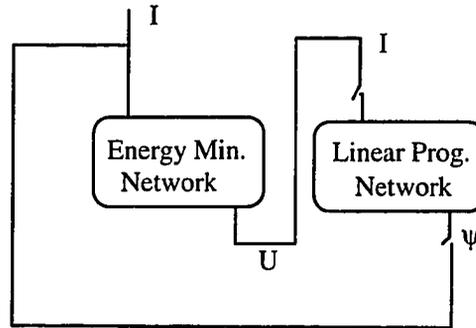


Figure 8.3: Neural network with feedback circuit

The transfer function of the s amplifiers of the linear programming network is similar to the transfer function of the energy minimization network. The transfer function of the f amplifiers of the linear programming network is chosen as:

$$\psi_j = f(z_j)$$

where:

$$f(z) = \begin{cases} 0 & -\epsilon \leq z \leq \epsilon \\ -z & z \leq -\epsilon, z \geq \epsilon \end{cases}$$

The linear programming circuit will give 0 output when

$$-\epsilon \leq (F\underline{V} - \underline{g}) \leq \epsilon \quad (8.30)$$

The switches between the two networks are left off for a period of a few time constants of the minimization network. The switches are then turned on. If the minimization network is trapped in a local minimum, the output ψ of the linear programming network will excite the minimization network again and move the energy

function away from the local minimum. When the condition of equation (8.30) is satisfied, both networks will stop, and the output of the energy minimization network is the desired solution.

An algorithm for solving inverse problems using neural networks is presented in the following section.

Inverse problem in electromagnetics

One commonly encountered inverse problem involves the reconstruction of the permittivity and conductivity profiles of a medium from electromagnetic measurements. Assuming axial symmetry around the surveying tool (Figure 8.4), the forward problem for the configuration is shown in [90], and the integral equation representation is given by [91]

$$E_\phi(\rho, z) - E_\phi^0(\rho, z) = \int_{z'=-\infty}^{\infty} \int_{\rho=R_i}^{R_o} G^0[(\rho - \rho'), (z - z')] k_0^2 \delta\epsilon_c E_\phi(\rho', z') d\rho' dz' \quad (8.31)$$

$$= \int_{z'=-\infty}^{\infty} \int_{\rho=R_i}^{R_o} G^0[(\rho - \rho'), (z - z')] k_0^2 \delta\epsilon_c E_\phi^0(\rho', z') d\rho' dz' \quad (8.32)$$

Equation (8.32) is obtained using Born approximation. E^0 is the electric field corresponding to the initially known complex permittivity distribution ϵ_c^0 , E is the electric field corresponding to a distribution $\epsilon_c^0 + \delta\epsilon_c$, and G is the Green's function. R_i and R_o are the inner and outer radii of the medium. For an axially symmetric problem with a circular coil of radius a as the source, the relation between electric

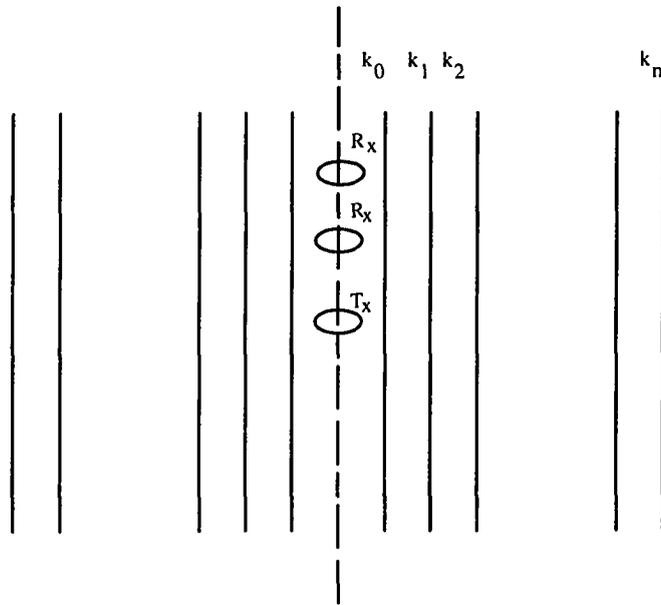


Figure 8.4: Coaxial multilayered media

field and Green's function is given by

$$E_\phi^0 = j\omega\mu I a G^0 \quad (8.33)$$

where ω is the radial frequency, μ is the magnetic permeability, and I is the current. Taking the Fourier transform of equation (8.32) and using equation (8.33), we can write equation (8.32) in terms of the Fourier transform g of Green's function

$$g(R, a, k_z) - g^0(R, a, k_z) = 2\pi k_0^2 \int_{R_I}^{R_o} d\rho' \delta\epsilon_c(\rho') g^0(\rho', a, k_z) g^0(R, a, k_z) \quad (8.34)$$

Expressing the difference between g and g^0 as function h_c , the change in complex permittivity $\delta\epsilon_c$ as function Q_c and the kernel $2\pi k_0^2 g^0(\rho', a, k_z) g^0(R, a, k_z)$ as K_c , equation (8.34) is reduced to the form

$$h_c(k_z) = \int_{R_I}^{R_o} d\rho' Q_c(\rho') K_c(\rho, k_z) \quad (8.35)$$

Equation (8.35) is a Fredholm integral equation, where the discrete-form matrix equation is in terms of complex variables. The equation can be transformed into an equation in real variables by considering the real and imaginary parts separately.

Alternatively, equation (8.35) can be converted into a real valued equation by using only amplitude measurements of the scattered electric field [92]. From equation (8.34), we get

$$\frac{\delta g^0}{g^0} = \int_{R_I}^{R_o} d\rho' Q_c(\rho') \frac{K_c(\rho', k_z)}{g^0} \quad (8.36)$$

Equating the real part of both sides, we get

$$\int_{R_I}^{R_o} d\rho' \left\{ \Re \left\{ \frac{K_c}{g^0} \right\} \Re \{Q_c(\rho')\} - \Im \left\{ \frac{K_c}{g^0} \right\} \Im \{Q_c(\rho')\} \right\} = \delta \log |g^0| \quad (8.37)$$

where \Re indicates the real part, and \Im is the imaginary part, and

$$\Re\left\{\frac{\delta g^0}{g^0}\right\} = \delta \log |g^0|$$

Equation (8.37) is in the form of a real valued Fredholm equation, which can be solved using a Hopfield network formulation.

Problem formulation

Expressing equations (8.35) and (8.37) by the Fredholm integral equation

$$\int_a^b k(x, y)f(y)dy = g(x) \quad (8.38)$$

where $g(x)$ represents the measured data, $f(y)$ represents the source function or the system states, and $k(x,y)$ represents the kernel of the transformation, the objective of inverse problem is then to solve for the source or state function from known measurements. Representing the source function z in equation (8.38) as a linear combination of basis functions [93], we get

$$f(x) = \sum_{i=1}^N v_i R_i(x) \quad (8.39)$$

where R_i are basis functions, chosen to be sine or cosine functions and v_i are unknown coefficients. Equation (8.38) can be written in a matrix form as

$$F\underline{V} = \underline{g} + \underline{n} \quad (8.40)$$

where $F_{ij} = \int_a^b k(x, y_j)R_i(x)dx$, $g_i = g(x_i)$, $\underline{V} = (v_1, v_2 \dots v_N)^T$, and \underline{n} represents the noise.

The solution \underline{V} is obtained by minimizing the quadratic error cost function

$$E = \frac{1}{2}(\underline{F}\underline{V} - \underline{g})^T(\underline{F}\underline{V} - \underline{g}) + \lambda \underline{V}^T D \underline{V} + \lambda_1 (\Psi \underline{V} - f_p)^T (\Psi \underline{V} - f_p) \quad (8.41)$$

The first term in the cost function is the model error, the second term is a smoothness constraint imposed for minimizing the ill-posedness of the problem, and the third term represents the boundary conditions. Ψ is a diagonal matrix whose elements are basis functions R_i , f_p is the system state at a given point p , and λ and λ_1 are Lagrange multipliers. In electromagnetic scattering problems, f_p represents the complex permittivity and can be expressed as

$$f_p = [\epsilon_p, \sigma_p/\omega\epsilon_0]^T$$

The value of λ is chosen to compensate for the measurement noise in the cost function. An iterative technique for arriving at the optimal value of λ is given in [94]. λ_1 is chosen high enough to ensure that the boundary conditions are satisfied.

The cost function in equation (8.41) can be written more compactly in matrix form such as

$$E = -\frac{1}{2}\underline{V}^T T^\dagger \underline{V} - \underline{V}^T \underline{I}^\dagger \quad (8.42)$$

where:

$$\begin{aligned} T^\dagger &= -F^T F - \lambda D - \lambda_1 \Psi^T \Psi \\ \underline{I}^\dagger &= F^T \underline{g} + 2\lambda_1 \Psi^T \underline{\psi}_p \end{aligned}$$

By comparing equations (8.42) and (8.25), we can derive the design values for the weights and external input currents of the neural network as

$$\begin{aligned} T &= T^\dagger + T_r \\ \underline{I} &= \underline{I}^\dagger \end{aligned} \quad (8.43)$$

The inputs \underline{B} in Figure 8.2 are chosen to be equal to \underline{g} of equation (8.40), and the weights D of the network are obtained from F in equation (8.40).

Some initial simulation results are described in the following section.

Simulation algorithm

A modified gradient descent algorithm is used to simulate the minimization of the energy function. Gradient descent algorithms depend on starting with an initial value of \underline{V} , and iteratively changing \underline{V} , such that

$$\underline{V} = \underline{V} + \mu \nabla E \quad (8.44)$$

where μ is a small number that determines the step size, and ∇E is obtained from equation (8.42)

$$\nabla E = -T^\dagger \underline{V} - \underline{I}^\dagger \quad (8.45)$$

The procedure is terminated when the gradient of E becomes sufficiently small.

A modification for the steepest-descent algorithm is shown in [95]. We update \underline{V} such that, at iteration $(r+1)$, v_i is given in terms of v_i at iteration r as follows:

$$v_i^{r+1} = v_i^r + \mu^r \delta_i^r \quad (8.46)$$

where δ_i is given by [95]

$$\delta_i = \frac{[1 + (v_i^r)^2] \left(\frac{\partial E}{\partial v_i}\right)^r}{\left[\sum_{j=1}^N \left([1 + (v_j^r)^2] \left(\frac{\partial E}{\partial v_j}\right)^r\right)^2\right]^{\frac{1}{2}}} \quad (8.47)$$

and μ is a step factor.

The modified gradient descent procedure for implementing the neural network solution to the inverse problem is summarized by the following steps.

1. Choose an initial value for μ
2. Calculate E^r from (8.42), and $\delta_1^r, \dots, \delta_n^r$ from (8.47).
3. Calculate E_{temp} which is the energy corresponding to $(v_i^r + \mu^r \delta_i^r)$.

If $E_{temp} > E^{r-1}$ then:

$$\mu = \mu/4$$

go to step 2

4. Calculate the cosine of the angle θ between the vectors $\underline{\delta}^r$ and $\underline{\delta}^{r-1}$.

$$\cos \theta = \sum_{j=1}^N \delta_j^{r-1} \delta_j^r$$

5. If $\cos \theta < 0$, then:

$$\mu = \mu/4$$

go to step 2

6. Calculate μ^{r+1}

$$\mu^{r+1} = \mu^r (.5 + \cos^4 \theta)$$

7. If δ_i for all i is less than a small number δ :

then:

terminate;

else:

increment r , and go to step 2.

In simulating the forward problem, the frequency was chosen to be 10 MHz. The radii of the transmitter and the receiver were chosen to be 5 cm each.

The Lagrange multiplier λ should be chosen to be large in the first few iterations to give more weight to the smoothness constraint. As the iterations progress, the value of λ is decreased in order to expedite minimization of the first term of the energy function. Figure 8.3 shows the true permittivity ϵ , and loss tangent $\frac{\sigma}{\omega\epsilon_0}$ in the

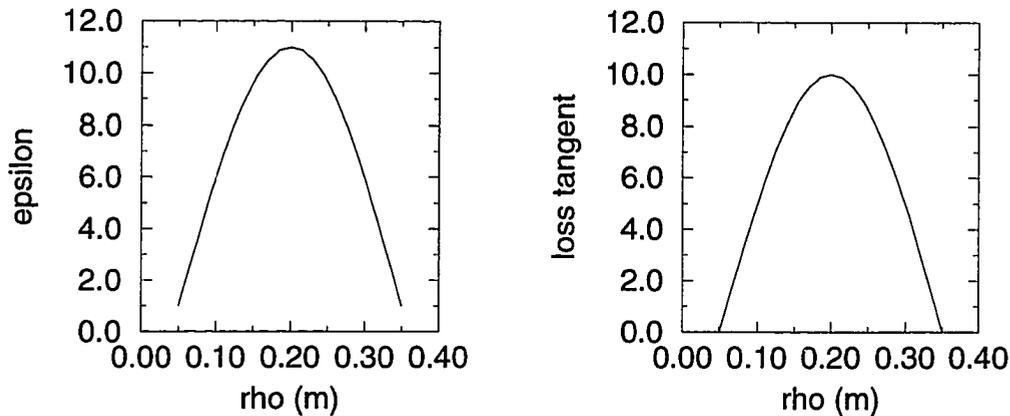


Figure 8.5: True distribution of relative permittivity and loss tangent

layer. Figures 8.5, 8.6 and 8.7 show the initial values and reconstructed values after the first, second and third iterations. Figures 8.8 and 8.9 show the reconstructed values using the amplitude of the scattered wave.

Application of weight space mapping and modular architectures for incremental learning networks

In the previous section, solutions to the inverse problem are attempted by directly solving the governing integral equation. In this section, an alternate approach is presented that depends on generating an appropriate mapping between the input and output space by training a multilayered perceptron (MLP) neural network. In addition, the characteristics of the conventional neural network are enhanced by building other features of the biological neural network into the network operation.

Human brains have many features that are highly desirable in automated sys-

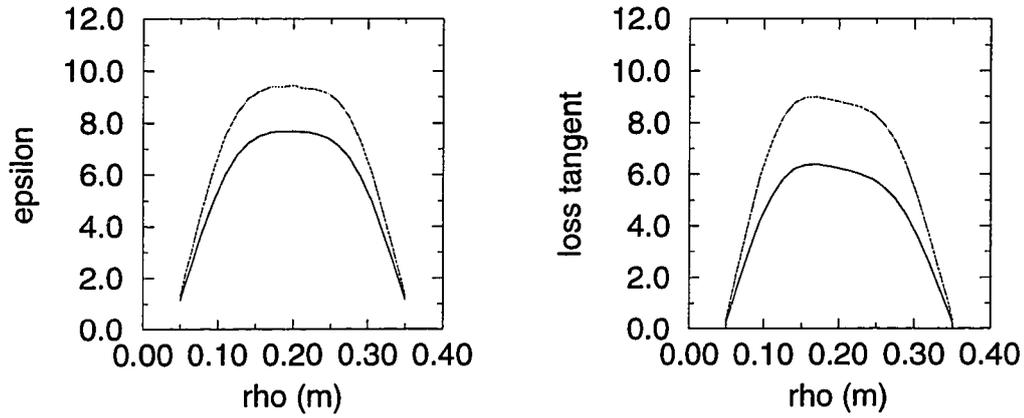


Figure 8.6: First, and second iterations

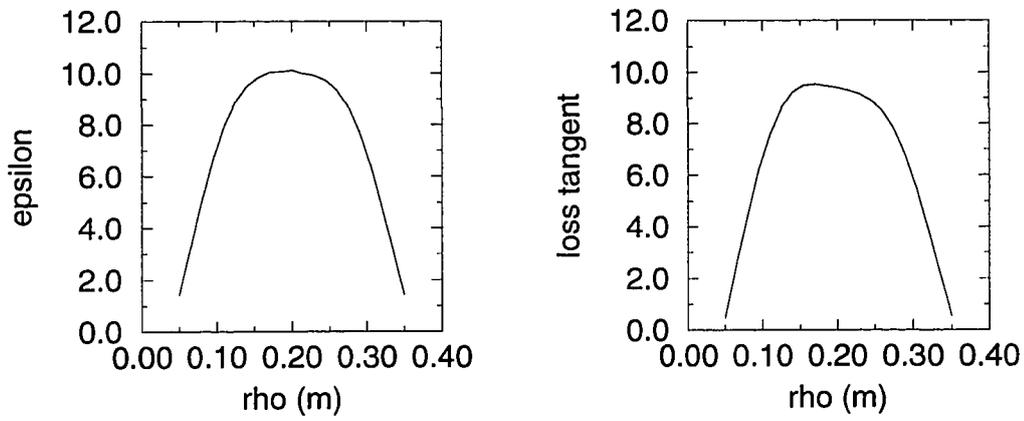


Figure 8.7: Third iteration

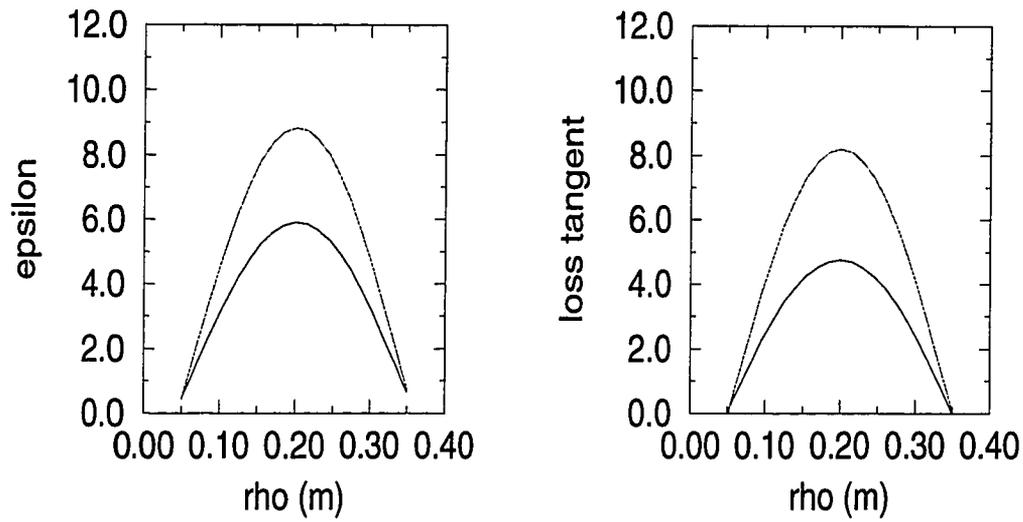


Figure 8.8: First, and second iterations using amplitude of measurements

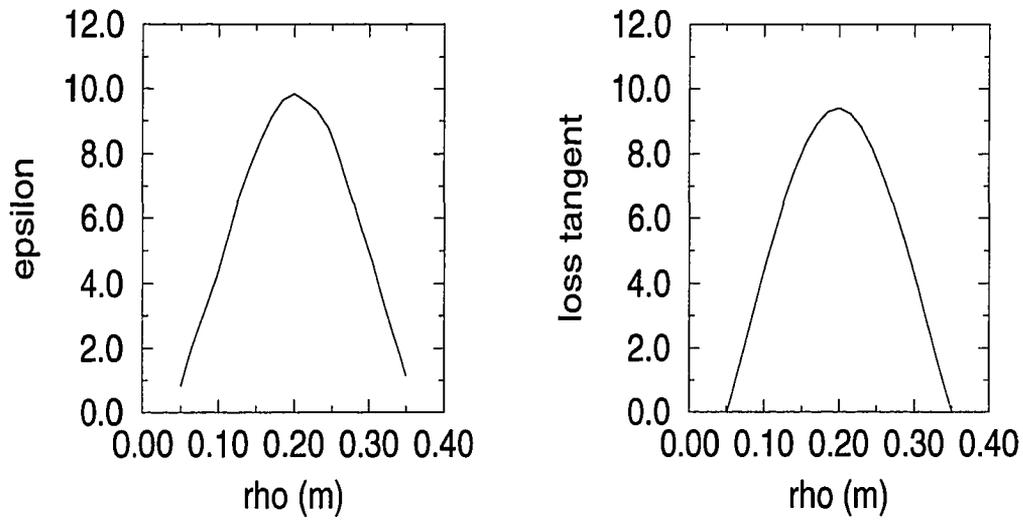


Figure 8.9: Third iteration using amplitude of measurements

tems. One of these features is the high degree of parallelism and interconnectivity among the neurons of the brain, which makes its performance robust, fault tolerant, and immune to noise. Human brains are also dynamic and adaptive to changing environments. The information stored in the brain keeps growing and changing with time. New information is learned without losing knowledge acquired earlier.

Artificial neural networks have the ability, with training, to generate complex decision surfaces in the feature space of pattern recognition problems. Such decision surfaces achieve enhanced classification performance, particularly in the case of nonlinearly separable, multiclass data sets. The commonly used networks are the multilayered perceptrons which are usually trained using the well-established back-propagation algorithm.

The back-propagation training algorithm provides a procedure for adapting the interconnection weights in order to learn the input-output mapping using the training data set. The back-propagation algorithm is based on the application of the gradient descent technique to minimize a selected cost/energy function typically chosen to be the quadrature error measure. The weights of the network are changed iteratively such that the error function is minimized in the steepest descent direction. A major issue related to this training algorithm is the speed of convergence which depends on the shape of the energy function. In general, this function can have a complex shape and the iterative scheme for updating weights may therefore exhibit oscillatory behavior resulting in long training times.

The training algorithm proposed uses a modified version of the steepest descent method, which involves transforming the weights to an appropriate space where the shape of the transformed energy function offers faster convergence. The procedure

allows the step size during the iteration process to be large and, at the same time, ensures that the error function decreases with iterations. The algorithm thereby approaches the minimum of the error function in a small number of iterations.

In addition to the increase of training time, due to the complexity of the error function, a substantial training time requirement is encountered in pattern recognition applications with a dynamically varying number of classes. Adding new classes to the problem requires the increase of the number of nodes at the output and at hidden layers and necessitates a retraining of the network. This is in contrast to the incremental and cumulative learning ability of the biological brain. A novel, dynamic, modular network architecture is presented in this section which makes use of past learning to hasten the learning of new classes and reduce the overall training time for such applications.

The overall incremental learning network (ILN) model is based on the multilayer perceptron architecture, a typical embodiment of which is shown in Figure 8.10. Results show that the proposed techniques result in a substantial reduction of the training time. Detailed description of the modification introduced to the standard back propagation algorithm and the proposed modular architecture network, along with typical implementation results, is presented next.

Modified steepest descent and error functions

The back-propagation training algorithm [96, 97] is based on the minimization of an energy function. The energy function is typically of a quadratic form derived using the error between the actual output O and desired output d . The cost function

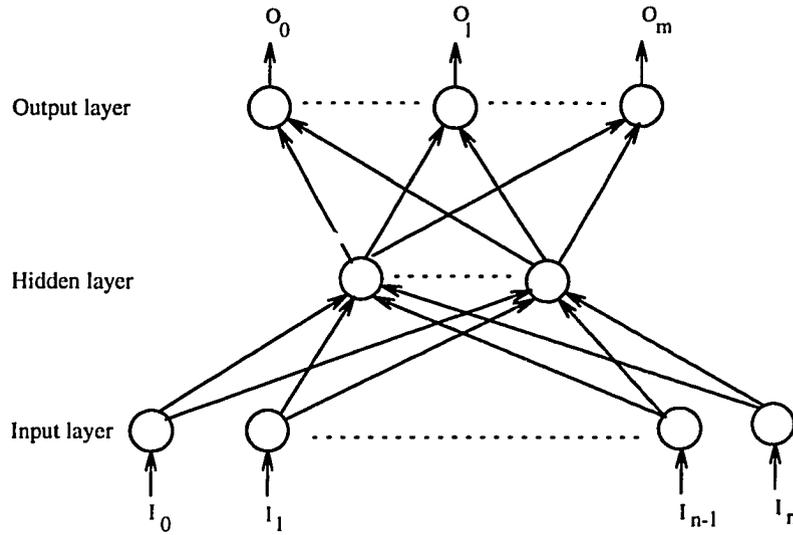


Figure 8.10: Multilayer perceptron

can be written as

$$E(W) = \frac{1}{2} \sum_{ri} (d_i - O_i^r)^2 \quad (8.48)$$

where r is the iteration number and i spans the output nodes.

Representing the output of the hidden layer nodes as V and the input to the output layer nodes as h , we have the output of neuron i given by

$$\begin{aligned} O_i &= g(h_i) \\ &= g\left(\sum_j w_{ij} V_j\right) \end{aligned}$$

where g is the transfer characteristic of the neurons, and w_{ij} is the weight representing the synapse that connects neuron i in the output layer to neuron j in the hidden layer. For a feed forward network with a single hidden layer, equation (8.48) can be

expanded as

$$E(W) = \frac{1}{2} \sum_{ri} \left[d_i - g \left(\sum_j w_{ij} g \left(\sum_k w_{jk} I_k^r \right) \right) \right]^2 \quad (8.49)$$

where I is the external input to the network, j spans the hidden layer nodes, and k spans the input layer nodes.

In each iteration r , the gradient descent algorithm changes the weights by an amount Δw which can be expressed as

$$\Delta w_{ij}^r = -\eta \frac{\partial E}{\partial w_{ij}} \quad (8.50)$$

$$= \eta \sum_r (d_i - O_i^r) g'(h_i^r) V_j^r \quad (8.51)$$

$$= \eta \sum_r \delta_i^r V_j^r \quad (8.52)$$

where the error function δ can be represented as

$$\delta_i^r = g'(h_i^r)(d_i - O_i^r) \quad (8.53)$$

The modified steepest descent method proposed in [95] can be understood by considering the two dimensional energy function in Figure 8.11, where the contours of the cost function are nearly circular in shape. It is seen that the steepest descent method provides the fastest approach for arriving at the desired minimum. However, the energy surface, in general, can take a “trough” like shape. Contours of such functions for the two dimensional case are illustrated in Figure 8.12. The minimization of this type of energy function using a steepest descent method results in a path that is oscillatory in nature as shown in Figure 8.12. Thus, the convergence of the method is slow, resulting in excessive training time.

A mathematical method that can be applied for overcoming this problem is to use a transformation function that maps the weights such that the transformed

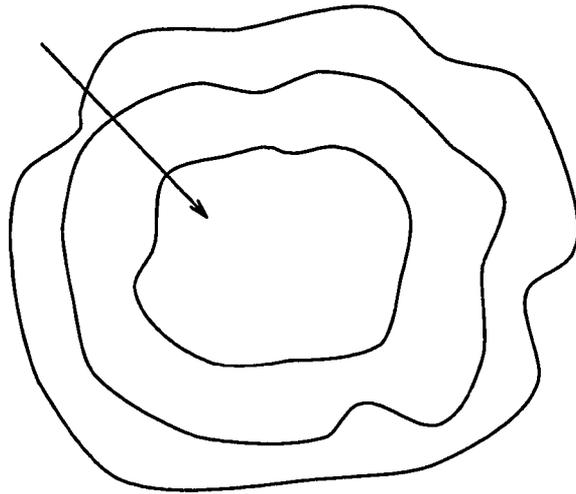


Figure 8.11: Near circular energy function

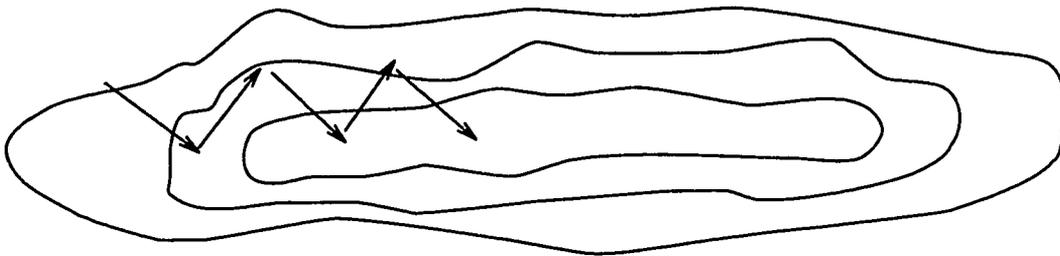


Figure 8.12: Trough energy function

energy function has contours of nearly equal dimensions along the different axes in the transformed weight space. Such a transformation can be achieved by magnifying small values of weights and saturating the large values.

A suitable candidate for accomplishing such a mapping is the arctan function. The iterative weight update equation is derived using the transformed weight values. This directs the minimization procedure towards the minimum of the error function faster. The implementation of the weight update equation in the transformed weight space is described below.

Mapping the weights w_{ij} to \hat{w}_{ij} using the transform

$$\hat{w}_{ij} = \tan^{-1} w_{ij} \quad (8.54)$$

the update rule for the weights in equations (8.50)-(8.53) is modified as follows

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial \hat{w}_{ij}} \quad (8.55)$$

$$= -\eta \frac{\partial E}{\partial w_{ij}} / \frac{\partial \hat{w}_{ij}}{\partial w_{ij}} \quad (8.56)$$

$$= -\eta \frac{\partial E}{\partial w_{ij}} (1 + w_{ij}^2) \quad (8.57)$$

As described earlier in equation (8.52), we can write the derivative of the energy function as

$$\frac{\partial E}{\partial w_{ij}} = \sum_r \delta_i^r V_j^r \quad (8.58)$$

Normalizing the weight update term, we have

$$\Delta w_{ij} = -\eta \frac{\frac{\partial E}{\partial w_{ij}} (1 + w_{ij}^2)}{\left[\sum_{lm} \left[\left(\frac{\partial E}{\partial w_{lm}} \right) (1 + w_{lm}^2) \right]^2 \right]^{\frac{1}{2}}}$$

where l and m span the nodes in the two layers that are connected by w_{ij}

In the case of a sigmoidal nodal transfer function, the output O_i is expressed as

$$\begin{aligned} O_i &= g(h_i) \\ &= \frac{1}{1 + \exp(-h_i)} \end{aligned} \quad (8.59)$$

which gives

$$g'(h_i^r) = O_i^r(1 - O_i^r)$$

and from equation (8.53), the error function in equation (8.58) is given by

$$\delta_i^r = O_i^r(1 - O_i^r)(d_i - O_i^r) \quad (8.60)$$

Alternatively, the energy surface can also be modified to a more tractable form by choosing alternate forms of candidate cost or nodal transfer functions as discussed in [98]. A popular cost function that can be derived from entropy considerations is of the form

$$E = \sum_{ir} \left[\frac{1}{2}(1 + d_i) \log \frac{1 + d_i}{1 + O_i^r} + \frac{1}{2}(1 - d_i) \log \frac{1 - d_i}{1 - O_i^r} \right] \quad (8.61)$$

with its derivative given by

$$\frac{\partial E}{\partial w_{ij}} = \sum_r \frac{1}{2} V_j^r g'(h_i^r) \left\{ \frac{-(1 + d_i)}{(1 + O_i^r)} + \frac{(1 - d_i)}{(1 - O_i^r)} \right\} \quad (8.62)$$

The corresponding error function reduces to a simple form

$$\delta_i^r = (d_i - O_i^r) \quad (8.63)$$

Yet another choice for the error function δ is the arctanh function, which can be interpreted as derived from differentiating an energy function that is different from the quadratic and entropy costs discussed earlier. For example, the function $(d_i - O_i^r)$ can be mapped to $\tanh^{-1}(d_i - O_i^r)$ giving δ in equation (8.63) to be

$$\delta_i^r = \tanh^{-1}(d_i - O_i^r) \quad (8.64)$$

and δ in equation (8.53) to be

$$\delta_i^r = g'(h_i^r) \tanh^{-1}(d_i - O_i^r) \quad (8.65)$$

In general, a weighted combination of different error functions can be used. For example, a weighted combination of the error functions described in equations (8.53) and (8.63) is expressed as

$$\delta_i^r = [g'(h_i^r) + \zeta](d_i - O_i^r) \quad (8.66)$$

where ζ is a weighting constant.

A combination of the error functions described in equations (8.65) and (8.63) is of the form

$$\delta_i^r = g'(h_i^r) \tanh^{-1}(d_i - O_i^r) + \zeta(d_i - O_i^r) \quad (8.67)$$

In the case of sigmoidal transfer functions, equations (8.66), and (8.67) reduce to

$$\delta_i^r = [O_i^r(1 - O_i^r) + \zeta](d_i - O_i^r) \quad (8.68)$$

$$\delta_i^r = O_i^r(1 - O_i^r) \tanh^{-1}(d_i - O_i^r) + \zeta(d_i - O_i^r) \quad (8.69)$$

The factor ζ is generally chosen to be a small value in the range [0,1]. Plots of various error functions described in equations (8.64), (8.68) and (8.69) are shown in Figures 8.13-8.17. The left-hand column represents the function when the desired output $d_i = 1$, and right-hand column is for $d_i = 0$.

A description of the training algorithm, using the modified steepest descent algorithm in conjunction with an appropriate transformation of the weight space, is given in the next subsection.

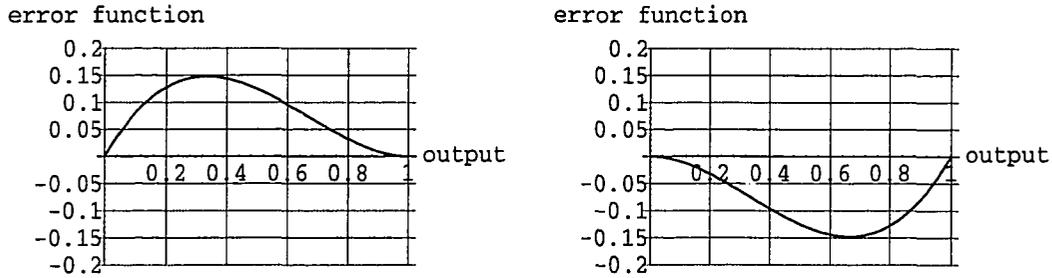


Figure 8.13: Error function in (8.68) $[\delta_i^r = [O_i^r(1 - O_i^r) + \zeta](d_i - O_i^r)]$, $\zeta = 0$

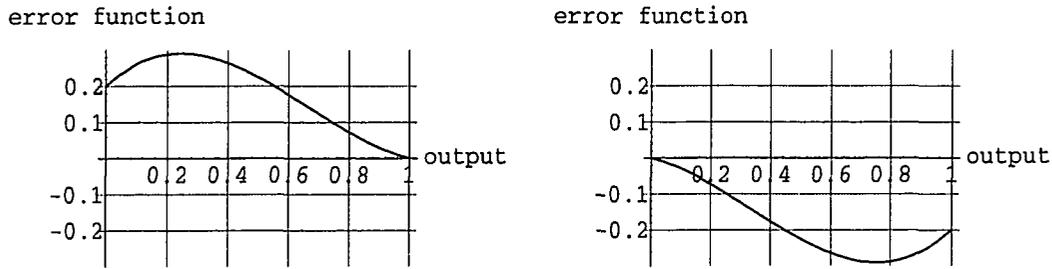


Figure 8.14: Error function in (8.68), $\zeta = 0.2$

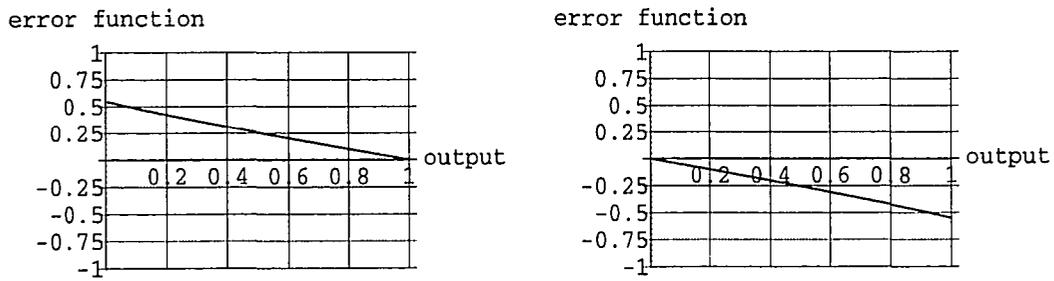
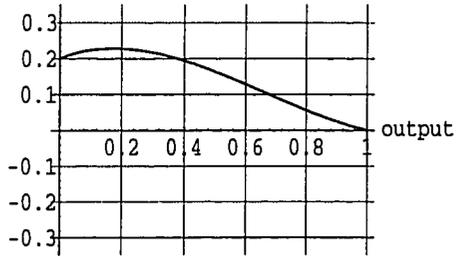


Figure 8.15: Error function in (8.64) $[\delta_i^r = \tanh^{-1}(d_i - O_i^r)]$

error function



error function

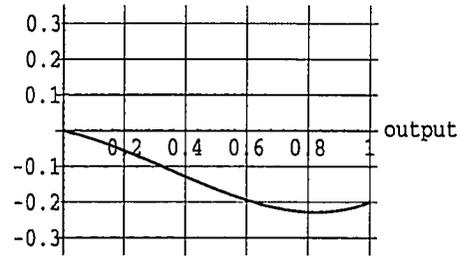
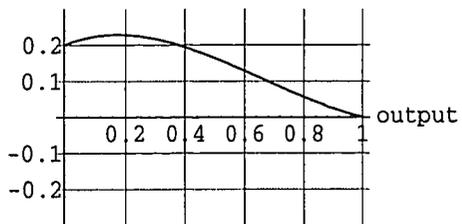


Figure 8.16: Error function in (8.69) $[\delta_i^r = O_i^r(1 - O_i^r) \tanh^{-1}(d_i - O_i^r) + \zeta(d_i - O_i^r)]$,
 $\zeta = 0$

error function



error function

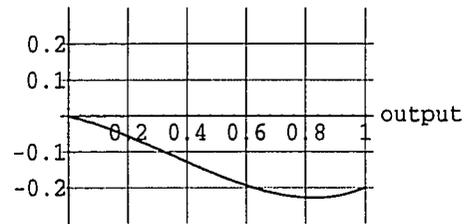


Figure 8.17: Error function in (8.69), $\zeta = 0.2$

Back-propagation algorithm using modified steepest descent procedure

Step 1: Choose an appropriate MLP network. The input and hidden layers may be augmented with a node of value “1”. Initialize network weights to small random values.

Step 2: Apply the first vector \underline{X}_1 for which the desired output \underline{d} is known.

Step 3: Calculate output vector \underline{Q} corresponding to this input.

Step 4: The weight update rule in iteration r is given as follows

$$w_{ij}^{r+1} = w_{ij}^r - \eta^{r+1} C_{ij}(r+1)x_i^{r+1} \quad (8.70)$$

where the normalized correction factor C_{ij} is

$$C_{ij}^{r+1} = \frac{[1 + (w_{ij}^r)^2] P_{ij}^r}{\left[\sum_{l,m} ([1 + (w_{lm}^r)^2] P_{lm}^r)\right]^{\frac{1}{2}} + \epsilon} \quad (8.71)$$

ϵ is a small number that serves to eliminate overflow conditions.

P_{ij}^r is defined as

$$P_{ij}^r = \delta_j^r x_i^r$$

x_i is either an output of node i in a hidden layer, or is an external input to the network, and δ_j^r is derived as follows:

(i) for output layer nodes, δ_j^r is given as in equations (8.68) or (8.69)

(ii) for hidden layers nodes,

$$\delta_i^r = [g'(h_i^r) + \zeta] \sum_l \delta_l^r w_{jl}^r \quad (8.72)$$

or,

$$\delta_i^r = g'(h_i^r) \tanh^{-1} \sum_l \delta_l^r w_{jl}^r + \zeta \sum_l \delta_l^r w_{jl}^r \quad (8.73)$$

The index l spans all nodes in the layers above node j .

The value of η in iteration equation (8.70) is made adaptive by using the following criterion:

If the angle θ between vectors \underline{C}^r and \underline{C}^{r-1} is acute, then the minimization is along the same direction, and the step size is increased to reach the minimum faster. However, if the angle is obtuse, then the weight update path is oscillatory indicating that the step size should be decreased.

Mathematically, the adaptive weight update procedure can be expressed as follows:

$$\cos \theta = \underline{C}^r \cdot \underline{C}^{r-1} \quad (8.74)$$

If $\cos \theta < 0$, then:

set η between 0 and 1

else :

$$\eta^{r+1} = \eta^r (.5 + \cos^4 \theta) \quad (8.75)$$

Step 5: Terminate when output \underline{Q} converges to the desired output \underline{d} .

Modular network architecture for incremental learning

Artificial neural networks have been fashioned to mimic the parallelism and interconnectivity features of human brains. However, networks developed to date lack

the flexibility and dynamic characteristics of human brains. MLP networks depend on learning all information in one learning session. For example, a network trained to recognize a specific number of classes has to be retrained from scratch when new classes are added, thereby resulting in complete loss of the old stored information.

A key feature in the process of human learning is the use of information learned earlier to make new learning sessions easier and faster. For example, learning advanced calculus is easier for a person with a good calculus background than for a person without requisite background knowledge. This concept is adopted here to develop a novel dynamic multilayer perceptron. The architecture of the network is modular and, in addition, the network training algorithm is based on the use of past learning to hasten the learning of new classes. The model is based on building an *accumulative learning* network that will be capable of dividing the information to be learned into smaller sets of information and individually processing the learning of each set. This strategy avoids convergence problems that frequently occur when a large amount of information is involved, and the corresponding error function is too complicated. The architecture of the proposed modular network is described next.

Network architecture

Figure 8.18 shows a modular network where the module labeled 'A' is a multilayer perceptron trained to recognize 'm' classes. When the network is required to learn a new class, module B is added to the original network. The outputs of the old network are treated as part of the hidden layer of the new module. During training, the weights from the old outputs to the new outputs are initialized to negative quantities. These weights work as an inhibitory synapses from the old outputs to the new outputs.

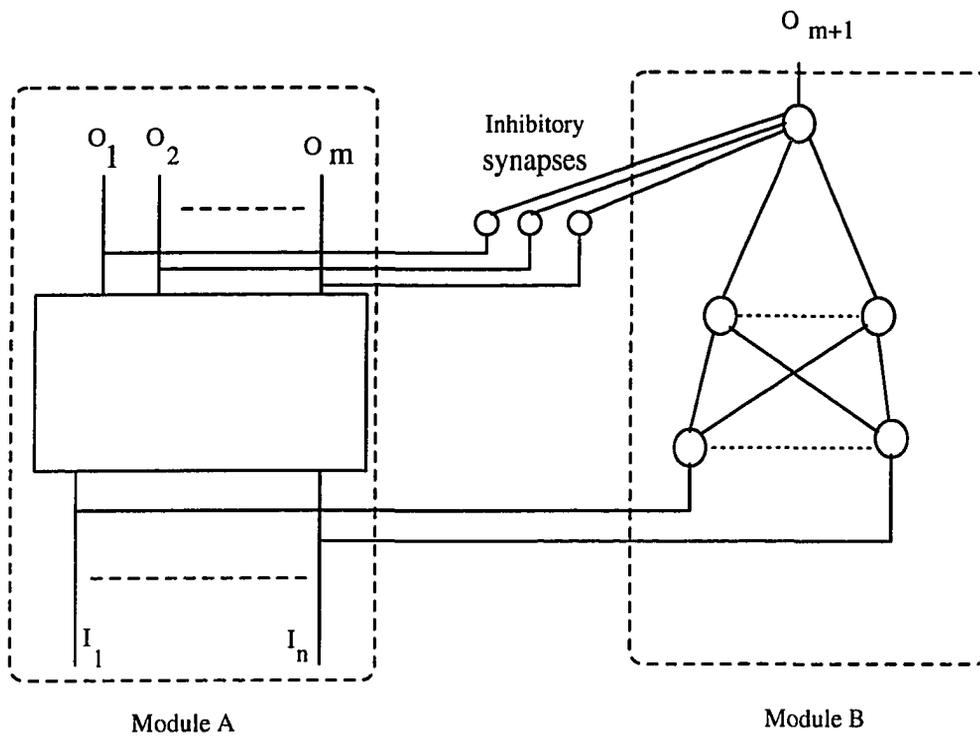


Figure 8.18: Modular network

When an input pattern belongs to one of the old classes, one of the old outputs will fire, and the negative weight will inhibit the new output from firing. Through this inhibitory synapse, the old stored information is passed to the new module and the whole network needs only to be incrementally trained for each additional class. This concept of inhibition is also found in human brains, as presented in the discussion of inhibitory interneurons in [99].

Simulation results

The proposed network and training algorithm presented were used for classification of signals from nondestructive evaluation (NDE) applications. The signals were obtained from electromagnetic and ultrasonic transducers that are used extensively in the inspection of nuclear power plant tubing. The electromagnetic transducer is an eddy current probe which is excited by alternating current and the impedance of the probe coil is recorded as the probe scans the tubing [100, 101]. Information relating to the flaw in the tubing is contained in the eddy current signal plotted on the impedance plane with resistance along the real axis and reactance along the imaginary axis. The ultrasonic transducer consists of a piezoelectric crystal used to launch an ultrasonic wave into an object under inspection. The reflected wave is received and analyzed for presence of cracks and inclusions in the tubing.

The objective of the signal classification exercise is to determine the nature of the defect (a crack on inner wall, a crack on outer wall, denting of the tube, a through-wall hole, etc.) on the basis of information contained in the signal. Additional details are given in [102]. The signals are generally processed, prior to neural network classification, for the purpose of data reduction and feature selection. Compressing the information leads to a shorter length input vector to the neural network, thereby minimizing problems of combinatorial explosion that are typically encountered when the input vector is long.

The algorithm was implemented on three different examples of data as described below:

Example 1:

The first data set used consisted of eddy current signals from four classes of

defects. The eddy current impedance plane trajectories of the four classes are shown in Figure 8.19. A set of 40 signals was generated and the data was first preprocessed using Fourier descriptor [102] approach to obtain eight descriptors that served as input to the network.

Example 2:

In the second example, 59 experimentally measured eddy current signals from seven classes of defects were considered [102]. The impedance plane trajectories were again compressed into eight Fourier descriptors and applied to the MLP network.

Example 3:

The third example consists of 905 patterns of ultrasonic waveforms from three types of reflectors: cracks, rootwelds, and counterbores [103]. These signals, consisting of 512 data points, were preprocessed using wavelet transform which provides a time-frequency distribution of the signal. Additional details of the application of wavelet transform are discussed in [104]. The wavelet transform preprocessor compresses the information in the raw data and provides an input vector of dimension 31 [105].

Figures 8.20-8.25 show a comparison between the modified steepest descent training algorithm and the regular back-propagation algorithm for training the network. The multilayer perceptron network architecture chosen consists of eight nodes in the input layer, five nodes in the hidden layer, and four nodes in the output layer. Figures 8.20 and 8.21 show the results of implementation for the first example. The vertical axis represents the error averaged over the training data, while the horizontal axis represents the iteration number.

The algorithm was implemented using various combinations of energy functions.

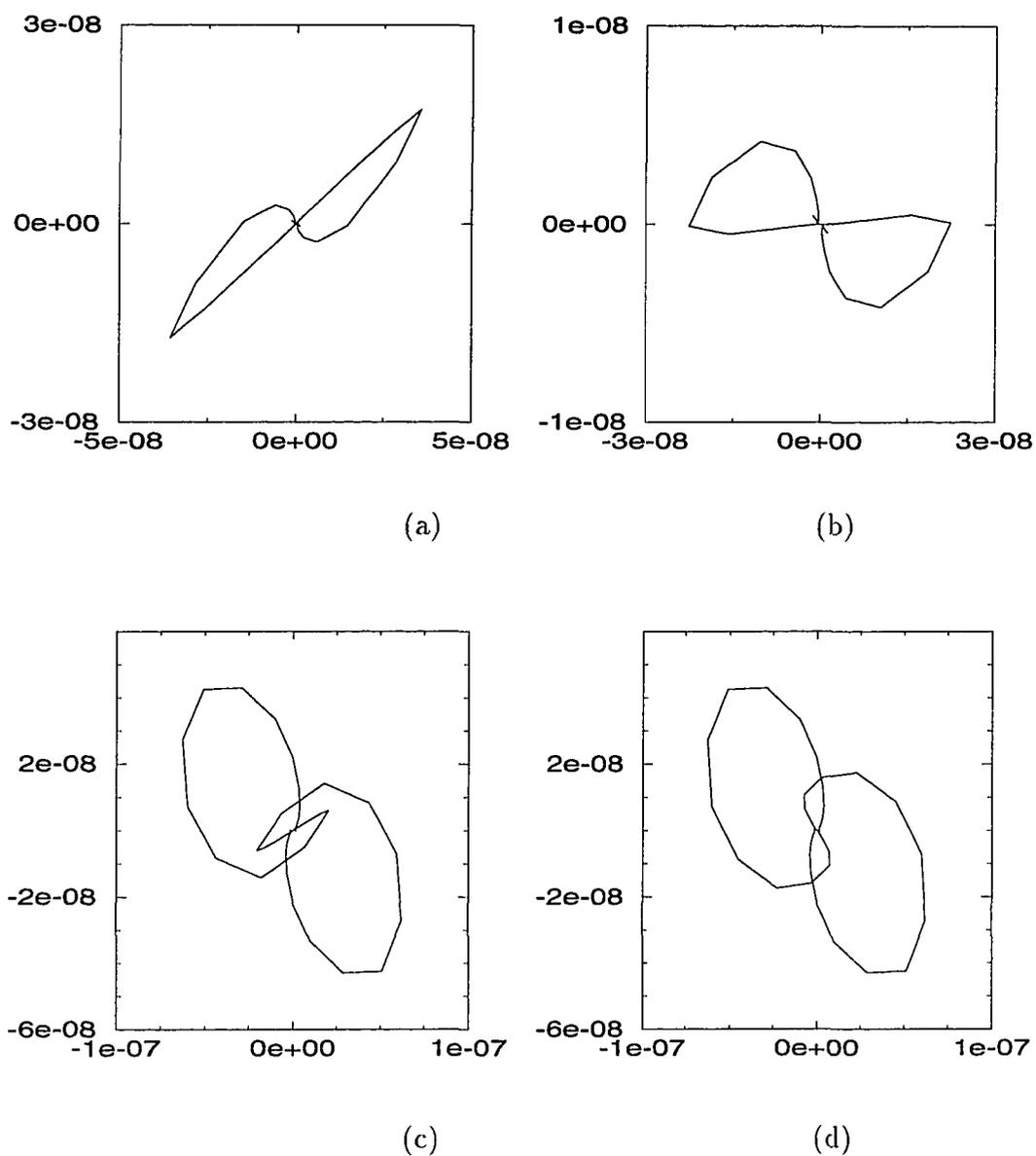


Figure 8.19: Typical eddy current trajectories for the four classes under consideration

- (a) Inner diameter (I.D.) defect
- (b) Outer diameter (O.D.) defect
- (c) I.D. defect in presence of a support plate
- (d) O.D. defect in presence in presence of a support plate

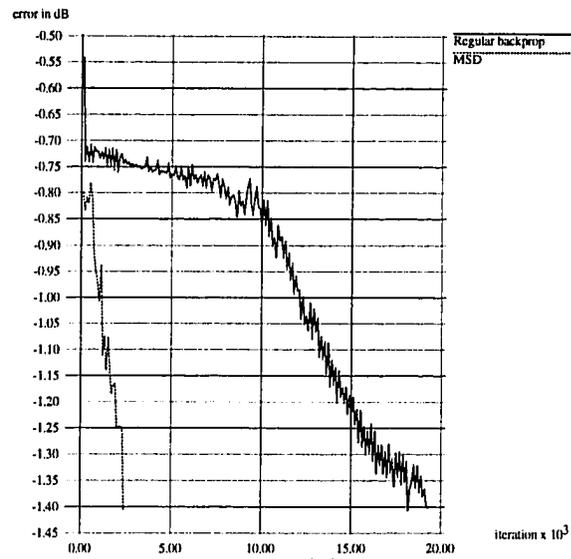
Figures 8.20.a and 8.20.b are obtained using the error function derived from a combination of a quadratic and entropy cost functions as described in equation (8.68). Figures 8.21.a and 8.21.b are obtained using the arctanh δ described in equation (8.69).

For the second example, network architecture consisted of eight input nodes, eight nodes in the hidden layer, and seven nodes in the output layer. Results of the performance of the training algorithm, using different cost functions, are presented in Figures 8.22 and 8.23.

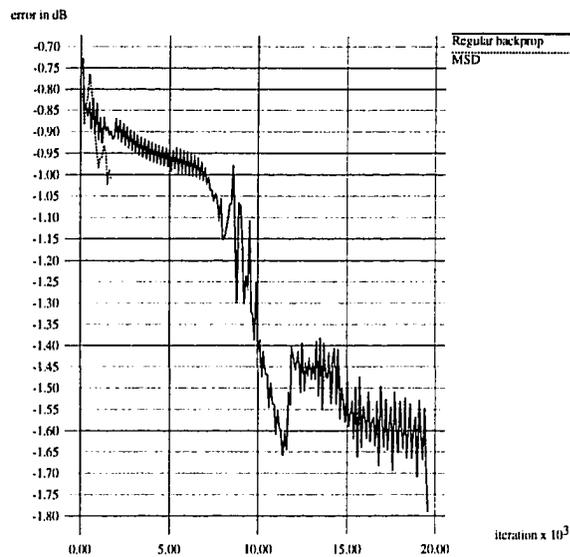
For the third example, a three layer network was chosen with 31 input nodes, 16 nodes in the hidden layer, and three output nodes. Roughly, a third of the total data was used in training the network, and the network was tested using the remaining patterns. Results of the performance of the training algorithm are shown in Figures 8.24 and 8.25. The corresponding classification performance is summarized in Tables 8.1 and 8.2.

Comparisons between the performance of the modular architecture network and a regular MLP network are shown in Figures 8.26-8.32. Figures 8.26 and 8.28 show the result for the four class problem. The modular architecture was developed by augmenting a multilayer network trained with three classes by a module to recognize the four classes.

In the second example, the modular network was developed by augmenting a network, trained with five class data to recognize two additional new classes and the results are shown in Figures 8.29-8.31. Results of the third example, where a new class is added to a network trained with two of the total three classes, are given in Figure 8.32.



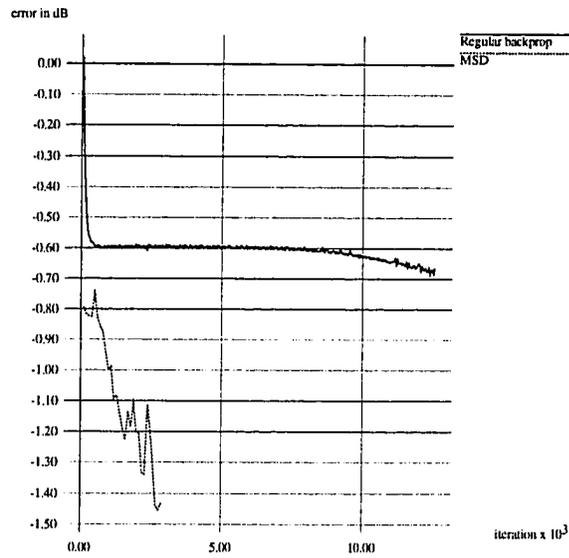
(a)



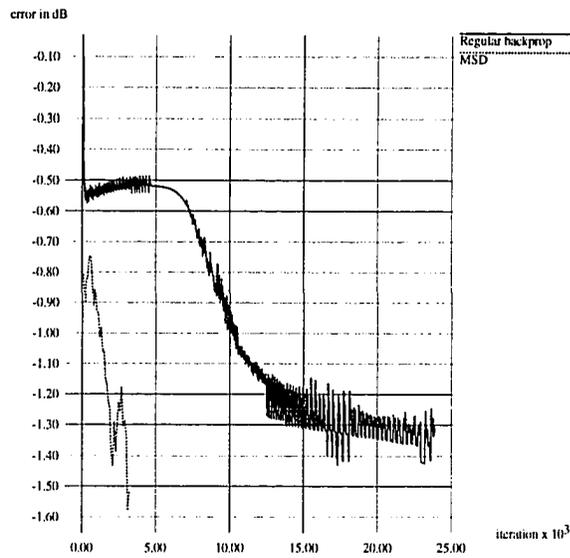
(b)

Figure 8.20: Progress of error with iteration until full recognition of patterns for regular back-propagation and MSD algorithms using error function in (8.68).

(a) $\zeta = 0.0$ (b) $\zeta = 0.2$ (4 class problem)



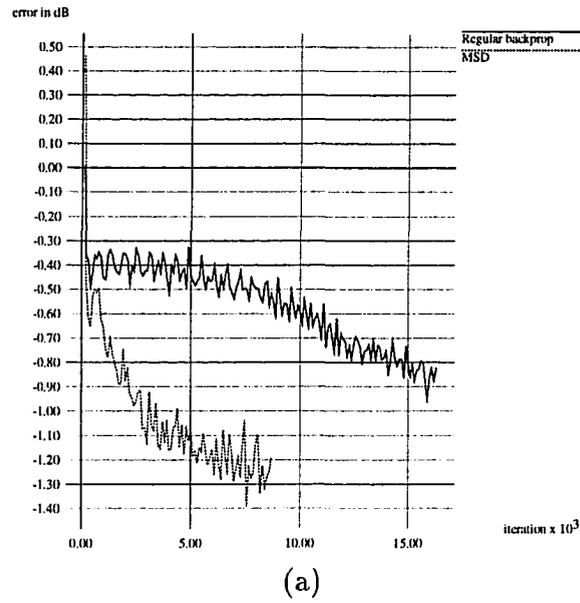
(a)



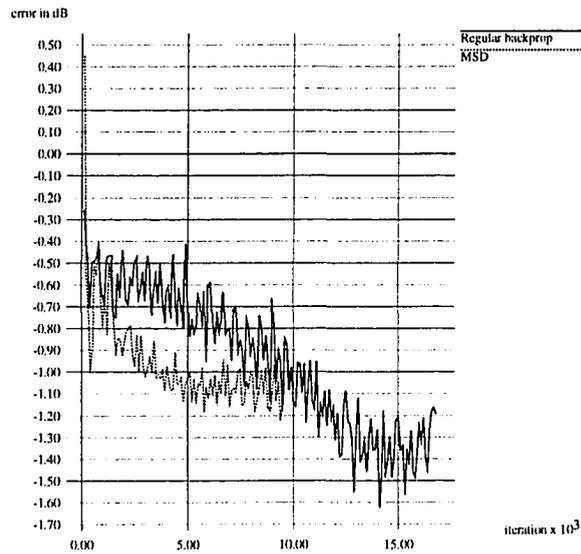
(b)

Figure 8.21: Progress of error with iteration until full recognition of patterns for regular back-propagation and MSD algorithms using error function in (8.69).

(a) $\zeta = 0$ (b) $\zeta = 0.2$ (4 class problem)



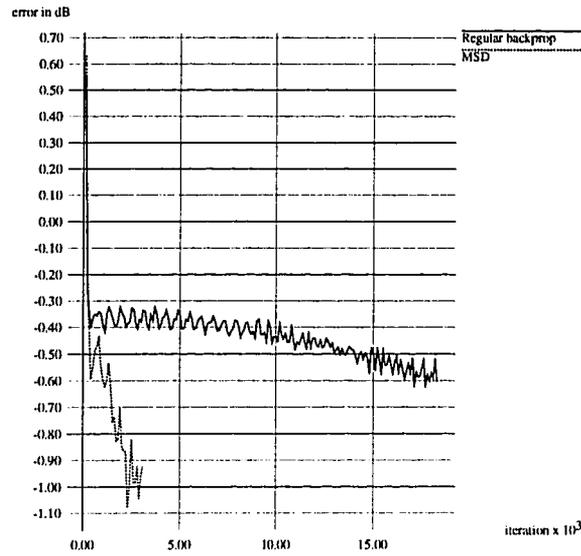
(a)



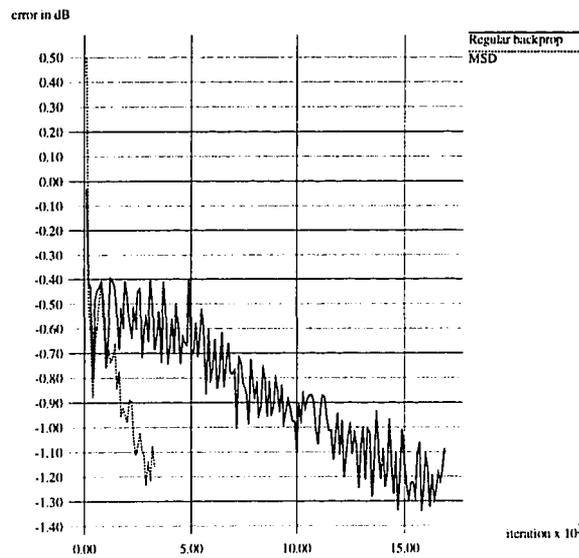
(b)

Figure 8.22: Progress of error with iteration until full recognition of patterns for regular back-propagation and MSD algorithms using error function in (8.68).

(a) $\zeta = 0.0$ (b) $\zeta = 0.2$ (7 class problem)



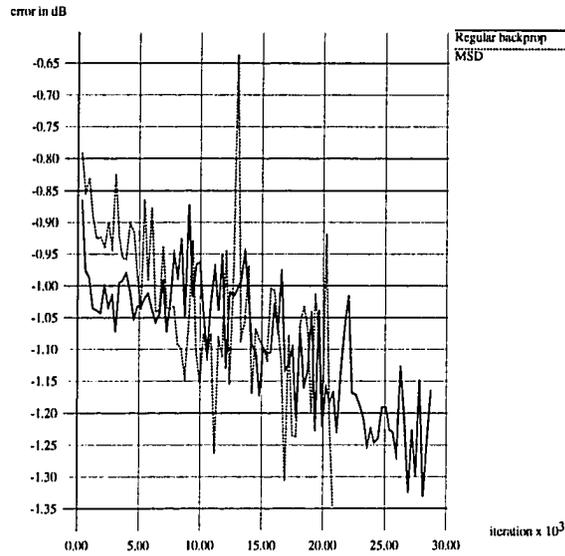
(a)



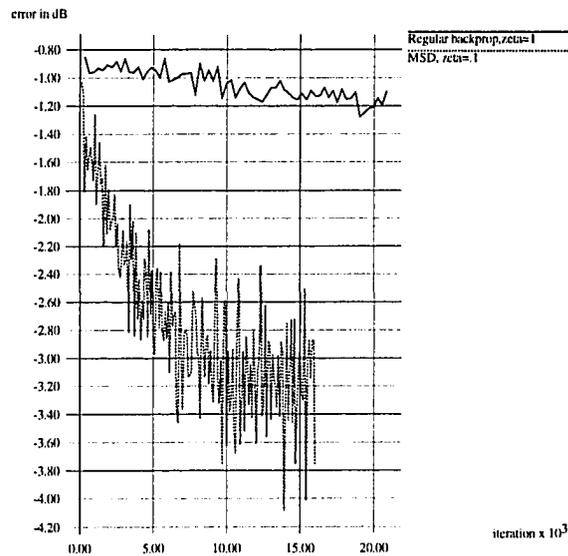
(b)

Figure 8.23: Progress of error with iteration until full recognition of patterns for regular back-propagation and MSD algorithms using error function in (8.69).

(a) $\zeta = 0$ (b) $\zeta = 0.2$ (7 class problem)



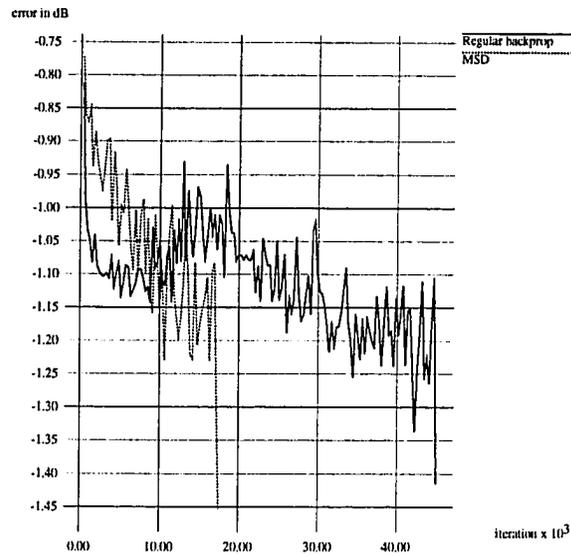
(a)



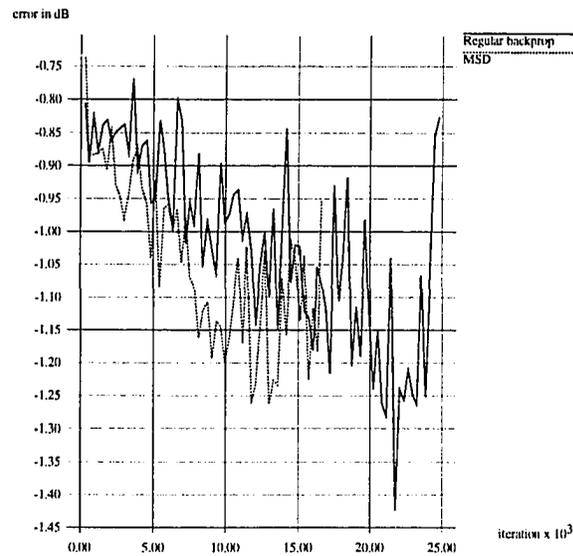
(b)

Figure 8.24: Progress of error with iteration until full recognition of patterns for regular back-propagation and MSD algorithms using error function in (8.68).

(a) $\zeta = 0$ (b) $\zeta = 1$ (3 class problem)



(a)



(b)

Figure 8.25: Progress of error with iteration until full recognition of patterns for regular back-propagation and MSD algorithms using error function in (8.69).

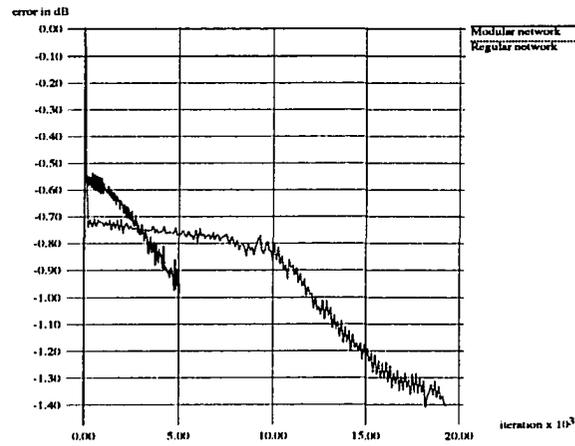
(a) $\zeta = 0.0$ (b) $\zeta = 0.3$ (3 class problem)

Table 8.1: Performance of training using regular back-propagation for various choice of error functions for Example 3

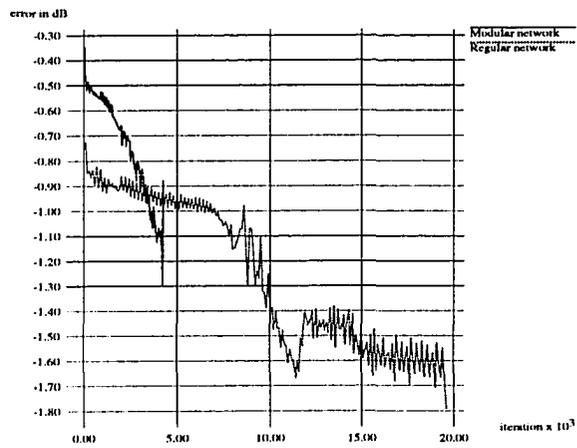
Training Algorithm	Number of iterations	Percentage of classification
Error function in (8.68) $\zeta = 0.$	28820	88.23%
Error function in (8.68) $\zeta = 0.1$	20948	88.56 %
Error function in (8.69) $\zeta = 0.0$	45117	90.21 %
Error function in (8.69) $\zeta = 0.3$	24807	89.69 %

Table 8.2: Performance of training using modified steepest descent for various choice of error functions for Example 3

Training Algorithm	Number of iterations	Percentage of classification
Error function in (8.68) $\zeta = 0.0$	20870	92.54 %
Error function (8.68) $\zeta = 0.1$	16062	90.38 %
Error function in (8.69) $\zeta = 0.0$	17577	90.54 %
Error function in (8.69) $\zeta = 0.3$	16614	90.22 %



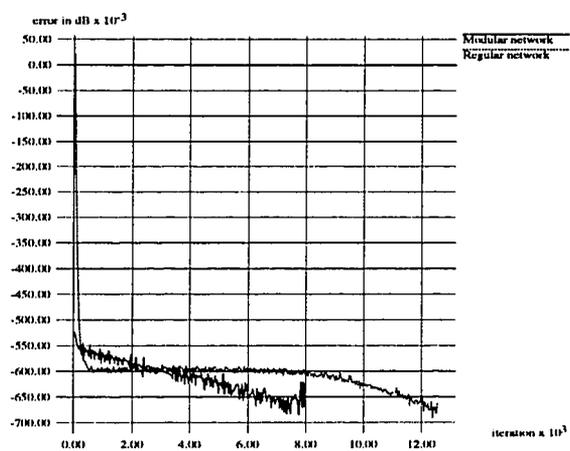
(a)



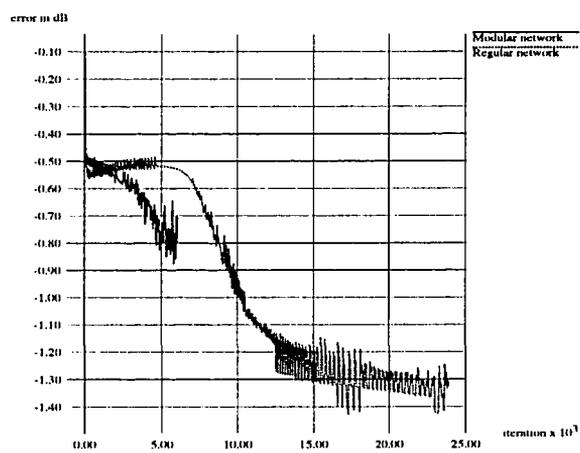
(b)

Figure 8.26: Comparing performance of a modular architecture network with a regular network for error function in (8.68).

(a) $\zeta = 0.0$ (b) $\zeta = 0.2$ (4 class problem)



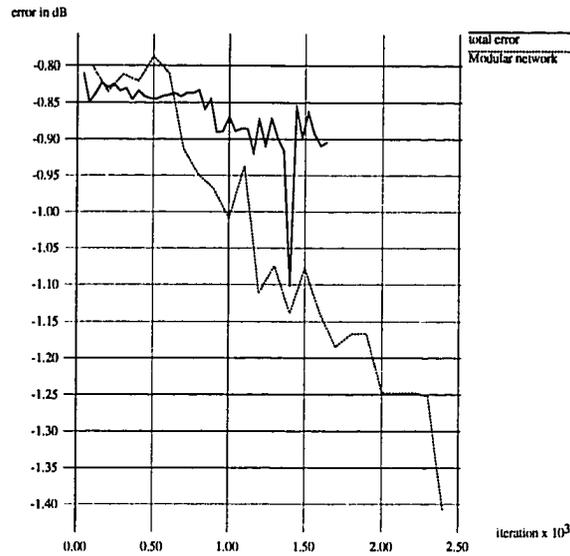
(a)



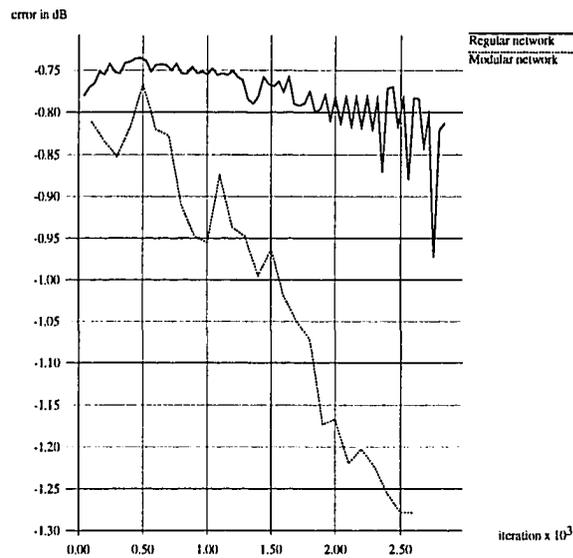
(b)

Figure 8.27: Comparing performance of a modular architecture network with a regular network for error function in (8.69).

(a) $\zeta = 0$ (b) $\zeta = 0.2$ (4 class problem)



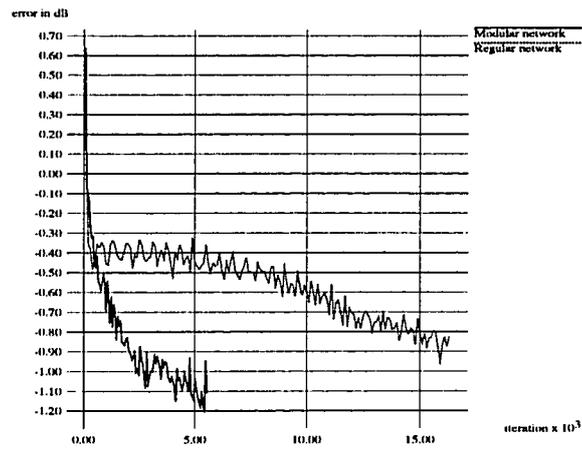
(a)



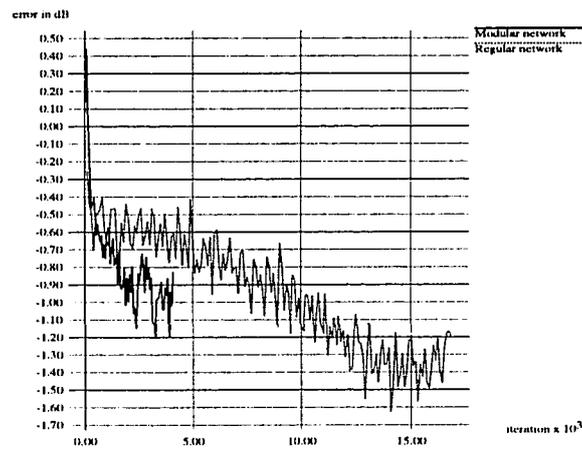
(b)

Figure 8.28: Comparing performance of a modular architecture network with a regular network using MSD algorithm.

(a) $\zeta = 0$ (b) $\zeta = 1$ (4 class problem)



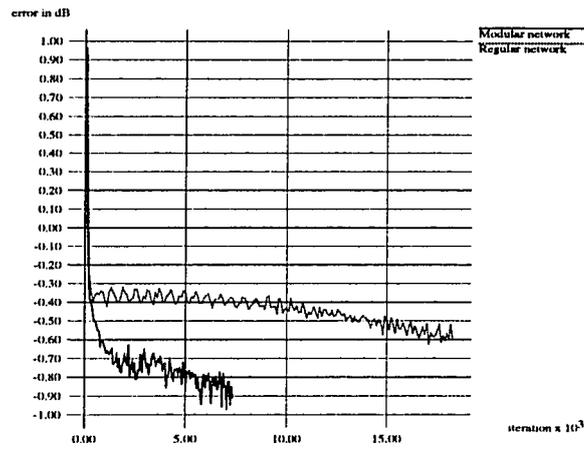
(a)



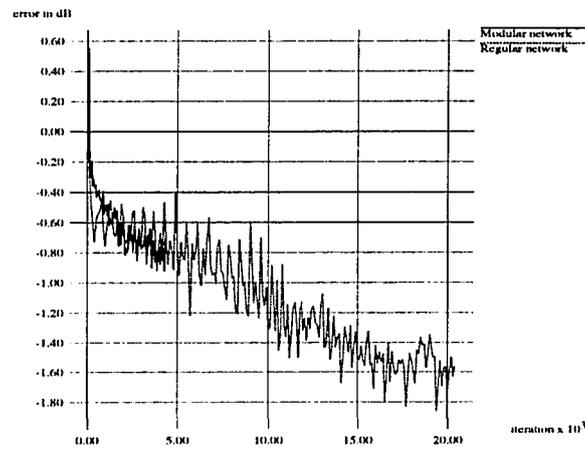
(b)

Figure 8.29: Comparing performance of a modular architecture network with a regular network using error function in (8.68).

(a) $\zeta = 0$ (b) $\zeta = 0.2$ (7 class problem)



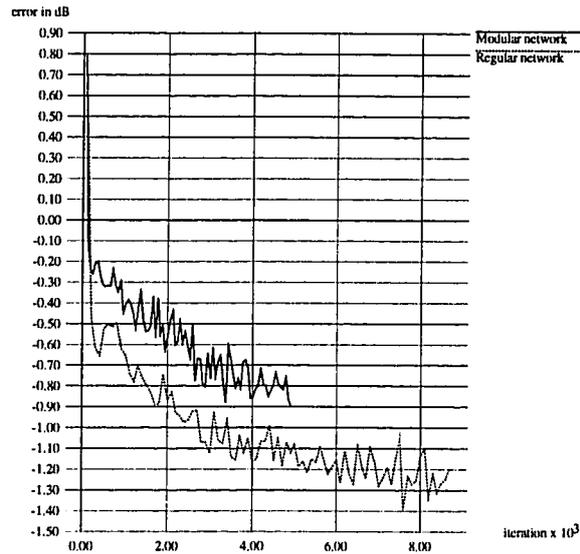
(a)



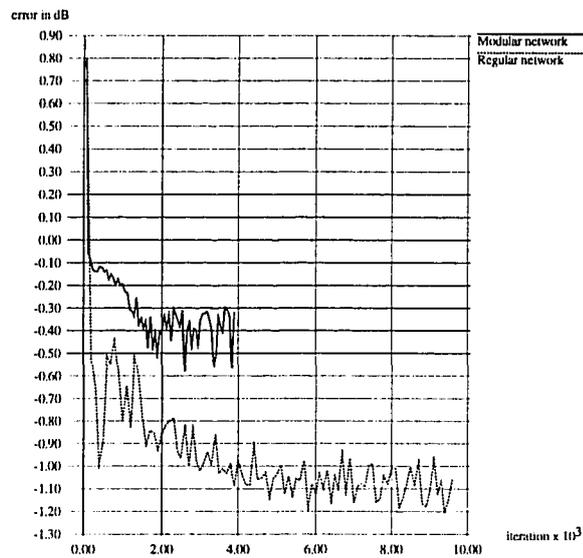
(b)

Figure 8.30: Comparing performance of a modular architecture network with a regular network using error function in (8.69).

(a) $\zeta = 0.0$ (b) $\zeta = 0.3$ (7 class problem)



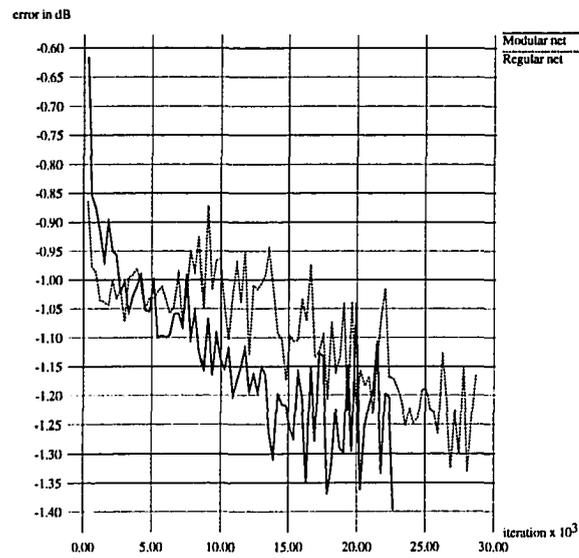
(a)



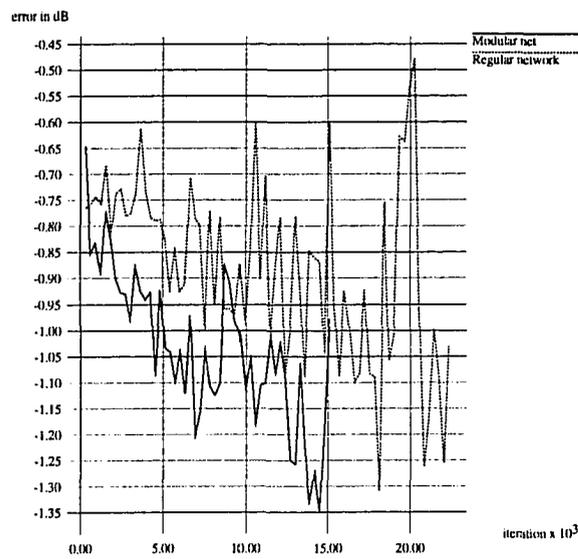
(b)

Figure 8.31: Comparing performance of a modular architecture network with a regular network using MSD algorithm.

(a) $\zeta = 0.0$ (b) $\zeta = 0.1$ (7 class problem)



(a)



(b)

Figure 8.32: Comparing performance of a modular architecture network with a regular network for error function in (8.68).

(a) $\zeta = 0$ (b) $\zeta = 0.3$ (3 class problem)

CHAPTER 9. CONCLUSIONS

This dissertation investigates various approaches for development of computational models for electromagnetic nondestructive evaluation. The techniques cover both the forward and inverse problem related to microwave and eddy current NDE phenomena. The modeling strategy depends on the problem, the computational approach chosen and the available computer power.

Two main approaches are presented for solving the forward problem: the differential equation and integral equation approaches. In the differential equation approach, the formulation is given in terms of the fields which are defined on the entire domain. In the integral equation approach, the formulation is given in terms of the sources which are of a non zero value only on the object.

The advantage of the differential approach is in the low storage requirements of its formulation. This advantage is however counterbalanced by the need for the large domain of solution as encountered in microwave NDE applications governed by the wave equation where the boundary condition is the Sommerfeld radiation condition at the infinite boundary. Appropriate techniques are presented which keep the discretization domain finite and couple this domain to the infinite space.

A hybrid finite element method for solving scattering problems is described by dividing the space into two regions. The scatterer is enclosed by a hypothetical

surface. The region interior to the surface is modeled using finite element analysis techniques to solve for the equivalent sources representing the scatterer which is then used to solve for the scattered fields in the region exterior to the scatterer.

The integral equation formulation is based on the moment method. The unknowns are restricted to a finite source domain, such as the surface or volume of a scatterer. However, the formulation leads to a linear system of equations characterized by a dense matrix. In most realistic applications the required number of discretization elements are usually high which results in large storage requirements. Efficient modeling techniques based on conjugate gradient fast Fourier transform techniques are presented for minimizing the storage requirements. These techniques also exploit the symmetry of different components of the Green's tensor.

The proposed methods are then applied to specific application problems and formulation is developed for modeling electromagnetic scattering from objects or inhomogeneities embedded in multilayered media. The model formulation is based on first decomposing the incident field into plane wave components. In the hybrid finite element method formulation, the field in the interior region is used to obtain equivalent current sources of the scatterer and the field in the exterior region is computed by the integration of plane waves propagating in the multilayered media. In the integral equation formulation, a secondary Green's function is included to account for the layered media. A variety of techniques for evaluating integrals with oscillating integrands are discussed. The model is applicable in a variety of NDE problems including the simulation of a crack or a reinforcement bar embedded in large structures. The model may also be applicable to other problems as in lunar geophysical explorations [106].

The minimization of memory requirements for differential and integral equation approaches has a significant impact in keeping the solution time within feasible limits by avoiding the need for secondary storage references. However, in many practical applications, resorting to massive parallel computing is inevitable. The dissertation briefly presents the subject of parallel computing and the development of a probability of detection model for eddy current NDT. The POD simulation scheme is done using Monte Carlo modeling to account for different sources of random variabilities that are encountered during actual testing. This demands thousands of executions of the 3D numerical model.

Finite element modeling is implemented on an NCUBE2 machine which employs a hypercube architecture. All implementations of distributing the global matrix among processors were designed to minimize the interprocessor communication. The number of simultaneous runs performed can also be optimized to obtain the maximum effective speedup. The implementation uses a finite element model with direct and iterative methods for solving the system of equations on the NCUBE. Results of direct methods showed a significant advantage over iterative methods in terms of accuracy obtained within a fixed time. Implementations involving the use of direct methods can avoid the need for disk access to store the global matrix, by choosing an adequate number of processors with enough memory. This results in a significant speedup both due to the use of parallel processing strategies as well as the avoidance of disk access.

The dissertation also investigates a new approach based on artificial neural networks for solving inverse problems. Neural networks offer the advantage of superior computational ability due to the high degree of parallelism and interconnectivity.

This ability makes neural networks attractive in many applications in engineering and science. Two class of approaches are presented. The first approach depends on solving the governing integral equation using a Hopfield type neural network. The second approach depends on training a multilayered perceptron with known input-output pairs to perform defect characterization.

In the first approach, the electromagnetic inverse problem is formulated in the form of a Fredholm integral equation. This equation is first discretized by expressing the desired solution as a sum of weighted basis functions and then the quadrature error function for the inverse problem is formed. This function consists of an error term and constraint terms. The constraints imposed in the formulation includes a smoothness function that is reduced progressively during the iterative procedure, and constraints arising due to boundary conditions are progressively emphasized. The error function obtained for the integral equation is then compared with the energy function of the neural network, thereby deriving the parameters of the Hopfield network. The limits on the parameters of amplifiers for the network to converge to the required solution are given.

An Incremental Learning Network (ILN) is suggested for building the input/output mapping for inverse problems. A new training algorithm using the modified steepest descent method and a dynamic modular structure are used to build he ILN network. The basic idea of the modified steepest descent training algorithm is to determine a transformation of the weight space so that the mapping produced by the transformation yields a better conditioned energy(cost) function. An example using the arctan function to map the weight space is presented. The training algorithm is derived and incorporated into the conventional back-propagation algorithm for training the

multilayered perceptron. The choice of the arctan function is natural because of its characteristic to saturate for large values of its argument. However, alternative forms of mathematical mappings which magnify small values and saturates large values can also be used in the transformation of the weight space.

The results of performance of the modified steepest descent algorithm shows a significant improvement over conventional back-propagation algorithm in the convergence rate and consequent reduction in training time and also in classification accuracy.

Reduction in training time can also be obtained using incremental learning algorithms in conjunction with modular architecture. The structure allows the network to learn to recognize new classes without losing the information obtained during previous learning sessions. The idea is similar to learning process in human brains where knowledge is gained by continuous accumulation of new information while retaining the old information unless it leads to a contradiction. Convergence behavior of the training algorithm using different error functions are presented illustrating the reduction of required training time using the new modular structure.

A variety of approaches and computational techniques have been used over the years in electromagnetic modeling. Although one approach might be advantageous over the others in a particular application, there is no one technique that has absolute and global superiority. Hybrid methods has always attracted attention for developing more efficient models that can be used in a wide variety of applications.

There is no doubt that the continuous growth in computing resources will have a substantial influence on electromagnetic modeling techniques, but what is still not clear is if this growth will favor one technique significantly over the others. As stated

by Harrington in the 1960s that, what was considered *fantastic modern computers speed and storage capabilities* then, led to shifting the interest from investing efforts in manipulating solutions analytically to minimize the required computational effort into applications of simple analytical methods with high computational requirements. At this stage the interest was to develop a *unifying general concept*. Harrington presented the method of moments as the large umbrella to explain other classical approaches such as eigenfunctions, Rayleigh-Ritz variational method and Galerkin method.

Today the interest should be to address new and more relevant issues. Some classical issues might practically disappear in the future, and new issues will arise. For example, requirements of storage reduction might be eased with the introduction of optical memories with two-dimensional parallel data access that will provide gigabit data storage capacities and terahertz data rates [107]. On the other hand, issues related to the choice of hardware and software structures will arise. In contrast to the past, it might be cheaper today to have an investment, developing a computing hardware that is suitable for the problem, rather than developing a software that has to account for the computer architecture. Both techniques were introduced in this dissertation. Neural networks for solving electromagnetic problems, where the choice of the network architecture and the design of its parameters to match the problem are presented. On the other hand, the POD model was designed to accommodate the model on the hypercube architecture parallel computer.

Other issues that should be addressed are related to the scalability and modularity of the model. The same model should be capable of execution on different number of processors, ranging from one processor on sequential computers to a larger number

on parallel computers. Isolating the software from the hardware is also an important requirement so that the user would be able to obtain a full use of the available computing resources without clearly including explicit directions during programming.

The scope for future research in the field of electromagnetic computation is very wide. Research needs to be done to build benchmarks for evaluating different computational techniques for various applications and computer architectures. Focus should be given to include random sources of variability to simulate the noise in practical tests. The anticipated neural computer is an issue of great attraction with its relatively inexpensive cost, modularity and scalability and the high degree of adaptability. With the emergence of advanced optical and VLSI techniques [108, 109, 110], the implementation of computational algorithms is a rapidly growing research field.

BIBLIOGRAPHY

- [1] C. A. Balanis, *Advanced Engineering Electromagnetics*, John Wiley & Sons, New York, 1989.
- [2] J. Warhus, S. Nelson, E. Johansson, J. Mast, and H. Lee, "Advanced Ground Penetrating, Imaging Radar for Bridge Inspection," *Review of Progress in Quantitative NDE*, D. O. Thompson and D. Chimenti, eds., Plenum, New York, 1994.
- [3] P. Laursen, and D. L. Atherton, "Effects of Line pressure Stress on Magnetic Flux Leakage," *British Journal of Non-Destructive Testing*, Vol. 34, No. 6, pp. 285-288, June 1992.
- [4] A. Dhar, and D. L. Atherton, "Correlation between Magnetic Flux Leakage and Magnetic Barkhausen Noise in Steel Pipelines," *British Journal of Non-Destructive Testing*, Vol. 35, No. 6, pp. 307-309, June 1993.
- [5] W. Lord, "Application of Numerical Field Modeling to Electromagnetic Methods of Nondestructive Testing," *IEEE Trans. on Magnetics*, Vol. 19, No. 6, pp. 2437-2442, November 1983.
- [6] T. Tortschanoff, "Survey of Numerical Methods in Field Calculations," *IEEE Trans. on Magnetics*, Vol. 20, No. 5, pp. 1912-1917, September 1984.
- [7] T. R. Cantor, "Review of Penetrating Radar as Applied to Nondestructive Evaluation of Concrete," *In Situ/Nondestructive Testing of Concrete*, V. M. Malhotra, ed., American Concrete Institute, Detroit, 1984.
- [8] J. Jaarinen, J. Hartikainen, and M. Luukkala, "Quantitative Thermal Wave Characterization of Coating Adhesion Defects," *Review of Progress in Quantitative NDE*, D. O. Thompson and D. Chimenti, eds., Vol. 8B, pp. 1311-1319, Plenum, New York, 1989.

- [9] F. Ansari, "Real-Time Condition Monitoring of Concrete Structures by Embedded Optical Fibers Sensors," *Nondestructive Testing of Concrete Elements and Structures proceedings of sessions sponsored by the Engineering Mechanics Division of the American Society of Civil Engineers*, F. Ansari, ed., American Society of Civil Engineers, New York, 1992.
- [10] R. M. Measures, N. D. W. Glossop, J. Lymer, M. Leblanc, J. West, S. Dubois, W. Tsaw, and R. C. Tennyson, "Damage Assessment for Composite Materials Based on Structurally Embedded Optical Fibers and Image Enhanced Backlighting," *Review of Progress in Quantitative NDE*, D. O. Thompson and D. Chimenti, eds., Vol. 8B, pp. 1457-1465, Plenum, New York, 1989.
- [11] C. O. Ruud, D. J. Snoha, M. E. Jacobs and S. D. Weedman, "Plastic Deformation, Residual Stress, and Crystalline Texture Measurements for In-Process Characterization of FCC Metal Alloys," *Review of Progress in Quantitative NDE*, D. O. Thompson and D. Chimenti, eds., Vol. 8B, pp. 1795-1801, Plenum, New York, 1989.
- [12] S. Simms, "MOI: Magneto-Optic/Eddy Current Imaging," *Materials Evaluation*, Vol. 51, pp. 529-534, 1993.
- [13] R. C. Booton, Jr., *Computational Methods for Electromagnetics and Microwaves*, John Wiley & Sons, Inc., New York, 1992.
- [14] T. L. Jordan, "A Guide to Parallel Computation and Some Cray-1 Experiences," *Parallel Computations*, Garry Rodrigue, ed., Academic Press, New York, 1982.
- [15] A. Trew and G. Wilson, eds., *Past, Present, Parallel: A Survey of Available Parallel Computer Systems*, Springer-Verlag, London, 1991.
- [16] I. Elshafiey, *Neural Network Approach for Solving Inverse Problems*, M.S. thesis, Iowa State University, Ames, 1991.
- [17] L. Udpa and W. Lord, "Diffusion, Waves, Phase and Eddy Current Imaging," *Review of Progress in Quantitative NDE*, D. O. Thompson and D. Chimenti, eds., Vol. 4A, pp. 499-506, Plenum, New York, 1985.
- [18] L. Udpa and W. Lord, "Imaging of Electromagnetic NDT Phenomena," *Review of Progress in Quantitative NDE*, D. O. Thompson and D. Chimenti, eds., Vol. 5A, pp. 465-471, Plenum, New York, 1986.
- [19] S. Nath, T. J. Rudolphi, and W. Lord, "Comparative Study of Finite-Element and Boundary-Element Analyses of the DC Potential Drop Method," *Materials Evaluation*, Vol. 51, pp. 51-55, 1992.

- [20] A. Taflove, and K. Umashankar, "Review of FD-TD Numerical Modeling of Electromagnetic Wave Scattering and Radar Cross Section," *Proc. IEEE*, Vol. 77, No. 5, pp. 682-699, 1989.
- [21] W. C. Chew, *Waves and Fields in Inhomogeneous Media*, Van Nostrand Reinhold, New York, 1990.
- [22] L. Ting, and M. Miksis, "Exact boundary conditions for scattering problems," *J. Acoust. Soc. Am.*, Vol. 80, No. 6, pp. 1825-1827, 1986.
- [23] A. Bayliss, and E. Turkel, "Radiation Boundary Conditions for Wave-Like Equations," *Communications on Pure and Applied Mathematics*, Vol. 33, pp. 707-725, 1980.
- [24] B. Engquist and A. Majda, "Absorbing Boundary Conditions for the Numerical Simulation of Waves," *Mathematics of Computation*, Vol. 31, No. 139, pp. 629-651, 1976.
- [25] G. Mur, "The Modeling of Singularities in the Finite-Difference Approximation of the Time-Domain Electromagnetic-Field Equations," *IEEE Trans. Microwave Theory and Techniques*, MTT-29, pp. 1073-1077, 1981.
- [26] J. D'Angelo, and I. D. Mayergoyz, "Finite Element Methods for the Solution of RF Radiation and Scattering," *Electromagnetics*, Vol. 10, pp. 177-199, 1990.
- [27] C. F. Lee, R. T. Shin, and J. A. Kong, "Finite Difference Method for Electromagnetic Scattering Problems," *Progress in Electromagnetics Research: Pier 4*, J. Kong, ed., Elsevier, New York, 1991.
- [28] R. W. Clough, "The Finite Element in Plane Stress Analysis," *Proc. 2nd American Society of Civil Engineers Conference on Electronic Computation*, Pittsburgh, Pa., September 1960.
- [29] D. H. Norrie and G. de Vries, *An Introduction to Finite element Analysis*, Academic Press, New York, 1978.
- [30] O. C. Zienkiewicz, *The Finite Element Method*, McGraw-Hill, London, 1977.
- [31] R. J. Collins, "Bandwidth Reduction by Automatic Renumbering," *Int. Journal for Numerical Methods in Engineering*, Vol. 6, pp. 345-356, 1973.
- [32] K. Dincer, "Implementation of a Finite Element Model for Inspecting the Steam Generators in Nuclear Power Stations," *Report in CS Dept.*, Iowa State University, Ames, 1991.

- [33] K. Sakiyama, H. Kotera, and A. Ahagon, "3-D Electromagnetic Field Mode Analysis Using Finite Element Method by Edge Element" *IEEE Trans. on Magnetics*, Vol. 26, No. 5, September 1990.
- [34] A. Chatterjee, J. M. Jin, and J. L. Volakis, "Edge-Based Finite Elements and Vector ABC's Applied to 3-D Scattering," *IEEE Trans. on Ant. and Propag.*, Vol. 41, No. 2, pp. 221-226, February 1993.
- [35] W. E. Boyse, D. R. Lynch, K. D. Paulsen, and G. N. Minerbo, "Nodal-Based Finite-Element Modeling of Maxwell's Equations," *IEEE Trans. on Ant. and Propag.*, Vol. 40, No. 6, pp. 642-651, June 1992.
- [36] K. K. Mei, "Unimoment Method for Solving Antenna and Scattering Problems," *IEEE Trans. Ant. Propag.*, Vol. 22, pp. 760-766, 1974.
- [37] A.C. Cangellaris, R. Lee, "The Bymoment Method for Two-Dimensional Electromagnetic Scattering," *IEEE Trans. Ant. Propag.*, Vol. 38, No. 9, pp. 1429-1437, 1990.
- [38] M.A. Morgan, C.H. Chen, S.C. Hill, and P.W. Barber, "Finite Element-Boundary Integral Formulation for Electromagnetic Scattering" *Wave Motion*, Vol. 6, pp. 91-103, 1984.
- [39] D. Givoli, *Numerical methods for problems in infinite domains*, Elsevier, Amsterdam, 1992.
- [40] D. K. Gartling and E. B. Becker, "Finite Element Analysis of Viscous Incompressible Fluid Flow," *Comp. Meth. Appl. Mech. Engineering*, Vol. 8, pp. 51-60, 1976.
- [41] P. Bettess and O. C. Zienkiewicz, "Diffraction and Refraction of Surface Waves Using Finite and Infinite Elements," *Int. Journal for Numerical Methods in Engineering*, Vol. 11, pp. 1271-1290, 1977.
- [42] G. Beer and J. L. Meek, "Infinite Domain' Elements," *Int. Journal for Numerical Methods in Engineering*, Vol. 17, pp. 43-52, 1981.
- [43] M. J. McDougall and J. P. Webb, "Infinite Elements for the Analysis of Open Dielectric Waveguides," *IEEE Trans. Microwave Theory and Techniques*, Vol. 37, No. 11, pp. 1724-1731, November 1989.
- [44] K. Hayata, M. Eguchi and M. Koshiba, "Self-Consistent Finite/Infinite Element Scheme for Unbounded Guided Wave Problems," *IEEE Trans. Microwave Theory and Techniques*, Vol. 36, No. 3, pp. 614-616, March 1988.

- [45] R.F. Harrington, *Field Computation by Moment Methods*, Macmillan, New York, 1968.
- [46] R. Harrington, *Time-Harmonic Electromagnetic Fields*, McGraw-Hill, New York, 1961.
- [47] A. Ishimaru, *Electromagnetic wave propagation, radiation, and scattering*, Prentice-Hall, Englewood Cliffs, N.J., 1991.
- [48] J. J. H. Wang, *Generalized Moment Methods in Electromagnetics: Formulation and Computer Solution of Integral Equations*, John Wiley & Sons, New York, 1991.
- [49] O. D. Kellogg, *Foundations of potential theory ... with 30 figures*, J. Springer, Berlin, 1929.
- [50] M. Catedra, E. Gago, and L. Nuno, "A Numerical Scheme to Obtain the RCS of Three-Dimensional Bodies of Resonant Size Using the Conjugate Gradient Methods and the Fast Fourier Transform," *IEEE Trans. Ant. Propag.*, Vol. 37, No. 5, pp. 528-537, 1989.
- [51] J. Van Bladel, *Electromagnetic Fields*, McGraw-Hill, New York, 1964.
- [52] G. Hohmann, "Three-Dimensional Induced Polarization and Electromagnetic Modeling," *Geophysics*, Vol. 40, No. 2, pp. 309-324, 1975.
- [53] S. Lee, J. Boersma, C. Law, and G. Deschamps, "Singularity in Green's Function and its Numerical Evaluation," *IEEE Trans. Ant. Propag.*, Vol. 28, pp. 311-317, 1980.
- [54] A. F. Peterson, "Absorbing Boundary Conditions for the Vector Wave Equation," *Microwave and Optical Technology Letter*, Vol. 1, No. 2, pp. 62-64, April 1988.
- [55] J. P. Webb and V. N. Kanellopoulos, "Absorbing Boundary Conditions for the Finite Element Solution of the Vector Wave Equation," *Microwave and Optical Technology Letter*, Vol. 2, No. 10, pp. 370-372, October 1989.
- [56] S. Chang, K. Mei, "Application of the Unimoment Method to Electromagnetic Scattering of Dielectric Cylinders," *IEEE Trans. Ant. Propag.*, Vol. 24, pp. 35-42, 1976.
- [57] J. Wait, *Electromagnetic Waves in Stratified Media*, Macmillan, New York, 1962.

- [58] J. D. Murray, *Asymptotic Analysis*, Clarendon Press, Oxford, 1974.
- [59] R. Lee and A. C. Cangellaris, "Scattering from an Arbitrary Cylinder in the Presence of a Parallel Planar Media Interface Using the Bymoment Method," *Progress in Electromagnetic Research: Pier 4*, J. Kong, ed., York Elsevier, New York, 1991.
- [60] M. Kunt, *Digital Signal Processing*, Artech House, Norwood, MA, 1986.
- [61] T. K. Sarkar, "On Solution of Large Systems of Linear Equations," *Application of the Method of Moments to Electromagnetic Fields*, B. J. Strait, ed., SCEE Press, St. Cloud, Florida, 1980.
- [62] J. Chao, *A Boundary Integral Equation Approach to Three Dimensional Electromagnetic Wave Scattering Problems*, Ph.D. dissertation, Iowa State University, Ames, 1994.
- [63] W. Anderson, "Computer Program: Numerical Integration of related Hankel Transforms of Orders 0 and 1 by Adaptive Digital Filtering," *Geophysics*, Vol. 44, No. 7, pp. 1287-1305, 1979.
- [64] W. Johnson, "Real Axis Integration of Sommerfeld Integrals Source and Observation Points in Air," *Radio Science*, Vol. 18, pp. 175-186, 1983.
- [65] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, *Numerical Recipes in C* Cambridge University Press, New York, 1992.
- [66] T. G. Lewis, H. El-Rewini, *Introduction to Parallel Computing*, Prentice-Hall, Englewood Cliffs, N.J., 1992.
- [67] J. Gustafson, "The Consequences of Fixed Time Performance Measurement," *Proceedings of the Twenty-Fifth Hawaii Int. Conference on System Sciences*, Vol. 3, pp. 113-124, January 1992.
- [68] J. Gustafson, G. Montry, and R. Benner, Development of Parallel Methods for a 1024-Processor Hypercube, *SIAM Journal on Scientific and Statistical Computing*, Vol. 9, pp. 532-533, 1988.
- [69] M.T. Heath, ed., *Hypercube Multiprocessors*, SIAM, Philadelphia, 1986.
- [70] S. N. Rajesh, "Probability of Detection Models for Eddy Current NDE Methods," M.S. thesis, Iowa State University, Ames, 1992.

- [71] A. Amin, S. Mohanachandran, and S. Martin, "The Incomplete Cholesky Conjugate Gradient Method on a Vector Hypercube," *The Fourth Conference on Hypercubes, Concurrent Computers, and Applications*, pp. 631-634, 1989.
- [72] M. Mu and J.R. Rice, "LU Factorization and Elimination for Sparse Matrices on Hypercube," *The Fourth Conference on Hypercubes, Concurrent Computers, and Applications*, pp. 681-684, 1989.
- [73] M. O'Sullivan and J. Rodriguez, "Storage Issues for the Finite Element Method in Parallel Processing," *The Fourth Conference on Hypercubes, Concurrent Computers, and Applications*, pp. 1357-1361, 1989.
- [74] D. S. Kershaw, "The Incomplete Cholesky-Conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations," *Journal of Computational Physics*, Vol. 26, pp. 43-65, 1978.
- [75] R. P. Lippman, "Introduction to Computing with Neural Networks." *IEEE ASSP Magazine*, Vol. 4, pp. 4-21, 1987.
- [76] J. Baumeister, *Stable Solution of Inverse Problems*. Friedr. Vieweg & Sohn, Braunschweig, 1987.
- [77] A. N. Tihonov, "Solution of Incorrectly Formulated Problems and the Regularization Method," *Soviet Mathematics*, Vol. 4, pp. 1035-1038, 1963.
- [78] L. M. Delves and J. Walsh, eds., *Numerical Solution of Integral Equations*, Clarendon Press, Oxford, 1974.
- [79] J. H. McWhirter, R. J. Duffin, P. J. Brehm and J. J. Oravec, "Computational Methods for Solving Static Field and Eddy Current Problems via Fredholm Integral Equations," *IEEE Trans. on Magnetics*, Vol. 15, No. 3, pp. 1075-1084, May 1979.
- [80] W. V. Lovitt, *Linear Integral Equations*, McGraw-Hil, New York, 1924.
- [81] C. T. H. Baker, *The Numerical Treatment of Integral Equations*, Clarendon Press, Oxford, 1977.
- [82] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley, Reading, Mass., 1974.
- [83] D. L. Phillips, "A Technique for the Numerical Solution of Certain Integral Equations of the First Kind," *J. Assoc. Comput. Mach.*, Vol. 9, pp. 84-97, 1962.

- [84] T. M. Habashy, and R. Mittra, "On Some Inverse Methods in Electromagnetics," *Journal of Electromagnetic waves and applications*, Vol. 1, No. 1, pp. 25-58, 1987.
- [85] J. A. Kong, *Electromagnetic Wave Theory*, John Wiley & Sons, New York, 1990.
- [86] H. A. Sabbagh and L. D. Sabbagh, "An Eddy-Current Model for Three-Dimensional Inversion," *IEEE Trans. on Magnetics*, Vol. 22, No. 4, pp. 282-291, July 1986.
- [87] J. J. Hopfield, "Neurons with Graded Response Have Collective Computational Properties like Those of Two-State Neurons," *Proc. Natl. Acad. Sci. USA*, Vol. 81, pp. 3088-3092, May 1984.
- [88] F.M.A. Salam, Y. Wang, and M. Choi, "On the Analysis of Dynamic Feedback Neural Nets," *IEEE Trans. on Circuits and Systems*, Vol. 38, No. 2, pp. 196-201, February 1991.
- [89] D. W. Tank, and J. J. Hopfield, "Simple 'Neural' Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit," *IEEE Trans. on Circuits and Systems* CAS-33, No. 5, pp. 533-541, May 1986.
- [90] W. C. Chew, "Response of a Current Loop Antenna in an Invaded Borehole," *Geophysics*, Vol. 49, No. 1, pp. 81-91, January 1984.
- [91] T. M. Habashy, W. C. Chew and E. Y. Chow, "Simultaneous Reconstruction of Permittivity and Conductivity Profiles in a Radially Inhomogeneous Slab," *Radio Science*, Vol. 21, No. 4, pp. 635-645, July 1986.
- [92] S. Coen, K. M. Kenneth, and D. J. Angelakos, "Inverse Scattering Technique Applied to Remote Sensing of Layered Media," *IEEE Trans. Ant. Propag.*, Vol. 29, No. 2, pp. 298-306, March 1981.
- [93] C. R. Vogel, "Numerical Solution of a Non-linear, Ill-posed Problem Arising in Inverse Scattering," *Inverse Problems*, Vol. 1, pp. 393-403, 1985.
- [94] B. R. Hunt, "The Application of Constrained Least Squares Estimation to Image Restoration by Digital Computers," *IEEE Trans. Comput.*, C-22, No. 9, pp. 805-812, September 1973.
- [95] D. W. Marquardt, "Solution of Nonlinear Chemical Engineering Models," *Chemical Engineering Progress*, Vol. 55, No. 6, pp. 65-70, June 1959.

- [96] D. C. Plant, S. J. Nowlan, and G. E. Hinton, "Experiments on Learning by Back Propagation," *Report CMU-CS-86-126, Computer Science Department Carnegie-Mellon University, Pittsburgh*, June 1986.
- [97] D. Rumelhart, J. McClelland, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, The MIT Press, Cambridge, MA, 1986.
- [98] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, 1991.
- [99] M. A. Gluck, D. E. Rumelhart, eds., *Neuroscience and Connectionist Theory*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1990.
- [100] L. Udpa and W. Lord, "A Discussion of the Inverse Problem in Electromagnetic Nondestructive Testing," *Review of Progress in Quantitative NDE*, D. O. Thompson and D. Chimenti, eds., Vol. 5A, pp. 375-382, Plenum, New York, 1986.
- [101] L. Udpa, W. Lord, and S. S. Udpa, "Frequency Domain Methods for the Analysis of Multifrequency Eddy Current Data," *Review of Progress in Quantitative NDE*, D. O. Thompson and D. Chimenti, eds., Vol. 8A, pp. 329-336, Plenum, New York, 1989.
- [102] S. R. Satish, *Parameteric Signal Processing for Eddy Current NDT*, PH.D. Dissertation, Colorado State University, Fort collins, 1983.
- [103] RF Waveform Database for BWR IGSCC Ultrasonic Examination, Revision 0, *EPRI Report*, 1989.
- [104] Y. Sheng, D. Roberge, and H. Szu, "Optical Wavelet Transform," *Optical Engineering*, Vol. 31, No. 9, September 1992.
- [105] J. Chao, L. Udpa and S.S. Udpa, "Ultrasonic Signal Analysis Using Wavelet Transfrom," *Review of Progress in Quantitative NDE*, D. O. Thompson and D. E. Chimenti, eds., Plenum Press, New York, 1993.
- [106] Z. Kopal and D. Strangway, eds., *Lunar Geophysics: Proceedings of a Conference at the Lunar Science Institute*, Houston, Texas, D. Reidel Publishing Co., Dordrecht-Holland, 1971.
- [107] A.S. Dvornikov, S. Esener, and P.M. Rentzepis, "Three-Dimensional Optical Storage Memory by Means of Two-Photon Interaction," *Optical Computing Hardware*, J. Jahns and S. Lee, eds., Academic Press, Inc., Boston, 1994.

- [108] D. Psaltis, and N. Farhat, "Optical Information Processing Based on an Associative-memory Model of Neural Nets with Thresholding and Feedback," *Optics Letters*, Vol. 10, No. 2, pp. 98-100, February 1985.
- [109] B. W. Lee, and B. J. Sheu, *Hardware Annealing in Analog VLSI Neurocomputing*. Kluwer Academic Publishers, Boston, 1991.
- [110] T. J. Cloonan, "Architectural Considerations for Optical Computing and Photonic Switching," *Optical Computing Hardware*, J. Jahns and S. Lee, eds., Academic Press, Inc., Boston, 1994.

ACKNOWLEDGMENTS

Praise be to God, The Cherisher and Sustainer of the Worlds; Most Gracious, Most Merciful; Master of the Day of Judgment. Thee do we worship, and Thine aid we seek.

I would like to express my sincere gratitude for my major professor Dr. Lalita Udpa for her continuous guidance and encouragement during my graduate study.

I wish also to thank Dr. Satish Udpa, Dr. William Lord, Dr. Mark Lusk, Dr. Frank Rizzo and Dr. Suraj Kothari for their valuable advice and comments as members of my graduate committee.

I am also appreciative to Dr. Joseph Chao for valuable discussions in electromagnetism, and to Ms. Linda Clifford for proofreading this dissertation.

The financial support provided by the Electrical Power Research Institute and by the FAA Center for Aviation Systems Reliability operated by Iowa State University and supported by the Federal Aviation Administration under grant No. 93-G-018 is gratefully acknowledged.

Finally, I would like to thank my parents for their invaluable encouragement and support throughout my education career, and all my family for being always with me while they were thousands of miles away.