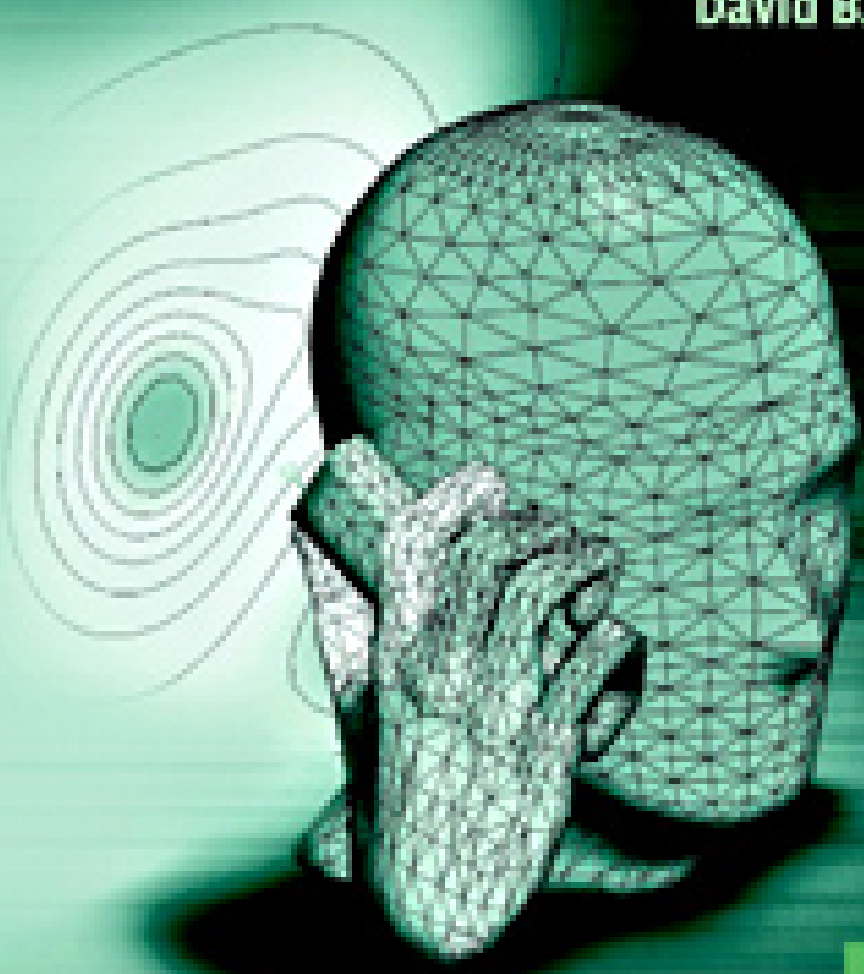
Computational Electromagnetics for RF and Microwave Engineering

David B. Davidson



CAMBRIDGE

CAMBRIDGE www.cambridge.org/9780521838597

This page intentionally left blank

COMPUTATIONAL ELECTROMAGNETICS FOR RF AND MICROWAVE ENGINEERING

The numerical approximation of Maxwell's equations, computational electromagnetics (CEM), has emerged as a crucial enabling technology for radio-frequency, microwave, and wireless engineering. The three most popular "full-wave" methods – the Finite Difference Time Domain method, the Method of Moments, and the Finite Element Method – are introduced in this book by way of one- or two-dimensional problems. Commercial or public domain codes implementing these methods are then applied to complex, real-world engineering problems, and a careful analysis of the reliability of the results obtained is performed, along with a discussion of the many pitfalls that can result in inaccurate and misleading solutions. The book will empower readers to become discerning users of CEM software, with an understanding of the underlying methods, and confidence in the results obtained. It also introduces readers to the art of code development. This book has a dedicated website making available a number of MATLAB scripts, implementing much of the theory discussed, and including additional material on the practical applications of CEM. Suitable for senior undergraduate and graduate students taking courses on CEM, this would also be a valuable reference book for practicing engineers in the industry.

DAVID DAVIDSON received his Ph.D. in 1991 from the University of Stellenbosch in South Africa where he is currently a professor in the Department of Electrical and Electronic Engineering. He has also been a visiting scholar at the University of Arizona; a visiting fellow at Trinity College, Cambridge; and was a guest professor at the International Research Centre for Telecommunications-Transmission and Radar at the Delft University of Technology in the Netherlands. He has authored or co-authored 38 papers in refereed journals and co-edits the "EM Programmer's Notebook" column in the *IEEE Antennas and Propagation Magazine*. David is a member of the IEEE, ACES, and SAIEE, and served as chairman of the south African IEEE AP/MTT Chapter from 1996 to 1998. He also served as the national convener of the NRF's Engineering Assessment Committee in 2002 and is the Principal Grantholder of the HF Electromagnetic and Electronic Engineering research consortium, funded by the NRF.

COMPUTATIONAL ELECTROMAGNETICS FOR RF AND MICROWAVE ENGINEERING

DAVID B. DAVIDSON
Department of Electrical and Electronic Engineering
University of Stellenbosch, South Africa

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo

Cambridge University Press
The Edinburgh Building, Cambridge CB2 8RU, UK
Published in the United States of America by Cambridge University Press, New York
www.cambridge.org
Information on this title: www.cambridge.org/9780521838597

First published in print format 2005

ISBN-13 978-0-511-41425-1 eBook (EBL)

ISBN-13 978-0-521-83859-7 hardback

ISBN-13 978-0-521-07012-6 paperback

To Amor.

Contents

Preface

On graduating twenty years back, in 1984, my first job was as a research engineer working on computational electromagnetics (CEM) at the National Institute for Aeronautical Systems Technology (as it was then called) of the Council for Scientific and Industrial Research (CSIR) in Pretoria, South Africa. It was an exciting time to be working in this field. Although a number of methods had already been successfully introduced, including the three which will be discussed in detail in this book, major advances were being made in all of these methods, and the power of desktop computers was growing in leaps and bounds. No commercial programs (or codes, as they are generally called) were then available for RF problems, but some US government-sponsored codes, in particular the NEC-2 code, were becoming available for general use.

The 1980s saw the final decade of the Cold War, which in some areas (such as Southern Africa) was far from cold. New military technologies, in particular stealth, were driving CEM to address progressively more electromagnetically complex problems. However, when the Cold War ended, far from CEM work coming to a halt, new commercial markets, such as the rapidly developing market in mobile telephony and personal communication systems, and the proliferation of electronic systems in motor vehicles, continued to drive the technology forward at breakneck speed throughout the 1990s. This was also due to the widespread availability of cheap and progressively more powerful personal computers as a crucial enabling technology.

CEM has now reached a modicum of maturity, with a number of powerful methods available, able to solve problems of real engineering interest at radio frequencies, and with a number of commercial codes available. This has brought a significant change in the profile of CEM practitioners, which has not been fully appreciated in the community at the time of writing. In addition to the traditional group of CEM users – largely academics, post-graduate students and research engineers at large corporations or research establishments – an entirely new generation of

users has arisen. Their interest is typically in using an existing commercial packet to solve a particular problem as rapidly as possible. They may well not have any post-graduate exposure to CEM methods, and questions which may appear elementary to CEM researchers (such as which technique is most appropriate for the problem at hand) are actually far from obvious to the beginner in the field; furthermore, marketing can "hype" a particular implementation/technique to the point where it appears omnipotent. Commercial codes aside, even academic papers are not free of such bias.

This book aims to serve the interest of both "traditional" CEM users, primarily academics, researchers and research students, and also this new non-specialist user community in industry. The book aims to fill the gap between traditional undergraduate textbooks, which generally have at most a very cursory discussion of numerical methods; antenna texts, which concentrate only on the analysis of antennas using the methods; and the specialist books on each method which are frequently formidable reading for students, or unnecessarily detailed for engineers whose primary interest is in *using* the powerful CEM codes now available. In this book, the computational methods will generally be introduced using simple one-dimensional or two-dimensional examples, so that the core of the method can be appreciated without being overwhelmed by the problems of handling complex three-dimensional geometries. Following this, the extensions required to deal with the real three-dimensional world of RF engineering are outlined, so that one gains an appreciation for the operation of complex codes. Such is the complexity of general-purpose three-dimensional CEM codes that realistic applications cannot be undertaken with anything a post-graduate student can realistically be expected to develop during a typical course, and product cycles are too short in industry to make the development of general-purpose three-dimensional codes feasible, given that off-the-shelf codes are now available.

Research students will find some features not often described in other books in this field, such as how to go about debugging and verifying a CEM code. Industrial users should find the discussions of the strengths and weaknesses of each method, as well as frequent modelling hints, comprehensive discussions of typical modelling errors, and the necessity of careful evaluation and verification of results, of great interest and utility. In short, the book discusses not only the *science* of CEM modelling, which can be gleaned from (much) reading, but also the *art* of developing and verifying reliable codes and computing reliable data, which is a skill generally derived from (sometimes bitter!) experience.

This book concentrates on the "big three" techniques in CEM – the Finite Difference Time Domain (FDTD) method, the Method of Moments (MoM) and the Finite Element Method (FEM). It was decided to focus on these three methods, since they are the most widely used in the field and all have been implemented

in successful commercial codes; some other methods are very briefly discussed so that readers are at least aware of them, but this book makes no pretense of addressing these other methods in any detail. Furthermore, the discussion in this book is focussed exclusively on applications in RF engineering. Methods such as the FEM have been used with great success for magnetostatic problems, such as motor design, but this will not be discussed here at all. A feature not often found in other books at this level is a discussion of stratified media, using the Sommerfeld potentials. Although a theoretically advanced topic, the widespread use of integrated antennas, especially microstrip, has made an appreciation of at least the basics of this approach very important. Finally, the book does not pretend to be a comprehensive text on electromagnetic theory, high-frequency circuit theory, or antenna theory and design. There are a number of superb books addressing these topics and this book is designed to complement, not compete, with them. Frequent references are made to suitable books.

Readers will also note that the level of the material becomes increasingly sophisticated as the book progresses. This is by design. The FDTD method is the only method where one can realistically hope to develop useful code oneself in a reasonable timeframe, so the discussion of this method is rather more "nuts and bolts" than for the MoM or FEM. CEM methods can also be approached as essentially an exercise in applied mathematics; although interesting theoretical insights can be thus gained, it is the author's experience that engineers do not readily take to this approach, certainly not for their initial introduction to the methods, so the introductory discussions of at least the FDTD method and MoM draw mainly on engineering physics, rather than applied mathematics. Some of the more theoretical approaches to CEM are introduced towards the end of the book, in the chapters on the MoM and FEM. (Perhaps because of the enormous amount of work on the FEM in applied mechanics, this is probably the method with the most well-developed mathematical background.) These include some elementary concepts from functional analysis, with the associated concepts of inner products and weighted residuals, as well as a brief mention of differential forms. A difficult decision was how much of the great volume of recent advances to reflect in the book. Topics such as the fast multipole method have revitalized the MoM in particular, and cannot be ignored, but the treatment of this and some other "research frontier" material is of necessity cursory.

A highly problematic issue was the selection of which commercial CEM codes to use to illustrate complex real-world implementations. One factor influencing this was the availability of a no-cost limited feature version of the software, as in the case of the MoM code FEKO; however, the FDTD and FEM codes discussed are unfortunately *not* available in such a format. The discussion tries to highlight generic features which a code should offer, and how users can exploit these.

User-manual style descriptions of how to use particular codes have been avoided as far as possible, so that discussions of one particular code should extend to other commercial codes implementing the same method, at least to a degree. At the time of writing, FEKO supported a type of scripting language, which has been used in places to automate the generation of complex geometries for MoM analysis; the constructs (`FOR` loops, `IF-THEN-ELSE` conditionals) are felt to be sufficiently generic to be useful in other codes supporting similar features.

Where appropriate, references are provided for further reading. In general, only those readily available in the English language archival open literature have been listed. On one or two occasions, internal reports have been included. The engineering community is divided on the use of such references; authors in the USA in particular often reference such reports in journal papers, which often prove frustratingly difficult to locate, sometimes being limited to US distribution only. In consequence, this has only been done when there is no other published version of the material. A similar problem can be encountered with theses; here, however some significant recent research has necessitated limited reference to recent dissertations, since these results are yet to appear in the archival literature.

The book draws primarily on the literature of Western science. Much work was done on computational electromagnetics in especially the former Soviet Union, but unfortunately little has been translated, and what has been is very difficult reading for electronic engineers trained in the Western tradition; it also tends to be at a much higher theoretical level than the main thrust of this book.

This book is an outgrowth of notes developed over a fifteen year period for a post-graduate course taught by the author at the University of Stellenbosch, South Africa, as well as a short course for industry taught by the author and several colleagues in 1999. Extensive integration of the material was undertaken during the author's sabbatical visit as a Guest Professor at the Delft University of Technology during 2003, where the course was also taught. Chapter 2 is adapted and extended from notes originally prepared by James T. Aberle at Arizona State University, Tempe, AZ, USA, and he is credited accordingly, but the rest of the authorship is that of DBD.

Acknowledgements

Stimulating careers are frequently the result of interactions with interesting people, and I would like to acknowledge a number of exceptional engineering scientists who have either mentored me, worked with me, or studied under me. My late father, an electronic engineer, spent much of his career working in the microwave and telecommunications industry in the UK and South Africa and sparked my early interest in electronic engineering; he started his career during the Second World War, working on some of the first radar sets deployed in South Africa (and later North Africa and Italy). John Cloete, Wynand Louw, Derek McNamara, and Jan Malherbe gave inspiring undergraduate and post-graduate courses at the University of Pretoria from 1981–1983, which originally fired my interest in this specific field. John and Derek continued as research supervisors for my M.Sc. and Ph.D. research on the MoM from 1986–1991. Dirk Baker gave me my first job at the CSIR in 1984; he is an outstanding antenna engineer and his scepticism of computed results was an invaluable baptism of fire. John Cloete offered me the opportunity to join the University of Stellenbosch in 1988 and we have continued to interact most fruitfully throughout my career.

Rick Ziolkowski taught me the power of the FDTD method during my post-doctoral stay at the University of Arizona in 1993. (Rick made significant contributions to the method and its applications, especially in complex material modelling.) Ron Ferrari and Ricky Metaxas kindly hosted me at Cambridge University during a sabbatical visit in 1997, where I had the opportunity to enrich greatly my knowledge of the FEM during frequent discussions with them and their students. Jim Aberle (Arizona State University) brought novel ideas to the teaching of the FDTD as well as spectral domain MoM methods, during a short course we taught in 1999; his ideas are reflected in places in this book. Leo Ligthart and Alex Yarovoy hosted me during my 2003 sabbatical at Delft University of Technology, during which time I initiated the actual writing of this book; their enthusiasm was very supportive.

Of my research students: in particular, the work of a number of my doctoral students is reflected in places in this volume, especially Frans Meyer – who went on to co-found Electromagnetic Software and Systems (Pty) Ltd., turning research ideas in CEM into commercially successful products – Marianne Bingle, Matthys Botha, Pierre Steyn, and Riana Geschke, and I would like to acknowledge their dedication to research excellence here. Frans and Matthys's work in particular is described in some detail in the final chapter. I would also like to thank Matthys for his proofreading and detailed comments on, and suggestions for, the final two chapters, which were most useful. Very useful interactions with a number of engineers (some of them previously my graduate students) at Electromagnetic Software and Systems are also reflected in this book, including Ulrich Jakobus (the original author of FEKO), Johann van Tonder, Isak Theron, Gronum Smith, Danie le Roux, and Sam Clarke. Many years of continuing discussions on electromagnetics with my colleagues at the University of Stellenbosch, in particular John Cloete, Petrie Meyer, Howard Reader, and Keith Palmer have also influenced the development of this book, as have those with colleagues in electronic engineering in general, in particular Dave Weber and the late David Frost.

I would also like to thank the (South African) National Research Foundation and its predecessor, the Foundation for Research Development, for many years of research funding, in particular grant-holder bursaries, equipment funding and sabbatical support.

Electromagnetic Software and Systems and Computer Simulation Technology kindly provided evaluation copies of FEKO and CST MICROWAVE STUDIO™ respectively. The former also provided the image on which the cover art was based. My thanks to Vanessa Weber for the graphic design she produced from this for the cover.

The love and forebearance of my wife Amor, who was bearing our first child Bruce during much of the period when this book was in preparation, was essential.

Finally, the support of the Cambridge University Press team is much appreciated.

To the reader

This book is designed to serve as an introduction to computational electromagnetics for radio-frequency applications. It assumes the reader has completed typical undergraduate courses in electromagnetic field theory, and has some basic knowledge of antenna design and microwave systems.

For readers in a hurry, who already know which of the techniques discussed they would like to learn more about, it is possible to go directly to the relevant chapters, but it would nonetheless be useful first to read the introductory chapter. For those in a hurry, but who need first to find out which method (or methods) to use, this chapter is essential reading.

For readers who intend working through most of the book, it would be best to work through it in the sequence presented, although the chapters on the Sommerfeld formulation and practical applications thereof could be omitted without interrupting the sequence of presentation. A more detailed outline of the book may be found in Section 1.11; this will also assist readers to locate rapidly the parts of the book of interest to them.

At the end of each chapter, a list of references linked to the chapter topic is presented, for further reading and study.

Notation

Throughout this book, the following notation is used. Spatial vectors are indicated as $\vec{E}$ (in this case, the electric field). Vectors in the linear algebra sense are indicated as $\{x\}$, and matrices as $[A]$. The individual elements of a vector or matrix are of course indicated as x_i or A_{ij} respectively. Otherwise, the notation is as generally encountered in engineering books on this topic. A summary is presented below.

The time convention used for phasor quantities is $e^{j\omega t}$, hence, an e^{-jkr} plane wave propagates in the direction of increasing r. (Note that physics books often adopt the $e^{-i\omega t}$ convention, in which case the sign also changes in the plane wave exponential factor.)

$\nabla\times$	the curl operation
$\nabla\cdot$	the divergence operation
$\times$	the vector cross product of two vectors
$\vec{E}$	the (field) vector E
ϵ_0	the permittivity of free space ($\approx 8.854 \times 10^{-12}$ F/m)
ϵ_r	relative permittivity of a dielectric material (dimensionless)
μ_0	the permeability of free space ($4\pi \times 10^{-7}$ H/m)
μ_r	relative permeability of a magnetic material (dimensionless)
c	the speed of light in free space ($\approx 2.9979 \times 10^8$ m/s)
λ	wavelength [m/s] *or* real part of spectral variable k_ρ (the meaning will be clear from the context)
λ_i	simplex coordinate i
$\mathcal{O}(M^n)$	of the order of M^n, formally, $\mathcal{N} = \mathcal{O}(M^n) \Rightarrow \lim_{M\to\infty} \log\mathcal{N}/\log M = n$
$[A]$	the matrix A
a_{ij}	the ijth element of matrix A
$\{x\}$	the (algebraic) vector x

x_i	the ith element of vector $\{x\}$
$\|\|\{x\}\|\|$	the Euclidean norm of the vector $\{x\}$ of length n, $\|\|\{x\}\|\| \equiv \sqrt{\sum_{i=1}^{n} \|x_i\|^2}$
$\equiv$	is defined as
$\forall$	for all
$\|z\|$	absolute value of z
$\Rightarrow$	implies

1

An overview of computational electromagnetics for RF and microwave applications

Even if we do discover a complete unified theory, it would not mean that we would be able to predict events in general . . . even if we do find a complete set of basic laws, there will still be in the years ahead the intellectually challenging task of developing better approximation methods, so that we can make *useful predictions of the probable outcomes in complicated and realistic situations*.

From [1, pp. 168–169] (the present author's emphasis).

Computations: no-one believes them, except the person who made them.
Measurements: everyone believes them, except the person who made them . . .
Attributed to Professor B. Munk, Ohio State University.

1.1 Introduction

Electromagnetics, the study of electrical and magnetic fields and their interaction, has been one of the core technologies of the twentieth century, and shows every sign of continuing this into the twenty-first. Whilst there are many useful ways of subdividing the field, power frequency versus radio frequency, or alternatively quasi-static versus full-wave, is one of the most insightful here. This book focusses exclusively on radio-frequency, full-wave electromagnetic modelling, as typically encountered in communication systems.

The core of modern electromagnetic engineering is of course Maxwell's equations. Written in modern form,[1] they are:

$$\nabla \times \vec{E} = -\frac{\partial}{\partial t}\vec{B} \tag{1.1}$$

$$\nabla \times \vec{H} = \vec{J} + \frac{\partial}{\partial t}\vec{D} \tag{1.2}$$

[1] Maxwell did not actually write his equations in this form; vector analysis was a late nineteenth-century development.

$$\nabla \cdot \vec{D} = \rho \tag{1.3}$$

$$\nabla \cdot \vec{B} = 0 \tag{1.4}$$

with the associated constitutive equations

$$\vec{B} = \mu \vec{H} \tag{1.5}$$

$$\vec{D} = \varepsilon \vec{E} \tag{1.6}$$

The actual *solution* of the Maxwell equations is complex, and for realistic problems, approximations are usually required – as indicated by the introductory quote from Hawking, although he had in mind an altogether more ambitious theory (of everything!). The numerical approximation of Maxwell's equations, the subject of this book, is known as *computational electromagnetics* (CEM).

CEM techniques have been available for close on four decades now. These techniques have gestated, grown and matured to the point where they form an invaluable part of current RF and microwave engineering practice [2]. However, the widespread adoption of computational methods to complement the traditional tools of analysis and measurement has attracted criticism, summarized with more than a grain of truth by the second quote at the beginning of the chapter. Ironically, the availability of powerful, commercial codes may well have made the situation worse, not better, since more and more frequently, codes are being applied by users unfamiliar with the basic formulations underlying the codes, and not infrequently to problems for which the codes were not designed. One of the major aims of this book is to make RF computational electromagnetics comprehensible and accessible to a far wider group of RF engineers than has been the case in the past.

CEM is a multi-disciplinary field. Its core disciplines are electromagnetic theory and numerical methods, but for useful implementations, geometric modelling and visualization, computer science and algorithms all have important roles to play. In this book, the focus falls on the core disciplines.

The applications of CEM are legion, and include antennas, biological EM effects, medical diagnosis and treatment, electronic packing and high-speed circuitry, superconductivity, microwave devices, monolithic microwave integrated circuits, law enforcement, environmental issues, materials, avionics, communications, energy generation and conservation, low observable vehicles (stealth), radars and imaging, surveillance and intelligence gathering. In this book, we focus primarily on applications in antennas, wireless communications, radar, and (passive) microwave devices, although an example will be given of a biological EM effect study.

An historical aside – a brief history of electromagnetics

Interest in static electricity and magnetism, of course, dates back to ancient times. The Ancient Greeks circa 400 BC noted that rubbing amber attracted bits of straw, and the Chinese reportedly found lodestones (natural magnets) circa 2600 BC, first using them for burial purposes, and later for navigation. The modern study of electromagnetic phenomena dates to the late eighteenth century, with the great progress in experimental methods by Alessandro Volta (1745–1827), Hans Christian Oersted (1777–1851) and Michael Faraday (1791–1867) on the one hand, and the more mathematical modelling approach of Charles Augustin de Coloumb (1736–1806) and André-Marie Ampère (1775–1836) on the other. Amongst these, the following milestones stand out: the development of the battery by Volta provided a continuous source of electricity for the first time; Coloumb's careful measurements of the electric force resulted in the famous inverse square law; Oersted's 1820 discovery showed that (direct) current deflected a magnet; Ampère developed mathematical laws describing this and the force between current carrying wires; and finally, Faraday's crucial contribution in 1831 showed that a changing magnetic field sets up an alternating current (i.e. an electric field), and for the first time connected two forces of nature which until then had been thought quite independent.

James Clark Maxwell (1831–1879), the most brilliant physicist of the nineteenth century,[a] combined the work of his predecessors in elegant theoretical fashion and postulated that changing electric fields should generate magnetic fields; he then showed that this implied *wave motion*. Hermann Ludwig-Ferdinand Helmholtz (1821–1894) was one of the first to recognize the significance of Maxwell's predictions in this regard; in 1888, his student Heinrich Rudolph Hertz (1857–1894) showed experimentally that electromagnetic fields indeed propagate, and at the speed of light. Oliver Heaviside (1850–1925) also made contributions in this regard, although his work is not widely recognized nowadays [3]. In what we would now describe as the first commercial spin-off of this work, Guglielmo Marconi (1874–1937) was the first to profit financially from the emerging field of wireless.

Electromagnetics was also to have a profound influence on the outstanding physicist of the twentieth century, Albert Einstein (1879–1955). Perhaps less well known than some of his results – certainly amongst the general public – Einstein showed that the magnetic field is the relativistic correction of the electric

[a] Maxwell not only unified electricity and magnetism in 1864, he also developed the kinetic theory of gases, before his life was cut tragically short by illness.

field, confirming the unified field theoretic nature of Maxwell's electromagnetic theory.

The above is the conventional Western history of electromagnetics. Contributions to the theory of light, intimately connected to electromagnetics, were made by many over an extremely long period of time, including contributions from Arabic scholars. An exceptionally erudite historical perspective may be found in [4].

1.2 Full-wave CEM techniques

Full-wave CEM methods approximate the Maxwell equations numerically, without any initial physical approximations being made. These are also sometimes called *low-frequency* methods, to distinguish them from asymptotic *high-frequency* methods, but this can be confusing for several reasons.[2] The full-wave techniques which will be studied in this book are the finite difference time domain (FDTD) method; the method of moments (MoM), and the finite element method (FEM). Whilst there are other methods available, these are the most widely used, and all have been implemented in powerful computer codes. These techniques are frequently classified further by whether they are based on integral or differential equations, and by whether they operate in the time or frequency domain. We will discuss this in the context of each method subsequently.

Sometimes, the expressions "static" or "quasi-static" will be used. The former applies obviously to the situation where one is dealing with either steady-state charges (and the associated electric fields) or currents (and the associated magnetic fields). The latter applies to situations where the time rate of change is low enough that the fields still satisfy the static equations to a very good approximation – or put differently, the $\frac{\partial}{\partial t}\vec{B}$ term in Eq. (1.1) is negligible (in which case one obtains electroquasistatics) or similarly for the $\frac{\partial}{\partial t}\vec{D}$ term in Eq. (1.2) (which yields magnetoquasistatics). A very detailed discussion of these approximations and their use may be found in [5]. However, we will not pursue this far in this book, which deals almost entirely with full-wave methods.

There is another class of numerical method for solving the Maxwell equations, generally called the *asymptotic* techniques. These methods require fundamental approximations in the Maxwell equations, the validity of which increases asymptotically with frequency. Examples are physical optics (PO), geometrical optics

[2] Firstly, the high-frequency radio band is specifically the spectrum from 3–30 MHz; secondly, the meaning of low and high are entirely relative, and the same methods may be, and are, useful from power frequencies up to the visible spectrum and beyond; and finally, "high-frequency" as a general term in electronic engineering is widely used to distinguish from "power frequency," with the latter usually using quasi-static approaches.

(GO) and the uniform theory of diffraction (UTD). This is a field of study in its own right. For suitable problems, these methods are very powerful, but the underlying approximations of the physics limits their use for general problems. Furthermore, unlike the full-wave methods, where Moore's law and the resulting increase in computer speed and memory continually extend the limits of applicability, the asymptotic methods have fundamental limits. Hence, in this book, only full-wave methods are considered. However, a hybridization with an asymptotic technique will be discussed as an example of an advanced application.

The full-wave techniques are potentially very accurate. Central to *all* these methods is the idea of *discretizing* some unknown electromagnetic property, typically the surface current for the MoM, and the $\vec{E}$ field for the FEM and FDTD method. (For the latter, the $\vec{H}$ field is also discretized.) This process of discretization is also known as *meshing*. It entails subdividing the geometry into a (large) number of small *elements*. These may be one-dimensional segments, two-dimensional surface "patches" (often triangles), three-dimensional tetrahedral elements, or a regular three-dimensional "staggered" grid, depending on the problem at hand and the method used. Within each element, a simple functional dependence is assumed for the spatial variation of the unknown – for instance, a linear approximation – but the *amplitude* (and possibly phase) of the unknown is determined by application of the method to the patchwork of elements which approximates the original geometry. This functional dependence is also known as the basis (or expansion) function.[3]

Generally, the accuracy of the methods is related to the discretization (i.e. mesh size). The finer is the mesh, the better is the accuracy of the methods.[4] The largest mesh size (alternatively, the finest geometrical resolution) is limited by the available computational resources. In other fields such as structural mechanics, the mesh fineness is usually determined by the requirement to resolve the *structural geometry* adequately; in radio-frequency electromagnetics, the requirement on the mesh is usually to sample the *phase* adequately. For many years, the CEM community has worked with a rule of thumb of ten segments per wavelength. This was originally derived for wire antenna problems, where the mesh is one-dimensional; for surfaces, this guideline becomes 100 segments per square wavelength (and a similar extension for volumetric meshes to 1000 per cubic wavelength). Much work on better elements has been done to reduce this requirement – it will readily be appreciated that as the dimensionality of the problem goes up, so this becomes

[3] With the FDTD method as usually introduced, the fields are sampled at points; it is however possible to define basis functions for the FDTD, a topic we discuss briefly in Chapter 10.

[4] This is not invariably true: limitations imposed by approximations in the formulations may place some lower bound on element size. A classic example is a thin-wire MoM formulation, where using too many segments may violate the underlying thin-wire assumptions. This is discussed in detail in Section 4.3.

progressively more crucial. It should also be noted that when very accurate field data are required – for example, when computing antenna input impedance – a *finer* mesh may be required, at least locally around the feed point of the antenna. Furthermore, this guideline ignores the problem of dispersion in differential equation based solvers, which effectively requires *denser* meshes for electromagnetically larger problems.

Although the full-wave methods share the basic idea of discretization, and indeed have been viewed within a very general framework as simply different implementations of one overarching theoretical formulation, in practice, the methods have quite different challenges for theoreticians, code developers and users, as well as different optimal areas of application, and as such, they will be considered separately in this overview chapter. In Chapter 10, some of the underlying mathematical connections between the methods will emerge.

In the rest of this overview chapter, the MoM, FEM and FDTD method will be reviewed *qualitatively*, emphasizing basic principles such as the underlying formulation (integral/differential equation based, frequency or time domain) and areas of application (perfectly or highly conducting materials versus homogeneous or inhomogeneous penetrable structures; microwave devices versus radiation or scattering analysis). This review is especially designed for readers who have a particular problem to solve, but are not sure which is the best method to use. Details of each method will be found in the subsequent chapters of the book. Key references only are given; a far more extensive list of references will be found at the end of each chapter.

By way of introduction, some of the most important characteristics of the MoM, FEM and FDTD method are presented in Tables 1.1 and 1.2. Table 1.1 provides a comparison of the methods for open region (radiation and scattering) problems. It is important to note that what is presented in this table are the key characteristics of the method *as widely implemented and understood in the CEM community*. As will be seen in the description of each method in the following sections, a number of

Table 1.1 *Strengths and weaknesses of CEM methods as widely implemented for open region problems*

Formulation	Equation type	Domain	Radiation condition	PEC only	Homogeneous penetrable	Inhomogeneous penetrable
MoM	Integral	Frequency	Yes	⌣	⌣	⌢
FEM	Differential	Frequency	No	⌢	⌣	⌣
FDTD	Differential	Time	No	⌢	⌣	⌣

Key: ⌣ good; ⌢ not optimal.

Table 1.2 *Strengths and weaknesses of CEM methods for guided wave problems*

Formulation	Equation type	Domain	Wideband	PEC only	Homogeneous penetrable	Inhomogeneous penetrable
MoM	Integral	Frequency	∼	⌣	⌣	⌢
FEM	Differential	Frequency	∼	⌣	⌣	⌣
FDTD	Differential	Time	⌣	⌣	⌣	⌣

Key: ⌣ good; ∼ satisfactory, but not necessarily the best; ⌢ not optimal.

simplifications have been made in this table: the MoM, for instance, can be seen in a more general sense as including the FEM, although this is not normal usage; and to give another example, the FEM can also operate in the time domain, but there are no commercial implementations of this at present. For the MoM, homogeneous penetrable materials (dielectrics, for instance) can either be modelled using equivalent surface currents or, if the problem consists of layered materials, using a Sommerfeld formulation. This has not been noted in the table, since it depends on the details of the problem. Table 1.2 provides a similar comparison of the methods for guided wave problems.[5] Again, the details of the precise implementation have not been commented on.

1.3 The method of moments (MoM)

The MoM is probably the most widely used numerical technique in RF CEM, and has a long history in the field; some of this is presented in Chapter 4. For antenna engineering, the MoM has been the most widely used CEM method.[6] In the method of moments, the radiating/scattering structure is replaced by equivalent currents. These are normally surface currents. (Volumetric currents can be used for inhomogeneous dielectric bodies. This is however very expensive computationally.) This surface current is discretized into wire segments and/or surface patches. A matrix equation is then derived, representing the effect of every segment/patch on every other segment/patch. This interaction is computed using the Green function for the problem. (Green functions will be discussed later in this

[5] It is tempting to use the term "closed problems" here, but a number of important guiding structures, such as microstrip, are partially open. It is assumed in this table that FEM and FDTD codes have an appropriate method of terminating this region. Since the energy decays rapidly away from the guiding structure, and this radiation is a secondary effect in most applications, the open boundary is usually less problematic here than in the case of the radiation and scattering problems.

[6] The name "method of moments" is peculiar to the CEM community. Perhaps the most descriptive alternative name is the "method of weighted residuals." The term "boundary element method" is frequently used synonymously with MoM, and for surface formulations this is correct, but there are some moment method formulations which use volume, not boundary, elements. We discuss this further in Chapter 4.

book – indeed, an entire chapter, Chapter 7, is devoted to one such function.) Most MoM codes use the free-space Green function. The relevant boundary condition is then applied to all the interactions, yielding a set of linear equations. The solution of this linear system yields the (approximate) current on each segment/patch. The resulting matrix which must be factored (or used in an iterative solution scheme) is fully populated, with complex valued entries. Typical matrix dimensions range from some hundreds for small antenna problems to several thousand – the upper limit is imposed by computational limitations, either limited memory and/or excessive run-time.

Traditionally, the MoM has been applied in the frequency domain, i.e. single frequency, or monochromatic, sinusoidal excitation, with an $e^{j\omega t}$ convention assumed. The working variables (unknowns) are thus complex valued, with a magnitude and phase, as for any phasor analysis. Time domain integral equation (TDIE) formulations have been used on occasions, but stability and other issues have proven difficult, and TDIE codes are rare.

The use of the MoM for antenna analysis was given a major boost by the US government's de facto decision during the late 1980s to release the Numerical Electromagnetic Code – Method of Moments (widely known as NEC-2) into the public domain. NEC-2 is a powerful, general-purpose antenna modelling program, but with no graphical abilities whatsoever and very limited meshing abilities. NEC-2 is discussed in Chapter 5. A later version, NEC-4, added some specialized functionality. At present, there are some excellent commercial codes which offer all the functionality of NEC-2, but with proper graphical user tools and frequently greatly enhanced abilities; examples are FEKO (which will be used quite extensively in this book), SuperNEC, Ensemble, and IE3D. (Only SuperNEC is a direct descendant of NEC, the others are independent implementations.) There are also some semi-commercial packages such as GEMACS which are limited to US Department of Defense contractors, and hence not generally available for commercial use world-wide.

The strong points of the MoM (as usually applied) are the following.

- Efficient treatment of perfectly or highly conducting surfaces. Only the surface is meshed; no "air region" around the antenna need be meshed. For wire antennas, the treatment is even more efficient, since only a one-dimensional discretization of the wire is undertaken.
- The MoM automatically incorporates the "radiation condition" – i.e. the correct behavior of the field far from the source (proportional to $1/r$ in free space). This is very important when dealing with radiation or scattering problems.
- The working variable is the current density, from which many important antenna parameters (impedance, gain, radiation patterns etc.) may be derived, some directly and some via straightforward numerical integration.

- Via the Sommerfeld potentials, efficient formulations may be derived for stratified (layered) media. Important examples are printed antennas, components and feed networks (e.g. microstrip technology) and antenna-above-real-earth calculations.
- The availability of NEC-2 in the public domain – this powerful code has served as the basis for much MoM based antenna design, and due to the open source nature, has lent itself to all manner of numerical experimentation and improvement.

The weak points of the MoM may be summarized as follows.

- The MoM does not handle electromagnetically penetrable materials as well as differential equation formulations. (If the materials are homogeneous a, fictitious, equivalent surface current formulation may be used, but inhomogeneous materials require fictitious equivalent volumetric currents, and become very expensive computationally.)
- The MoM does not scale gracefully with frequency – for typical applications requiring a surface mesh, the scaling is $\mathcal{O}((kd)^6)$ where kd is the electromagnetic size of the structure.[7] (This assumes a cubic structure, for simplicity.) Note that this is implies an $\mathcal{O}(f^6)$ scaling – doubling the frequency can result in a run-time 64 times as long! We will see that this is a major problem with all the computational methods, although the details do vary slightly from method to method. For an MoM volumetric mesh, required by an inhomogeneous structure, the scaling is $\mathcal{O}((kd)^9)$; this is so large that such methods are usually very limited in application.
- Some MoM formulations, in particular those based on the magnetic field integral equation (MFIE), require the surface to be *closed*. This is frequently impractical.

In conclusion, the MoM is the preferred method for frequency domain radiation and scattering problems involving perfectly or highly conducting wires and/or surfaces. If the problem involves inhomogeneous dielectric materials, it is unlikely to be the best formulation, but if hybridized with the FEM a very efficient formulation can result.

1.4 The finite difference time domain (FDTD) method

The finite difference time domain (FDTD) method is of a similar vintage to the MoM and FEM in electromagnetics, dating back to the 1960s. Like the FEM, it is partial differential equation based, and one does *not* need a Green function. *Unlike* the FEM, the FDTD method does not use variational functionals or weighted residuals – it directly approximates the differential operators in the Maxwell curl equations, on a grid staggered in time and space. $\vec{E}$ and $\vec{H}$ fields are computed on a regular grid, with a marching-on-in-time discretization of time, with field

[7] The notation $\mathcal{O}(x)^p$ means *of the* (*asymptotic*) *order of* and indicates to the highest power (p) present in the variable (x); note that it says nothing of the constants. This can be important, since CEM analysis is quite often undertaken in the "pre-asymptotic" region, where lower powers in x may dominate especially run-time.

components being offset by $\Delta s/2$ relative to each other and the $\vec{E}$ and $\vec{H}$ fields evaluated $\Delta t/2$ apart in time, where Δs and Δt are the spatial and temporal discretizations respectively. This permits a scheme which uses first-order numerical differentiation to provide second-order accuracy. It is also the only widely used CEM scheme to operate in the time domain. (Time domain MoM and FEM formulations have been used, but usually for a rather specialized application. Frequency domain finite difference formulations are also available, but again have never become very popular for general problems.)

Some history of the FDTD method may be found in Chapter 2. For various reasons, the method languished in relative obscurity throughout most of the 1960s and 1970s, but sprang to prominence in the 1980s. There were both technological driving factors behind this – on the one hand, increasing interest in the modelling of inhomogeneous materials, in particular for the assessment of human exposure to RF fields, and on the other, the development of low-observable "stealth" technology – and enabling technology in the shape of the enormous growth in computer power – in particular, memory, for which the FDTD method has a voracious appetite in three dimensions. The development by Berenger of the perfectly matched layer in 1994 solved the previously problematic issue of mesh termination, and removed the last hurdle to the widespread adoption of the method. In the new millennium, with desktop PCs with hundreds of megabytes available at relatively low cost, the FDTD method has firmly established itself as one of the most popular methods in CEM, both in industry and academia. The apparent simplicity of the basic implementation also means that it is very popular with graduate students in the university research community, where "do-it-yourself" FDTD codes are commonly encountered.

Critics have dismissed the method as a "brute force" technique and, certainly, compared to the mathematical elegance and subtleties of a Sommerfeld integral formulation, the basic method appears to make limited demands on higher mathematics. Most engineers trying to solve tough problems are of course more impressed by how well a code works, rather than by how elegant the formulation is, and the FDTD method has been enormously successful in many diverse applications. Nonetheless, extensions of the FDTD method have required subtle thinking, as have stability proofs, so the "brute force" epithet is undeserved.

The FDTD method is an "explicit" finite difference approach, i.e. no matrix equation is set up and solved.[8] The term comes from the update equations, where the field values at the next time-step are given entirely in terms of the field at this and the previous time-steps. (Implicit finite difference approaches, where the field

[8] In Chapter 10, it is shown that one can alternatively view the FDTD as derived from a diagonalized matrix equation.

values at the next time-step at a point in space also involve the values at adjacent points at the next time-step, generate a sparse matrix, which must be solved at each time-step.) This has the great advantage of keeping the required operations very simple – essentially just a stencil involving differencing neighbors in time and space – but does mean that the method is not unconditionally stable. This means that there is an upper limit on the time-step, and it turns out to be rather less than the Nyquist sampling criteria would imply, which is the price one pays for the simplicity of the explicit approach. In three dimensions, the stability criterion (widely known as the Courant limit) is $\Delta t < \Delta s/(\sqrt{3}c)$ where Δs is the smallest grid dimension and c is the speed of light in the mesh.

There are three very good texts on the FDTD method; Kunz and Luebber's was the first [6], appearing in 1993, but Taflove's 1995 volume [7] (recently revised [8]) and the 1998 companion [9] are currently the state of the art. (Kunz and Luebbers were unfortunate to publish their book just before the revolutionary perfectly matched layer (PML) was invented by Berenger in 1994, although the book still contains useful material, not least a working FDTD code. This code has served as the basis for a number of academic codes.)

The time domain formulation of the FDTD method is both an attractive feature and a drawback. Using wideband sources, the FDTD method can compute a wideband response in one run, whereas frequency domain methods must obviously recompute the system response for each frequency point.[9] However, the majority of RF devices operate over quite a narrow frequency band, and this may be less of an advantage than one might expect. In particular, for high-Q devices, a very large number of time-steps may be required to obtain sufficient frequency resolution.[10] Many systems exhibit *dispersive* properties; examples are waveguides and most real dielectric and magnetic materials. In the frequency domain, this is simple to handle by the obvious expedient of simply changing the material/device properties with frequency, but in the time domain this is more challenging since a convolution is implied. Many techniques have been proposed and implemented to address this issue, but do complicate the method somewhat [8, 9].

Although there are some FDTD codes available on the internet, they are really "toy" codes by comparison with NEC-2, for instance. Commercial versions are available, including CST MICROWAVE STUDIO® and REMCON's XFDTD; the former actually uses the finite integration technique, but this is very closely related to the FDTD. It is perhaps surprising that more contenders have not emerged, but this is in no small part because a useful commercial code has to incorporate not

[9] Continuing research for frequency domain codes on model based parameter estimation (MBPE) aims to reduce dramatically the number of frequency points required, and good results have been obtained; some commercial frequency domain codes already incorporate this.

[10] Again, work similar to the MBPE, using system approximation techniques, can assist here.

only a decent user interface, but also a number of extensions to the standard FDTD to make it generally useful.

The strong points of the FDTD method are the following.

- Exceptionally simple implementation for a full-wave solver – at least an order of magnitude less work than either an MoM or FEM implementation for a basic FDTD implementation. (One should be warned however that there are a number of subtleties which can take a while to appreciate, even with an apparently simple problem. Also, many practical problems require more than just a basic implementation, and the simplicity of the method is often compromised by these extra factors.) It is the only method which one can realistically implement oneself in a reasonable timeframe, although even then only for quite specific problems.
- Very straightforward treatment of material inhomogeneities (as for the FEM).[11]
- Fairly accurate geometrical modelling ability (but not as versatile as the FEM in this regard, due to the "stair-stepping" effect of the regular mesh – see comments below on non-orthogonal grids). (Commercial codes frequently include extensions to the method to improve this, so it is not necessarily a problem.)
- Since the method is a time domain one, wideband data are potentially available from one run.
- Reasonable scaling behavior, $\mathcal{O}((kd)^{5.5})$, with the same $N \propto (kd)^{1.5}$ assumption to control dispersion as for the FEM, which we will discuss shortly. Note that as for the FEM, this is not affected by the material composition of the structure. For wideband systems, this is very attractive, since the other methods have an implied f_n multiplicative term (not shown in the preceding sections), where f_n is the number of frequency points required.
- The PML has made implementing very good absorbing boundary conditions as mesh termination relatively straightforward.

The main drawbacks are the following.

- Inflexible meshing – much work has been done on non-orthogonal FDTD grids, but the method then loses much of its appealing simplicity.
- Some uncertainty about the precise position of boundaries – usually an uncertainly of about $\Delta s/2$. This is due to the offset nature of the $\vec{E}$ and $\vec{H}$ field grids.[12]
- Dispersive materials require considerable effort to implement correctly – but it is possible and good results have been obtained.
- As with the FEM, the FDTD method is not as efficient as the MoM when modelling structures consisting entirely of perfectly or highly conducting radiators/scatterers.

[11] A point worth making here is that for typical RF applications, the dielectric properties of materials are usually the most significant, and relative permittivities at RF and microwave frequencies rarely exceed single figures. For low-frequency magnetoquasistatic problems, magnetic properties are often the most significant, with relative permeabilities which can be very large indeed. In this case, the matter is not quite as simple when accurate modelling is desired, and both the FDTD method and the FEM can exhibit problems. This, however, is not the focus of this book.

[12] Work has been done on improving this, typically using some averaging of properties in the FDTD cells on the boundaries, but this can impact on the second-order accuracy of the method.

- Although considerable theoretical work has been done on higher-order FDTD approaches, none appears to have been successfully implemented in a general-purpose FDTD code. The problem is intimately linked to that mentioned above regarding the ambiguity of the boundary positions.

In conclusion, the FDTD method is the preferred method for wideband systems. Even in its standard Yee form, it is also a strong contender for *any* electromagnetic radiation or scattering problem for which quick answers are needed, great accuracy is not the primary concern, and quite large run-times and memory usage is acceptable. Furthermore, by using a sufficiently fine mesh, and in particular using various extensions to the standard FDTD method, very accurate results may be obtained; the potential accuracy of the FDTD method should not be underestimated.

1.5 The finite element method (FEM)

The finite element method (FEM) has been widely used in structural mechanics and thermodynamics; its first application in the modern form dates to the 1950s, although its mathematical roots are older, and the first application in electromagnetics was undertaken in the late 1960s. Chapter 9 gives some more historical background on the method.

As with its main competitor, the FDTD method, the FEM handles inhomogeneous materials and complex geometries with aplomb; these become problems in mesh generation rather than in electromagnetic theory. The FEM may be derived from two viewpoints: one uses variational analysis, the other weighted residuals. Both start with the partial differential equation (PDE) form of Maxwell's equations. The former finds a variational functional whose minimum[13] corresponds with the solution of the PDE, subject to certain boundary conditions. The latter also starts with the PDE form of Maxwell's equations, and then introduces a "weighted" residual (error); using Green's theorem, one of the differentials in the PDE is "shifted" to the weighting functions.[14] For most applications, these procedures result in identical equations. In both cases, the unknown field is discretized using a finite element mesh; typically, triangular elements are used for surface meshes and tetrahedrons for volumetric meshes, although many other types of elements are available. Triangles and tetrahedrons have certain attractive properties best summarized as "simplicial" – these are the simplest geometrical forms with which two-dimensional and three-dimensional regions respectively can be meshed.

[13] More precisely, extremal point, since it may also be a maximum or stationary point.
[14] From which comes the name "weak" formulation, sometimes encountered in the literature, since the finite element basis functions need only be once differentiable, whereas the wave equation has second-order derivatives.

Finite element analysis (FEA) can handle two different types of problem, viz. eigenanalysis (source-free) and deterministic (driven) FEA problems.[15] Problems *without* any internal (or external) field source fall into the category of eigenanalysis problems. A classic example is a cavity resonator. What emerges from the analysis is a set of eigenvalues and associated eigenmodes; these represent the resonant frequencies and associated field distribution within the cavity. For microwave dielectric heating, this information can be used to design feed locations and optimize load positioning.[16] It should be noted that eigenanalysis applications are neither time nor frequency but rather eigenvalue domain solvers; using a simple transformation, it is possible to include operating frequency in a waveguide simulation, to compute dispersion curves.

Deterministic problems analyzed using FEA involve a source; the response of the structure to this excitation is then computed. This represents a very large class of electromagnetic engineering applications of the FEM, including antenna, radar cross-section, microwave circuit and periodic structure analyses.

As with the FDTD method, the FEM does *not* include the radiation condition. For closed regions (e.g. waveguide devices or cavities) this is of no concern. However, for open regions (e.g. radiation or scattering problems), this requires special treatment, and this must be incorporated using either an artificial absorbing region within the mesh (the numerical analogy of an anechoic chamber) or using a hybridization with the MoM to terminate the mesh.

Traditionally, the FEM has been formulated in the frequency domain, although time domain formulations have also been used for specialized applications.

There are a number of excellent and up-to-date texts on the FEM, including those by Jin (revised in 2002) [10], Silvester and Ferrari [11] (the third edition appearing in 1996), Volakis *et al*. [12] and Peterson *et al.* [13] (both published in 1998, the last also including comprehensive coverage of the MoM). The collection of papers edited by Silvester and Pelosi [14] is also very useful, although quite a number of significant papers have appeared since its 1994 publication. Another useful source is the 1996 volume edited by Itoh *et al*. [15].

Several companies market commercial finite element products for radio-frequency electromagnetics. Ansoft's HFSS package is widely regarded as the market leader; Ansys have a suitable product, and a fairly recent entry, FEMLAB, has also attracted users.

[15] The MoM and finite difference methods in general can also be used for eigenanalysis, but are not very commonly encountered. Harrington's original text on the MoM included a chapter on eigenvalue problems, but the MoM has not been as widely used as the FEM for this class of problem. The FDTD method is by definition deterministic, and requires a source.

[16] In this real-world application, there is now a source and the problem is strictly speaking no longer an eigenanalysis one, but the source location can be optimized by knowledge of the resonant field behavior within the cavity, since these fields are what one is attempting to excite with the feed.

The strong points of the FEM are the following.

- Very straightforward treatment of complex geometries and material inhomogeneities.
- Very simple handling of dispersive materials (i.e. materials with frequency-dependent properties).
- Ability to handle eigenproblems as above.
- Potentially better frequency scaling than the MoM – although the requirement to mesh a volume rather than a surface means that the number of unknowns in the problem is usually much larger.[17]
- Straightforward extension to higher-order basis functions. The FEM lends itself to the use of higher-order basis functions; although the book-keeping within an FEM code is a little complicated by this, the theoretical extensions are now well understood. It is also possible to use conformal elements to better approximate curved geometries.
- "Multi-physics" potential – this means the ability to couple EM solutions with, for instance, mechanical or thermal solutions. Due no doubt to the widespread popularity and maturity of the FEM in other fields of engineering, one is starting to see packages which can compute such coupled solutions. It is probably only significant in high-power applications, where thermal effects can be important – either desired, as in the case of microwave dielectric heating, or undesired, such as with high-power transmitter design.

The weak points of the FEM include the following.

- Inefficient treatment of highly conducting radiators when compared to the MoM (due to the requirement to have some mesh between the radiator and the absorber).[18]
- The FEM meshes can become very complex for large three-dimensional structures – indeed, some workers have reported mesh generation times starting to exceed solution time.
- The FEM is rather more complex to implement than the FDTD method. This impacts in particular in terms of the suitability of the FEM for parallel computing. It also implies that "home-built" FEM codes are quite rare compared to such FDTD codes.
- Efficient preconditioned iterative solvers are required when higher-order elements are used; so important is this in commercial applications that these are usually treated as proprietary information, making "do-it-yourself" implementation even more challenging.

[17] The exact scaling behavior depends on how efficiently the sparsity of the finite element matrix can be exploited – and the sparsity pattern is problem dependent. The lowest bound on this is $\mathcal{O}(N)$, N being the number of degrees of freedom (unknowns). For a scheme with second-order accuracy, $N \propto (kd)^{1.5}$, where the exponential indicates that as the problem size grows, so the mesh must become proportionally finer to control mesh dispersion. This effect was often overlooked in earlier analyses of differential equation based solvers. With these assumptions, the lowest bound on the FEM is $\mathcal{O}((kd)^{4.5})$; it must be emphasized that this is a lowest bound and assumes that matrix sparsity is essentially fixed, which is not so in reality. A more realistic estimate is probably $\mathcal{O}((kd)^5)$ – $\mathcal{O}((kd)^6)$.

[18] The FEM/MoM hybrids overcome this problem.

In conclusion, the FEM is the preferred method for microwave device simulation and eigenproblem analysis. Using FEM/MoM hybrids, scattering problems involving electromagnetically penetrable media and specialized antenna problems can be accurately and efficiently solved.

1.6 Other methods

The MoM, FEM and FDTD method are the most popular methods in current use. There are a number of other methods which will be encountered in the literature, and some commercial codes are based on these methods. Here we will briefly outline some of them.

1.6.1 Transmission line matrix (TLM) method

The TLM method is conceptually very similar to the FDTD method. Instead of directly discretizing the Maxwell equations, an equivalent array of short transmission lines is used. The method is appealing to engineers with a strong circuit but weak field background, but for most CEM practitioners the circuit approximation of the field equations seems rather circuitous. It should be commented that the circuit approach can be more direct than the FDTD approach when one is dealing with high-frequency circuits, which is a major reason for continuing work on the TLM. The method has a dedicated following in some circles, and at least one commercial code, Micro-Stripes, is available.

1.6.2 The method of lines (MoL)

The MoL is a specialized method for primarily waveguiding structures. It uses a semi-analytical solution along a number of lines (in its two-dimensional form) and is especially memory efficient. It is also very accurate. Because it requires the extraction of eigenvalues, it can be computationally expensive. Most MoL applications can be done as well with an FEM formulation. A commercial implementation does not appear to be available at present.

1.6.3 The generalized multipole technique (GMT)

The GMT uses multipoles as the basis functions; these are special function solutions of the Maxwell equations. It is not especially similar to any of the methods that we have discussed thus far. It does require some intelligent user input in terms of placing the multipoles. Good results have been obtained for a variety of problems; a good reference is Hafner's book [16], which is also useful for placing

the other methods in perspective. (The book will appear somewhat idiosyncratic though for CEM novices.)

1.7 The CEM modelling process

Before we now proceed to study these methods in more detail in subsequent chapters, it is useful to comment on the modelling process in general. Some astonishing claims have been made about the predictive power of Maxwell's equations [17, p. 4]:

> Most physicists believe that if you lock a graduate student in a room and have him perform an electromagnetic calculation correctly, and if you perform an experiment that does not agree with the graduate student's calculation, then you better check your experiment.

Whilst at its heart this observation is true, in that we believe that for non-quantum interactions, Maxwell's equations provide a complete description of electromagnetic phenomena,[19] for many aspects of CEM modelling, one needs to be *extremely* cautious of such sentiments. The modelling process is about the art of acceptable approximation, and this path is strewn with pitfalls.

Firstly, we are replacing the real world with a mathematical model, or put differently, replacing a *real* field problem with an *approximate* one. Here are some examples of possible problems.

Limitations of the mathematical model Mathematical models of electromagnetic devices usually have some underlying assumptions. An example is the infinitely large planar ground assumed in a Sommerfeld formulation. Most integrated antennas radiate primarily on broadside, so the finite ground of a real antenna is usually not a problem. However, endfire integrated antennas (a Yagi, for instance, photoetched on a printed circuit board) radiate most strongly along the ground interface, and the main beam on endfire apparently disappears when a Sommerfeld code is used.

Tolerances Any engineered system has some measure of tolerances. Some are really of little concern; others impact directly on device performance. An example of the former is surface roughness of average dimensions far less than a wavelength; this usually has little impact on the operation of the system. An example of the latter are uncertainties in dielectric constant or overall device dimensions. For antennas relying on standing-wave operation (most wire antennas, microstrip

[19] By replacing the field vectors with operators, Maxwell's equations become quantum theoretically correct.

patch structures etc.) these translate directly into variations in resonant frequency. (Since such devices are usually quite narrowband, this can be highly problematic.)

Manufacturing deviations The device which is being simulated may differ subtly from what was designed and analyzed, in a more fundamental way than simply due to device tolerances. An example of a frequency selective surface with such a problem will be discussed in the next section in detail; there, an air inclusion was de-tuning the device, and measurements and computations stubbornly refused to agree until the problem was identified.

Simplifications in the formulation Currents flowing on thin wires are usually approximated by current filaments (in other words, wire thickness is ignored). This can cause problems when the wire thickness no longer justifies this assumption.

Once we have an acceptable approximate field problem, it will then be solved using an *approximate numerical* solution. Once again, there are many pitfalls in this next step.

Finite discretization This is usually the biggest single limitation on the accuracy of numerical techniques in electromagnetics. There are typically two different types of error which accompany this: one is *interpolation* error, and is the (in)ability of the basis functions to model the field locally; the other is *mesh dispersion* error (also sometimes called pollution error) which is cumulative error through the mesh.[20] Both can usually be controlled by refining the mesh. Unfortunately, the computational cost of especially three-dimensional modelling is such that this is not always practical.

Finite problem space Neither the FDTD method nor the FEM incorporates the radiation condition, and the mesh needs to be truncated at some point when a radiation or scattering problem is undertaken. Absorbing boundary conditions are widely used for this. After creating an (in)adequately refined mesh, this is probably the second largest source of error in FDTD and FEM computations; in reality, the problems are interwoven, since a poor mesh termination scheme requires a larger solution region, which in turn makes it difficult to ensure that the mesh is fine enough.

[20] The MoM does not suffer from mesh dispersion, only the differential equation based FDTD method and FEM do.

Numerical approximations FEM and MoM codes in particular require numerical integration. This is usually done using quadrature or cubature (multi-dimensional numerical integration), and if not carefully done, may easily result in poor performance, in particular for the MoM which involves the integration of nearly singular or singular fields.

Finite machine precision The infinite number of real (or complex) numbers are represented on a computer in a very finite fashion. Typically, in single precision, 4 bytes (i.e. 16 bits) is used to represent a real number; this gives some 5 significant digits of accuracy. (Double precision uses 8 bytes, and approximately doubles this.) For problems which are ill conditioned (that is, the solution depends rather drastically on small changes in input data) this may not be adequate. This is usually a less serious problem than the others.

1.8 Verification and validation

The discussion of the previous section leads directly to the issues of the verification and validation of code. One might define the former as ensuring that the code has correctly implemented the formulation, and the latter as checking that the formulation as implemented in the code produces results agreeing with reality. However, for users of codes in practice, the processes are integrated, especially since users cannot change commercial codes. Throughout this book, the necessity of validating and verifying code will be continually emphasized, but it is such an important topic in CEM that it deserves this section on its own.

There are several methods currently in widespread use.

Comparison with analytically computed solutions This was the classical approach taken by most of the early researchers in the field. Typically, radiation or scattering solutions involving canonical shapes (usually cylinders in two dimensions, and spheres in three dimensions) were used to compare results with those of the code. The problem with this is that it is a necessary, but not sufficient, condition for a code to be working correctly.

Comparison with approximate solutions Quite often, approximate solutions of electromagnetic problems are known from simplified models, which have usually been experimentally tested, or may even represent experimental data. Many antennas are a good example of this, with parameters such as gain often a design parameter. Comparison of computed results with these provides some reassurance that the code is in the correct "ballpark," although of course this is not a rigorous process.

Comparison with measurements In a sense, this is the most satisfying and convincing method. However, it is strewn with difficulties. Unlike CEM, where the basic tools have dropped enormously in price, radio-frequency measurement devices have remained expensive, and accurate measurements of radiation or scattering require an anechoic chamber. Making reliable and repeatable measurements is also both a science and an art, and usually requires considerable experience.

Comparison with other CEM codes This is a relatively recent innovation, prompted both by the availability of powerful general-purpose codes and the difficulty of obtaining reliable measured data. Once again, caution is required. This is one place where the difference between validation and verification can be significant: to give an example, validating a thin-wire code by comparison to another thin-wire code will not detect a fundamental problem with the thin-wire assumption. In general, this is most convincingly done by comparing results computed using codes implementing different formulations.

We will see examples of the use of all these techniques throughout this book.

1.8.1 An example: a frequency selective surface

The process of validating computations can sometimes lead to enhanced understanding of the device under test. An example is the following, originally presented in [18, 19] for a device called a frequency selective surface (FSS). There are various applications thereof: in this case, a bandpass radome was required. The structure consists of a slot cut in a metallic sheet, which transmits an incident wave when the frequency is such that the slot is resonant.

When an FSS is fabricated, a dielectric support is generally required, lowering the resonant frequency and complicating the analysis. An FDTD code, originally developed by the author, was used to simulate the dielectric support. However, initial results yielded a consistent offset in center frequency between measured and computed data, which was sufficiently large to be problematic. The usual FDTD checks – refining the mesh, and moving the absorbing boundary condition further away – did not solve the problem.

Eventually, the problem was traced to a very subtle manufacturing problem. When manufacturing a dielectric-supported FSS structure, the finite thickness of the metal screen can be surprisingly significant, whether a sandwiched or single-sided support is used; this results effectively in a slot. Although the slot is small in cross-section, the material filling it plays a significant role in the electromagnetic behavior of the device. An example of the slot forming the FSS element in a finite thickness conductor is illustrated in Fig. 1.1. This is a cross-section of a circular ring FSS element, with diameter 5.9055 mm, slot width 0.537 mm, and element spacing 10.738 mm.

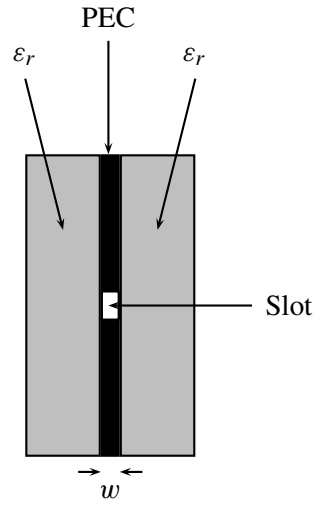


Figure 1.1 A cross-section of a slot forming the FSS in a conductor of *finite* thickness, w, sandwiched between two dielectrics. (Reproduced by permission of IEE from [18].)

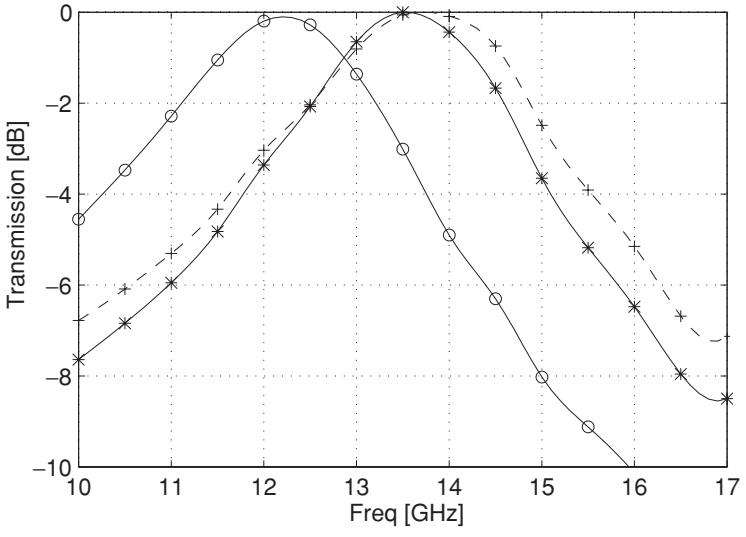


Figure 1.2 Predicted transmission coefficient of an O-ring FSS with one side perspex only. Legend: solid line (o), infinitely thin metal sheet, single-cell perspex in the slot; solid line (x), infinitely thin metal sheet, single-cell air in the slot; dashed line (+), actual 0.26845 mm thick metal sheet; air in the slot. (Reproduced by permission of IEE from [18].)

The FDTD code can accurately predict the effect of different dielectrics, provided the significance of this effect is realized and correctly modelled. Figure 1.2 shows this effect clearly; the resonant frequency is off by around 13% for the FDTD model which (incorrectly) assigned perspex to the slot. Two of the models in Fig. 1.2 used an infinitely thin metal sheet in the FDTD model; in one case

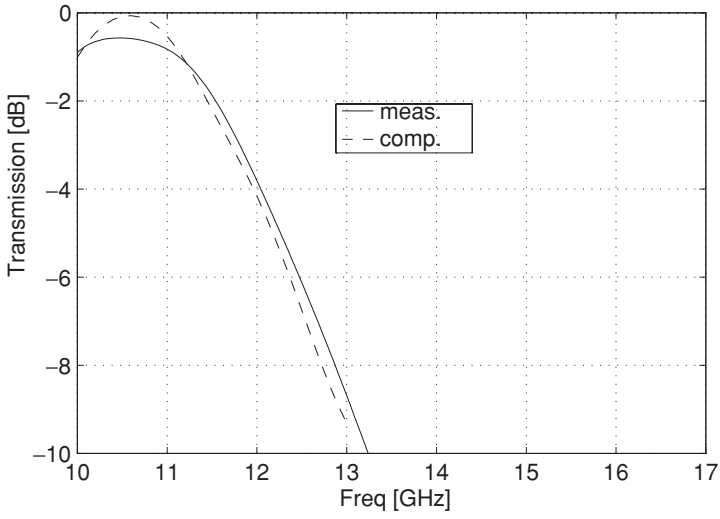


Figure 1.3 Transmission coefficient of PVC sandwiched FSS with petroleum jelly filling in the slot. The solid line is measured data, the broken line is the FDTD simulation. (Reproduced by permission of SAIEE from [19].)

perspex of a single FDTD cell thickness was used to model the cavity formed by the slot; in the other case air was assigned to the slot cavity – however, the depth of the actual slot was not entirely correctly simulated. The final model used the correct actual metal thickness and the slot was air filled, also to the correct thickness. The difference in predicted resonance frequency is significant.

Figure 1.3 shows measured and predicted results at normal incidence for a horizontal tri-slot sandwiched between PVC ($\epsilon_r = 2.86$), with petroleum jelly (Vaseline, $\epsilon_r = 2.16$) used to fill the slot which has been carefully modelled with the FDTD mesh. (This particular tri-slot had arm length 3.732 mm, arm width 1.0 mm, and inter-element spacing 12.5 mm.) The results demonstrate the accuracy achievable with careful modelling.

This is an example of a discrepancy between the real and approximate field problems, due in this case to a manufacturing problem. It is especially useful in that it led to improved understanding of the design, and a revised manufacturing process. CEM tools (the FDTD in this case) allowed very quick experimentation to establish that the air inclusion (in this case) was the problem; laboratory experiments with various prototypes would have been very tedious and time consuming indeed.

1.9 Extending the limits of full-wave CEM methods

It should be clear from the preceding sections that no one CEM method should lay claim to being able to address all problems with optimum efficiency. Both the FEM

and FDTD method are theoretically capable of addressing any arbitrary problem, but both can be unnecessarily or prohibitively expensive computationally for problems more suited to the MoM, and in practice of course *no* full-wave method still works at asymptotically high frequencies.

The computational cost of the major methods has been briefly reviewed in this chapter. Although the individual methods vary significantly, it is clear that in the case of *all* the full-wave solvers, as the frequency increases, so the mesh must become finer (and thus the number of unknowns become larger). We have seen that although the methods have different scaling properties, *all* of them scale badly with frequency. Even the most attractive scales at around the fifth to sixth power of frequency.[21] Much ingenuity has been devoted to developing new or modified full-wave methods with better scaling behavior – the present class of "fast methods" being an excellent case in point – and to exploiting high-performance computers, in particular parallel computing.

"Fast" methods, including the fast multipole method and the adaptive integral method, aim to reduce the *asymptotic* cost of the methods. Put very simply, the methods replace the traditional direct matrix solution algorithms with iterative solvers, and use methods to approximate the interaction between parts of the mesh which are separated by some reasonable distance (usually at least a few wavelengths). The matrix-vector product – which lies at the heart of iterative solvers – is implemented using a fast technique similar to the FFT, which reduces the cost from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$ per iteration. Recent work has claimed an asymptotic dependence of $\mathcal{O}(kd)^3$. This appears very attractive indeed, but one should be warned that this is an *asymptotic* calculation and there are some very large constants associated with this, as well as some possibly optimistic assumptions about rapid convergence of the iterative solver. Hence this attractive scaling behavior only manifests itself for electromagnetically very large problems. The convergence of iterative methods is also very problem dependent, so a particular analysis may not yield the expected asymptotic behavior if the solver should converge unexpectedly slowly. For an overview of recent progress on fast methods, see Chew *et al.*'s review paper [20]. We will discuss fast methods in Chapter 6.

Whilst great advances have been made, the full-wave techniques eventually make impossible demands on even the largest supercomputers, and asymptotic techniques become important. These methods generally use rays as field propagators, and essentially localize electromagnetic interaction, describing the field at a point as the sum of the direct, reflected, and various diffracted rays, all of which originate at points (or sometimes lines) on the structure. With these methods, there is no concept of discretization of an unknown *field* – although the surface may

[21] Assuming at least a surface discretization for the MoM.

well be approximated by facets.[22] These methods generally rely on the asymptotic nature of some underlying integral or series solution, and the approximation improves with frequency. An excellent overview of this may be found in [21]. Because the asymptotic techniques rely on approximations of the physics from the start, they do not lend themselves as well as the full-wave methods to general-purpose computer programs. However, when *hybridized* with the full-wave methods, some very significant extensions to the frequency range of full-wave codes become possible. Jakobus has made significant contributions recently using hybrid MoM/PO approaches [22, 23] and much of this work is reflected in the commercial program FEKO. Again, in Chapter 6, hybrid methods will be discussed in more detail.

To paraphrase Hafner [16], CEM is a field which depends not just on "big ideas," but also on getting lots of details right. This chapter has concentrated on the former, with the aim of providing the CEM beginner with some idea of what method is appropriate for what problem. Actually *implementing* a reliable CEM code makes enormous demands on the latter, and requires an on-going process of validation. One should be warned that even the most apparently straightforward method (the FDTD) is not as straightforward to implement as one might expect; development times for even the most specialized CEM codes involve at least months of work, and powerful, general-purpose codes involve many years of effort.

1.10 CEM: the future

CEM has passed through several phases: the 1960s and 1970s saw primarily work on CEM formulations; the 1980s saw the techniques starting to receive significant acceptance by non-specialists, and the 1990s saw the first widely available commercial codes for radio-frequency electromagnetic problems appear on the market. What does the next decade or so hold in store?

Firstly, it appears that we can look forward to continuing giant strides in computer performance. Looking back over the last decade, a typical PC has increased its clock-speed from some tens to megahertz to over a gigahertz, while memory sizes have grown from under one megabyte to hundreds of megabytes, and disk sizes have increased from ten or twenty megabytes to the same or more in gigabytes. (Workstations have also grown greatly in power, although their edge over top-end PCs is rather tenuous compared to the situation a decade ago.) This revolution in affordable computing has revolutionized potential CEM applications for engineers based in industry.

[22] In the case of physical optics (PO), the surface current is indeed discretized, but the amplitude of the current is *assumed* in terms of the known incident field, rather than being computed from a matrix equation enforcing a boundary condition.

CEM theory has also advanced enormously since the first work in the 1960s, and far more RF electromagnetic problems are potentially amenable to a CEM solution. Much work can be expected on hybridizing methods – the benefits of FEM/MoM and MoM/PO hybrids have been noted in this chapter, and will be discussed in more detail later in this book.

Intelligently refining meshes automatically is also an important topic, both in the research community and in commercial codes. Closely linked to this are methods for estimating errors in computed solutions; how this can be done will be briefly described in Chapter 10. Another significant trend is the incorporation of automatic optimizers using full-wave CEM tools for the analysis part of each iterative step in the optimization procedure. A number of commercial packages are starting to incorporate such abilities.

It can also be expected that the user interfaces will continue to improve, making modelling complex three-dimensional devices quicker and easier. Furthermore, it is notable that some commercial packages are starting to offer more than one method within the same graphical user interface. A point that has been made often in this chapter, and will continue to be made frequently in this book, is that one should chose the appropriate method for the problem at hand; working within a consistent user interface, it will be far easier for users to exploit the full power of the CEM techniques available.

Perhaps the most important trend will be the use of increasingly powerful commercial packages, and a decline in the number (or at least use) of CEM "freeware." This reflects both the difficulty (and hence expense) of developing general-purpose CEM packages; unless government sponsored (such as NEC), the cost of developing and maintaining code has to be recovered by licensing. Intimately connected to this, CEM developers should expect an increasing number of non-expert users of CEM tools (in much the same way that FEM analysis is now routinely taught to undergraduate civil and mechanical engineers, and routinely used in industrial design). Codes increasingly need to be robust, incorporating warnings of inappropriate meshing etc. for users without an extensive post-graduate training in electromagnetics. Electromagnetics remains a challenging discipline, and educating users of CEM tools, as well as making the tools more robust, will become increasingly important – it is hoped that this book will contribute to the former.

1.11 A "road map" of this book

This book comprises essentially three parts. The first part, Chapters 2 and 3, deals with the finite difference time domain method, in one and (primarily) two dimensions respectively. Chapter 2 uses a simple transmission line problem to introduce many of the basic ideas of the FDTD method. Chapter 3 goes on to extend

these ideas to two dimensions, and considers a number of the issues raised when handling radiation and scattering in free space, in particular the use of absorbing boundary conditions. In this context, an example is given of a perfectly matched layer. The three-dimensional FDTD method is briefly discussed, and examples of the use of a commercial three-dimensional code are presented. These two chapters form an integrated unit.

The second part, Chapters 4–8, deals with the method of moments. Here, the five chapters largely alternate theoretical devolopment with practical application. Chapters 4 and 5 form a unit, first introducing MoM theory for thin-wire antennas, and then applying it using both a commercial and a public domain code. Chapter 6, on modelling surfaces (and also volumes) using the MoM, is largely self contained. The material in Chapter 6 on the hybrid MoM/PO, as well as on high-performance computing and fast methods, could be omitted without interrupting the flow of the book on a first reading. Chapters 7 and 8 form a further unit on the theory and application of the Sommerfeld mixed potential integral equation approach to modelling stratified media (of which microstrip antennas are the most widely encountered application at radio frequencies). The material in Chapter 7 is amongst the most theoretically challenging in the book, and could be omitted or covered only superficially, whilst still allowing time for some of the examples in Chapter 8 to be studied. Similar comments also apply of course to readers in industry whose prime focus is on using the MoM.

The third and final part, Chapters 9 and 10, is devoted to the finite element method. Chapter 9 goes directly into two-dimensional vector element FEM theory; it is also used to illustrate the solution of an eigenvalue problem. The material in the last chapter, Chapter 10, is primarily to sensitize readers to more advanced formulations and applications (in this case, of the FEM).

For a course on CEM methods, there is probably more material here than can be covered in a typical semester course, and instructors can be guided by the above discussion regarding what to omit. Some suggested exercises are also included in Appendix D. They are intended primarily for use in a formal classroom environment, but would be useful for self-study as well.

Regarding the other appendices: good antenna and electromagnetic texts usually include material on vector calculus, and it is assumed that the reader has at least one, so repeating it here seems superfluous. Instead, the appendices contain material which the author has found useful specifically in CEM, and which is not easy to find in the literature.

A final comment. Electromagnetics, antenna engineering and microwave circuit design are all extremely well-established fields, with excellent textbooks available. This book is designed to complement, not compete, with them. It is a text specifically on the theory and applications of *computional* electromagnetics. It is

assumed that the reader has a suitable reference in his or her field of interest, so this book does not define antenna radiation patterns, $\mathcal{S}$-parameters or other well known and widely used concepts in this field. For readers who would like to add to their libraries, the following can be highly recommended. On electromagnetics in general, a very comprehensive reference is [24]; for antenna engineering, [25], [26] or [27] are all excellent references, as is [28] for microwave circuits and systems. There are many older texts which would also of course be suitable; the above are highlighted since they are all currently in print and have almost all been recently revised.

References

[1] S. Hawking, *A Brief History of Time*. London: Bantam, 1988.
[2] E. K. Miller, "A selective survey of computational electromagnetics," *IEEE Trans. Antennas Propag.*, **36**, 1281–1305, September 1988.
[3] P. J. Nahin, *Oliver Heaviside: Sage in Solitude*. New York: IEEE Press, 1988.
[4] R. S. Elliott, *Electromagnetics: History, Theory and Applications*. Piscataway, NJ: IEEE Press, 1993.
[5] H. A. Haus and J. R. Melcher, *Electromagnetic Fields and Energy*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[6] K. S. Kunz and R. J. Luebbers, *The Finite Difference Time Domain Method for Electromagnetics*. Boca Raton, FL: CRC Press, 1993.
[7] A. Taflove, *Computational Electrodynamics: The Finite Difference Time Domain Method*. Boston, MA: Artech House, 1995.
[8] A. Taflove and S. Hagness, *Computational Electrodynamics: The Finite Difference Time Domain Method*. Boston, MA: Artech House, 2nd edn, 2000.
[9] A. Taflove, *Advances in Computational Electrodynamics: the Finite Difference Time Domain Method*. Boston, MA: Artech House, 1998.
[10] J. Jin, *The Finite Element Method in Electromagnetics*. New York: Wiley, 2nd edn., 2002.
[11] P. P. Silvester and R. L. Ferrari, *Finite Elements for Electrical Engineers*. Cambridge: Cambridge University Press, 3rd edn., 1996.
[12] J. Volakis, A. Chatterjee, and L. Kempel, *Finite Element Method for Electromagnetics: Antennas, Microwave Cicuits and Scattering Applications*. Oxford and New York: Oxford University Press and IEEE Press, 1998.
[13] A. F. Peterson, S. L. Ray, and R. Mittra, *Computational Methods for Electromagnetics*. Oxford and New York: Oxford University Press and IEEE Press, 1998.
[14] P. P. Silvester and G. Pelosi, eds., *Finite Elements for Wave Electromagnetics*. New York: IEEE Press, 1994.
[15] T. Itoh, G. Pelosi, and P. P. Silvester, eds., *Finite Element Software for Microwave Engineering*. New York: Wiley, 1996.
[16] C. Hafner, *Post-Modern Electromagnetics: Using Intelligent Maxwell Solvers*. New York: Wiley, 1999.
[17] W. C. Chew, J. Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagentics*. Boston, MA: Artech House, 2001.

[18] D. B. Davidson, A. G. Smith, and J. J. van Tonder, "The analysis, measurement and design of frequency selective surfaces," in *Proceedings of the 10th International Conference on Antennas and Propagation*, vol. 1, pp. 1.156–1.160. Edinburgh: IEE, April 1997.
[19] D. B. Davidson, A. G. Smith, and J. J. van Tonder, "FDTD analysis and Gaussian beam measurement of frequency selective surfaces," *Trans. S. Afr. Inst. Electr. Eng.*, **88**, 72–81, September 1997.
[20] W. C. Chew, J. Jin, C. Lu, E. Michielssen, and J. M. Song, "Fast solution methods in electromagnetics," *IEEE Trans. Antennas Propag.*, **45**, 533–543, March 1997.
[21] P. H. Pathak, "High frequency techniques for antenna analysis," *Proc. IEEE*, **80**, 44–65, January 1992.
[22] U. Jakobus and F. M. Landstorfer, "Improved PO-MM hybrid formulation for scattering from three-dimensional perfectly conducting bodies of abritrary shape," *IEEE Trans. Antennas Propag.*, **43**, 162–169, February 1995.
[23] U. Jakobus and F. M. Landstorfer, "Improvement of the PO-MM hybrid method by accounting for effects of perfectly conducting wedges," *IEEE Trans. Antennas Propag.*, **43**, 1123–1129, October 1995.
[24] C. A. Balanis, *Advanced Engineering Electromagnetics*. New York: Wiley, 1989.
[25] C. A. Balanis, *Antenna Theory: Analysis and Design*. New York: Wiley, 2nd edn., 1997.
[26] W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design*. New York: Wiley, 2nd edn., 1998.
[27] J. D. Kraus and R. J. Marhefka, *Antennas for All Applications*. Boston, MA: McGraw-Hill, 3rd edn., 2002.
[28] D. M. Pozar, *Microwave Engineering*. New York: Wiley, 2nd edn., 1998.

2

The finite difference time domain method: a one-dimensional introduction

David B. Davidson and James T. Aberle

2.1 Introduction

The finite difference time domain method, usually referred to as the FDTD, is a particular implementation of a general class of methods known as finite difference techniques. The FDTD is so widely used in the CEM community that although finite difference methods cover a wide spectrum of complexity and accuracy, it is the FDTD which is almost always implied in CEM when finite differences are mentioned.

Finite difference methods are numerical methods in which derivatives are directly approximated by finite difference quotients. The general class of such methods is the most intuitive numerical approach, and was the first to be extensively developed by the scientific computing community. To this day, it probably remains the most universally applicable numerical technique and the one most widely used for scientific computation. As just discussed, for dynamic problems in CEM, the most popular is the FDTD method. The opening discussion in this chapter will discuss finite differences in general, before moving on to the specifics of the FDTD.

At this point, a general comment about the philosophy underlying the mathematical treatment of the computational algorithms in this book would be in order. Although we endeavor not to be "sloppy" mathematically, the emphasis in this book is in presenting well-known methods for well-known problems in CEM, rather than on the basic mathematical requirements of the methods, as one would expect to find in an applied mathematics text, for instance. An example of the type of issue which we will gloss over, at least initially, is the differentiation of discontinuous functions, which requires the generalized (weak) derivative, properly the field of functional analysis. Fortunately, the physics-based problems we are addressing usually do not evidence the type of pathological behavior which can (rightly so) concern mathematicians, and issues such as existence proofs will

generally be treated superficially, if at all, in this book. The reader should be cautioned about applying the methods discussed here in other fields of engineering or applied science without first mastering the underlying theory in more detail.

2.2 An overview of finite differences

2.2.1 The basic solution procedure

The basic steps of any finite difference method can be summarized as follows.

- Divide the solution region into a grid of nodes.
- Approximate derivatives in the given partial differential equation by finite differences involving the value of the solution at various nodes.
- Solve the finite difference equations for the value of the solution at each node subject to boundary and/or initial conditions. If operating in the time domain, this amounts to finding the values at the next time-step. This process is variously called time marching, time integration, or specifically in the context of the FDTD, "leap-frogging."

The FDTD method, being a time domain approach, is an initial value method (although material boundaries are of course included). Finite difference methods in general can operate as either boundary value or initial value methods.

2.2.2 Approximating derivatives using finite differences

Central to all finite difference methods is the approximation of derivatives with finite differences. From the basic definition of the derivative of a function, various numerical approximations can be proposed. However, these are usually derived from a Taylor series expansion, since this provides a handle on the error. Depending on whether the "next," "previous," or "central" nodes are involved, one obtains forward, backward or central differencing as follows:

Forward difference formula for first derivative

$$\frac{dU(x)}{dx} = \frac{U(x+\Delta x) - U(x)}{\Delta x} - \frac{(\Delta x)}{2}\frac{d^2U}{dx^2} + \mathcal{O}(\Delta x)^2 \qquad (2.1)$$

Backward difference formula for first derivative

$$\frac{dU(x)}{dx} = \frac{U(x) - U(x-\Delta x)}{\Delta x} + \frac{(\Delta x)}{2}\frac{d^2U}{dx^2} + \mathcal{O}(\Delta x)^2 \qquad (2.2)$$

Central difference formula for first derivative

$$\frac{dU\,(x)}{dx} = \frac{U\,(x+\Delta x)\,-\,U\,(x-\Delta x)}{2\Delta x} - \frac{(\Delta x)^2}{6}\frac{d^3U}{dx^3} + \mathcal{O}(\Delta x)^4 \tag{2.3}$$

These expressions are obtained by performing a Taylor series expansion of the function around x. Let us consider the derivation of the central difference formula. The Taylor series expansion about x_0, evaluated at $x_0 + \Delta x$, is:

$$U(x_0+\Delta x) = U(x_0) + \Delta x\frac{\partial U}{\partial x}\bigg|_{x=x_0} + \frac{(\Delta x)^2}{2}\frac{\partial^2 U}{\partial x^2}\bigg|_{x=x_0} + \frac{(\Delta x)^3}{6}\frac{\partial^3 U}{\partial x^3}\bigg|_{x=x_0} + \frac{(\Delta x)^4}{24}\frac{\partial^4 U}{\partial x^4}\bigg|_{x=\xi} \tag{2.4}$$

ξ is a point located in the interval $(x_0, x_0 + \Delta x)$. This can alternatively be written as

$$U(x_0+\Delta x) = U(x_0) + \Delta x\frac{\partial U}{\partial x}\bigg|_{x=x_0} + \frac{(\Delta x)^2}{2}\frac{\partial^2 U}{\partial x^2}\bigg|_{x=x_0} + \frac{(\Delta x)^3}{6}\frac{\partial^3 U}{\partial x^3}\bigg|_{x=x_0} + \frac{(\Delta x)^4}{24}\frac{\partial^4 U}{\partial x^4}\bigg|_{x=x_0} + \mathcal{O}(\Delta x)^5 \tag{2.5}$$

A similar expansion is performed about x_0, evaluated at $x_0 - \Delta x$:

$$U(x_0-\Delta x) = U(x_0) - \Delta x\frac{\partial U}{\partial x}\bigg|_{x=x_0} + \frac{(\Delta x)^2}{2}\frac{\partial^2 U}{\partial x^2}\bigg|_{x=x_0} - \frac{(\Delta x)^3}{6}\frac{\partial^3 U}{\partial x^3}\bigg|_{x=x_0} + \frac{(\Delta x)^4}{24}\frac{\partial^4 U}{\partial x^4}\bigg|_{x=x_0} + \mathcal{O}(\Delta x)^5 \tag{2.6}$$

Subtracting the two expressions, grouping terms, dividing by Δx and noting that the remaining terms in Δx cancel, we obtain Eq. (2.3).

A mathematical aside – finite difference approximations of the second derivative

If, instead of differencing Eqs. (2.5) and (2.6) as above, we add them, we obtain a formula for the central difference approximation of the *second* derivative of the function. The result, with remainder term of second order, is:

$$\frac{d^2U\,(x)}{dx^2} = \frac{U\,(x+\Delta x)\,-\,2U\,(x)\,+\,U\,(x-\Delta x)}{(\Delta x)^2} - \frac{(\Delta x)^2}{12}\frac{d^4U}{dx^4} + \mathcal{O}(\Delta x)^4 \tag{2.7}$$

(Note that the terms in $(\Delta x)^3$ also cancel.) In the FDTD, we will not directly use this formula, but it turns out that the FDTD scheme is also second-order accurate. The reason is that the central difference formula for the second derivative can also be derived by combining the expressions for the forward and backward derivatives to first order, which is what the FDTD effectively does.

Although Eqs. (2.1), (2.2) and (2.3) appear similar – indeed, the first part of each is identical – the remainder (error) terms are not. For both forward and backward differencing, the error is proportional to the cell length (Δx) (also known as a first-order scheme), but for central differencing, it is proportional to the square thereof, or alternatively, a second-order scheme.[1] Clearly, in the limit $\Delta x \to 0$, the central difference formula will converge more rapidly to the true value of derivative.

This idea of direct discretization of the derivatives underlies the FD method; one should rather view this as a class of methods, since there are a variety of choices which one can make with regard to the specific FD algorithm. Before moving onto the FDTD method, one last general point should be made: FD methods can be either *implicit* or *explicit*. This is particularly relevant when time is one of the variables. An *implicit* method requires the solution of a set of simultaneous equations – a matrix equation – in order to evaluate the unknowns. (The resulting matrix is generally highly *sparse*, i.e. has only a few non-zero entries. Efficient FD solvers exploit this to save both memory and computational time.) From a physics viewpoint, with an explicit method, the "next" value at a point is a function not only of the current and past values at this and the surrounding points, but also the "next" values of some or all of these. In an *explicit* method, each unknown can be obtained directly in terms of given or previously computed values. Physically, the next value is computed entirely from current or past values. Explicit methods do *not* require any matrix solution. However, they usually have some maximum time-step size, which if exceeded, produces instability (generally known as the *Courant limit*).

It should be noted that there are other methods for obtaining numerical derivatives. By using more points, higher-order schemes can be derived. However, the Yee scheme, to be discussed, does not readily accommodate these in general.

2.3 A very brief history of the FDTD

The FDTD is based on a particular FD scheme (Yee's algorithm) that is applied to Maxwell's curl equations in the time domain. It is an explicit marching-in-time procedure that simulates the propagation and interaction of electromagnetic waves

[1] A reminder: if a function $\sigma(x)$ is said to be $\mathcal{O}(x^n)$, then there exists some constant A such that $\sigma(x) < Ax^n$, $\forall x$.

in a region of space. At present, the FDTD is probably the most popular numerical method for the solution of RF electrodynamic problems, due to its simplicity and generality.

The algorithm was first proposed by Yee in 1966 [1]. For around a decade, the method attracted little, if any, attention; the computational electromagnetics community was primarily exploring the method of moments during this period. In 1975, Taflove and Brodwin obtained the correct stability criteria, and computed sinusoidal steady-state solutions using the method. In 1977, Holland, Kunz and Lee applied the method to electromagnetic pulse problems. In 1981, Mur obtained the first numerically stable, second-order accurate absorbing boundary conditions. From then on, the popularity of the method grew in leaps and bounds, as a number of theoretical issues were solved in rapid succession, culminating in Berenger's perfectly matched layer in 1994. The rapid adoption of the method was also due to the explosive growth in especially personal computing; in 1966, realistic applications of the FDTD made what were then outrageous demands on contemporary computers, whereas those of the MoM were decidedly more modest; by the 1990s, Moore's law had ensured that many realistic FDTD simulations could be undertaken on a PC in minutes or at most hours. Hence both theoretical developments and technological progress played crucial roles in the development of the method.

Theoretical work on the FDTD continues to this day, although the main thrust of most work is now in terms of applications. A detailed chronology, with extensive references, may be found in [2, Section 1.5].

2.4 A one-dimensional introduction to the FDTD

2.4.1 A one-dimensional model problem: a lossless transmission line

To introduce the FDTD algorithm, we will consider a lossless transmission line problem. From basic transmission line theory, the reader will be aware that for transverse electromagnetic (TEM) modes, there is a one-to-one correspondence between electric fields and voltage, and magnetic fields and current. Hence in the following, although we use voltage and current, this is fully equivalent to a field description of a TEM transmission line.

A reminder – TEM modes

As noted in the main text, for fields which are entirely transverse electromagnetic in nature, there is a one-to-one correspondence between electric fields and voltage, and magnetic fields and current. The best known example is a coaxial line. If the voltage between the inner (radius a) and outer (radius b) is V, then the

radial electric field is $\frac{V}{\ln(b/a)}\frac{1}{r}$, and with current I, the circumferential magnetic field is $I/2\pi r$. The simplest example is the parallel plate waveguide, separation d, at potential difference V, where the electric field is V/d and the surface current density and magnetic field are numerically equal, although orthogonal in space since $\vec{J}_s = \hat{n} \times \vec{H}$.

For guiding structures supporting more complex modes, such as TE or TM, a correspondence may still be found but it is no longer unique.

The well-known equivalent circuit of an infinitesimal piece of transmission line is shown in Fig. 2.1. L is the inductance per unit length and C is the capacitance per unit length of the lossless transmission line. On this section of line, the voltage and current on the line are described by a pair of coupled first-order differential equations, frequently known as the telegraphist's equations:

$$\frac{\partial I(z,t)}{\partial z} = -C\frac{\partial V(z,t)}{\partial t} \tag{2.8}$$

$$\frac{\partial V(z,t)}{\partial z} = -L\frac{\partial I(z,t)}{\partial t} \tag{2.9}$$

As already noted, the transmission line equations are a special case of Maxwell's curl equations in one dimension.

At this stage, we could decouple the differential equations to obtain the wave equation (which is a second-order partial differential equation) for the voltage on the line as a function of position and time. (This is the approach generally taken in introductory electromagnetics texts; the result is the one-dimensional wave equation, in either voltage or current.) However, we will instead work directly with the coupled pair of first-order equations.

Consider the following transmission line circuit problem, illustrated in Fig. 2.2. Assume $L = 1$ H/m, $C = 1$ F/m, $h = 0.25$ m, $R_S = 1\,\Omega$, and $R_L = 2\,\Omega$. (Note that

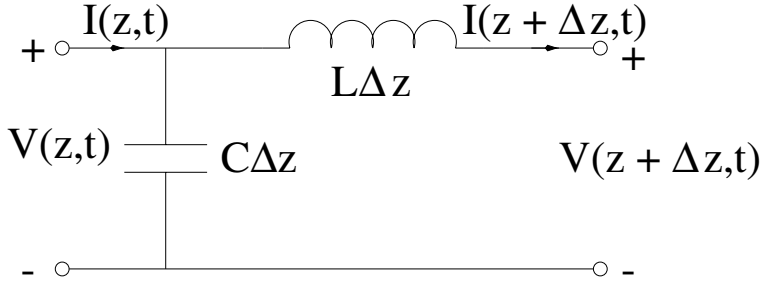


Figure 2.1 Infinitesimal section of a one-dimensional transmission line.

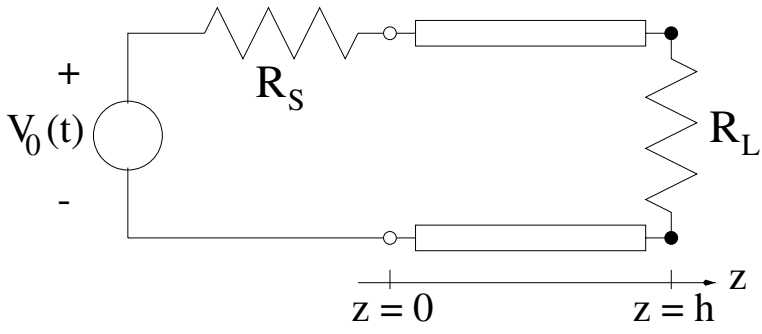


Figure 2.2 Model transmission line problem.

this choice of L and C produces a characteristic impedance of 1 Ω, and velocity of propagation of 1 m/s. Clearly, this is a normalized version of the actual problem; normalized equations such as these are quite frequently used in physics [3, 4].) The following, then, will be our model problem:

The model 1D problem

Determine the phasor representation of the steady-state response $V(z)$ versus z for

$$V_0(t) = \cos(8\pi t), \qquad t > 0 \tag{2.10}$$

The boundary conditions (BCs) at $z = 0$ and $z = h$ are:

$$V(0,t) = V_0(t) - R_S I(0,t) \tag{2.11}$$

$$V(h,t) = R_L I(h,t) \tag{2.12}$$

Take the initial conditions (ICs) to be:

$$V(z,0) = I(z,0) = 0 \tag{2.13}$$

A mathematical aside – classification of this problem

This problem is a deterministic, interior problem controlled by a hyperbolic partial differential equation, with mixed boundary conditions. It is deterministic since there is a source. It is an interior problem since the domain lies inside the boundaries. The vector wave equation is a hyperbolic partial differential equation, and the boundary conditions involve both voltage and current.

The exact solution can of course be readily derived used standard transmission line theory. (This is extremely useful, since it will permit us to test the accuracy of

our solution.) Noting that the source is matched, the result is:

$$V_{ss}(z) = V^+ \left(e^{-j\beta(z-h)} + \Gamma e^{j\beta(z-h)} \right) \tag{2.14}$$

where

$$V^+ = 1/2 \tag{2.15}$$

$$\Gamma = 1/3 \tag{2.16}$$

$$\beta = 8\pi \text{ rad/m} \tag{2.17}$$

Before moving on to the FDTD solution, it should be noted that the above solution is the *phasor* – i.e. frequency domain – solution of the problem. The excitation is a single-frequency sinusoid, radial frequency $\omega = 8\pi$ rad/s, or $f = 4$ Hz. Since the speed of propagation is 1 m/s, the phase constant/wavenumber is also 8π rad/m, and the wavelength is 1/4 m. The FDTD is a time domain solver, so we need to bear in mind that we are either going to have to convert the above solution into the time domain, or transform our FDTD solution into the frequency domain. The Fourier transform will of course provide the connection. We will also have to bear in mind that the above is the *steady-state* solution of the problem; there is also the transient part of the solution, which the FDTD solution is also going to include. We will discuss how to deal with these issues subsequently.

2.4.2 FDTD solution of the one-dimensional lossless transmission line problem

The first step in obtaining an FDTD solution is to set up a a regular grid in space and time. The points on this grid can be designated as (z_k, t_n) where

$$z_k = (k-1)\Delta z, \qquad k = 1, 2, \ldots, N_z \tag{2.18}$$

$$\Delta z = \frac{h}{N_z - 1}, \qquad N_z \geq 2 \tag{2.19}$$

$$t_n = (n-1)\Delta t, \qquad n = 1, 2, 3, \ldots \tag{2.20}$$

$$\Delta t = \frac{T}{M - 1}, \qquad M \geq 2 \tag{2.21}$$

As noted in the introductory remarks of this chapter, additional grid points at half-time and half-space points are now also introduced. These additional points can be designated as $(z_{k+1/2}, t_{n+1/2})$ where

$$z_{k+1/2} = \left(k - \frac{1}{2} \right) \Delta z, \qquad k = 1, 2, \ldots, N_z - 1 \tag{2.22}$$

$$t_{n+1/2} = \left(n - \frac{1}{2} \right) \Delta t, \qquad n = 1, 2, 3, \ldots \tag{2.23}$$

We shall compute $V(z,t)$ at the points (z_k, t_n), and $I(z,t)$ at the points $(z_{k+1/2}, t_{n+1/2})$, i.e. the voltage and currents are computed at offset locations in space *and also in time*. We now have two two-dimensional arrays representing the voltage and current. In each array, a row represents the temporal evolution of the field at a particular point in space, and a column represents the spatial distribution of the field at a particular point in time. (This is very convenient in understanding the method, but we should note now that the FDTD generally stores *only* two or three rows of each array – we will see subsequently how this is possible.)

To assist us in imposing the mixed BCs at $z = 0$ and $z = h$, two additional fictitious columns outside of the boundaries of the problem will be introduced, corresponding to:

$$z_{1/2} = -\frac{1}{2}\Delta z \tag{2.24}$$

$$z_{N_z+1/2} = h + \frac{1}{2}\Delta z \tag{2.25}$$

Similarly, to assist in the imposition of the the initial conditions at $t = 0$, an additional row will be introduced corresponding to:

$$t_{1/2} = -\frac{1}{2}\Delta t \tag{2.26}$$

Figure 2.3 shows these grid points graphically. The "$\circ$" indicates points at which $V(z,t)$ is computed, and "+" points at which $I(z,t)$ is computed. (As drawn here, vertical cuts correspond to temporal evolution at a particular point in space, horizontal cuts to spatial distribution at a particular point in time.)

The first transmission line equation, Eq. (2.8), is approximated at z_k and $t_{n+1/2}$ using central differencing in both space and time, i.e.,

$$\frac{\partial I\left(z_k, t_{n+1/2}\right)}{\partial z} \approx \frac{I_{k+1/2}^{n+1/2} - I_{k-1/2}^{n+1/2}}{(\Delta z)} \tag{2.27}$$

$$\frac{\partial V\left(z_k, t_{n+1/2}\right)}{\partial t} \approx \frac{V_k^{n+1} - V_k^n}{(\Delta t)} \tag{2.28}$$

Thus, the update equation for V may be obtained as

$$V_k^{n+1} = V_k^n - \frac{\Delta t}{C\Delta z}\left(I_{k+1/2}^{n+1/2} - I_{k-1/2}^{n+1/2}\right) \tag{2.29}$$

This update equation for V may be represented schematically by the "computational molecule" or "stencil" shown in Fig. 2.4. From this, it is clear that the update equation can be used for $k = 2, ..., N_z - 1$ and $n \geq 2$. Special update equations

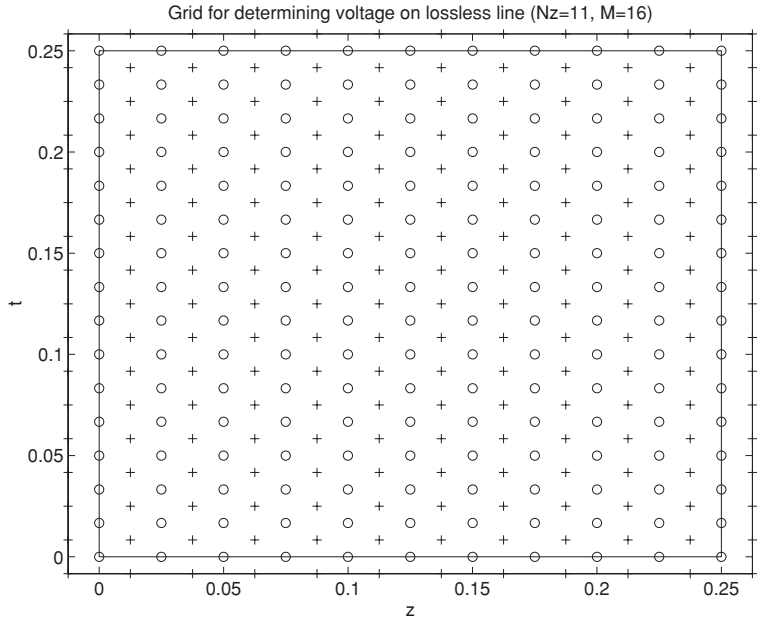


Figure 2.3 The Yee grid.

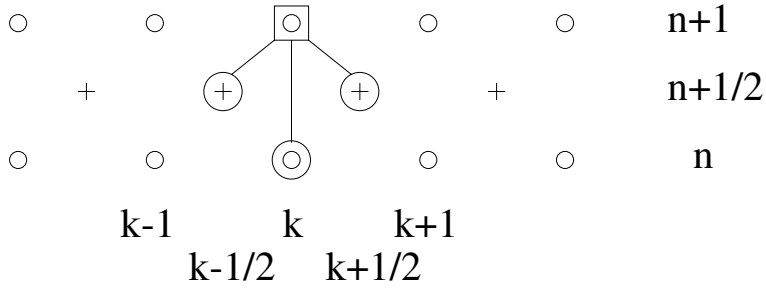


Figure 2.4 The voltage stencil.

must be devised from the initial and boundary conditions to treat $n = 1$ and $k = 1$ and $k = N_z$.

The second transmission line equation, Eq. (2.9), is approximated at $z_{k+1/2}$ and t_n using central differencing in both space and time, i.e.,

$$\frac{\partial V\left(z_{k+1/2}, t_n\right)}{\partial z} \approx \frac{V_{k+1}^n - V_k^n}{(\Delta z)} \tag{2.30}$$

$$\frac{\partial I\left(z_{k+1/2}, t_n\right)}{\partial t} \approx \frac{I_{k+1/2}^{n+1/2} - I_{k+1/2}^{n-1/2}}{(\Delta t)} \tag{2.31}$$

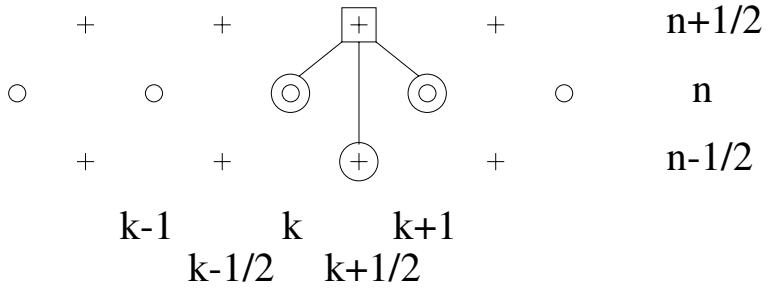


Figure 2.5 The current stencil.

Thus, the update equation for I may be obtained as

$$I_{k+1/2}^{n+1/2} = I_{k+1/2}^{n-1/2} - \frac{\Delta t}{L\Delta z}\left(V_{k+1}^{n} - V_{k}^{n}\right) \tag{2.32}$$

The update equation for I may be represented schematically as in Fig. 2.5. Again, it is clear that the update equation can be used for $k = 1, \ldots, N_z - 1$ and $n \geq 2$. Special update equations must be devised from the ICs to treat $n = 1$. However, no special treatment for the boundaries at $z = 0$ or $z = h$ needs to be implemented.

The update equation for V must be modified at $k = 1$ and $k = N_z$ to incorporate the BCs into the solution. Consider the update equation for V at $k = 1$ and $k = N_z$:

$$V_{1}^{n+1} = V_{1}^{n} - \frac{\Delta t}{C\Delta z}\left(I_{3/2}^{n+1/2} - I_{1/2}^{n+1/2}\right) \tag{2.33}$$

$$V_{N_z}^{n+1} = V_{N_z}^{n} - \frac{\Delta t}{C\Delta z}\left(I_{N_z+1/2}^{n+1/2} - I_{N_z-1/2}^{n+1/2}\right) \tag{2.34}$$

The values of $I_{1/2}^{n+1/2}$ and $I_{N_z+1/2}^{n+1/2}$ must be obtained from the BCs. Consider the BC at $z = 0$:

$$V(0, t_n) + R_S I(0, t_n) = V_0(t_n) \tag{2.35}$$

Using

$$V(0, t_n) = V_1^n \tag{2.36}$$

$$I(0, t_n) \approx \frac{1}{2}\left[I_{1/2}^{n+1/2} + I_{3/2}^{n+1/2}\right] \tag{2.37}$$

the discretized BC gives

$$I_{1/2}^{n+1/2} = -I_{3/2}^{n+1/2} - \frac{2}{R_S}V_1^n + \frac{2}{R_S}V_0(t_n) \tag{2.38}$$

A mathematical aside – semi-implicit approximations

Equation (2.38) is sometimes known as a semi-implicit approximation, since the "next" value at point $1/2$ also uses the "next" value at point $3/2$. This is also used when conduction currents are included in a full-wave solver. Although widely and successfully used, this approximation can degrade both the stability and accuracy of the solver.

Consider the BC at $z = h$:

$$V(h, t_n) - R_L I(h, t_n) = 0 \tag{2.39}$$

Using

$$V(h, t_n) = V_{N_z}^n \tag{2.40}$$

$$I(h, t_n) \approx \frac{1}{2}\left[I_{N_z-1/2}^{n+1/2} + I_{N_z+1/2}^{n+1/2}\right] \tag{2.41}$$

the discretized BC gives

$$I_{N_z+1/2}^{n+1/2} = -I_{N_z-1/2}^{n+1/2} + \frac{2}{R_L} V_{N_z}^n \tag{2.42}$$

Using the values of $I_{1/2}^{n+1/2}$ and $I_{N_z+1/2}^{n+1/2}$ derived from the BCs, we obtain the update equations for V at $k = 1$ and $k = N_z$:

$$V_1^{n+1} = \left(1 - \frac{2\Delta t}{R_S C \Delta z}\right) V_1^n - \frac{2\Delta t}{C\Delta z} I_{3/2}^{n+1/2} + \frac{2\Delta t}{R_S C \Delta z} V_0(t_n) \tag{2.43}$$

$$V_{N_z}^{n+1} = \left(1 - \frac{2\Delta t}{R_L C \Delta z}\right) V_{N_z}^n + \frac{2\Delta t}{C\Delta z} I_{N_z-1/2}^{n+1/2} \tag{2.44}$$

To start the FD scheme, we need to obtain the values of V_k^1 and $I_k^{3/2}$ for $k = 1, \ldots, N_z$.

The values of V_k^1 may be obtained from the initial condition $V(z, 0) = 0$. Hence, $V_k^1 = 0$ for $k = 1, \ldots, N_z$.

The update equation for I for $n = 1$ is:

$$I_{k+1/2}^{3/2} = I_{k+1/2}^{1/2} - \frac{\Delta t}{L\Delta z}\left(V_{k+1}^1 - V_k^1\right) = I_{k+1/2}^{1/2} \tag{2.45}$$

The value of $I_{k+1/2}^{1/2}$ must be obtained from the initial condition. Consider the initial condition:

$$I(z_{k+1/2}, 0) = 0 \tag{2.46}$$

Clearly, I must be zeroed at all points at both time-step $1/2$ and $3/2$.

Summary: FDTD scheme for the model problem

In summary, the FD scheme for this problem is:

$$V_k^1 = 0, \qquad \text{for } k = 1, \ldots, N_z \tag{2.47}$$

$$I_{k+1/2}^{3/2} = 0, \qquad \text{for } k = 1, \ldots, N_z - 1 \tag{2.48}$$

For $n \geq 2$,

$$V_1^n = \left(1 - \frac{2\Delta t}{R_S C \Delta z}\right) V_1^{n-1} - \frac{2\Delta t}{C\Delta z} I_{3/2}^{n-1/2} + \frac{2\Delta t}{R_S C \Delta z} V_0(t_{n-1}) \tag{2.49}$$

$$V_k^n = V_k^{n-1} - \frac{\Delta t}{C\Delta z}\left(I_{k+1/2}^{n-1/2} - I_{k-1/2}^{n-1/2}\right), \qquad \text{for } k = 2, \ldots, N_z - 1 \tag{2.50}$$

$$V_{N_z}^n = \left(1 - \frac{2\Delta t}{R_L C \Delta z}\right) V_{N_z}^{n-1} + \frac{2\Delta t}{C\Delta z} I_{N_z-1/2}^{n-1/2} \tag{2.51}$$

$$I_{k+1/2}^{n+1/2} = I_{k+1/2}^{n-1/2} - \frac{\Delta t}{L\Delta z}\left(V_{k+1}^n - V_k^n\right), \qquad \text{for } k = 1, \ldots, N_z - 1 \tag{2.52}$$

Programming aspects: avoiding half-steps

Half-integer values are inconvenient to program. To avoid them, we can simply make the following changes:

$$n + 1/2 \longrightarrow n \tag{2.53}$$

$$k + 1/2 \longrightarrow k \tag{2.54}$$

However, *this is only a matter of notational convenience, the voltages and currents are still located at the relevant points and times, with half-offsets as appropriate!* This must always be kept in mind. This also extends to both two- and three-dimensional FDTD solvers.

Programming aspects: "in-place" operations

A careful study of the update equations above shows that once all the next values of voltage (i.e. V_k^n) have been obtained, the current values $V_k^{(n-1)}$ are never needed

again. (The update equation for current, viz. $I_{k+1/2}^{n+1/2}$, requires only V_k^n.) As such, it is usual practice in an FDTD code to overwrite $V_k^{(n-1)}$ with V_k^n at the end of each time-step. Indeed, it can be done immediately on a point-by-point basis, but it is usually more convenient to do this all in one vector update. Hence only four vectors need be stored, two for voltage and two for current. If, for some reason, one wants the complete time history at a point (or plane or volume) then this is usually stored in a separate array. In signal processing, such operations are known as "in place" operations.

Programming aspects: reducing the "operation count"

It is also possible to reduce the number of operations per time-step (and reduce memory requirements in inhomogeneous problems) if the following change of variables is made:

$$\tilde{V}_k^n = \frac{C\Delta z}{\Delta t} V_k^n \tag{2.55}$$

The algorithm becomes:

$$\tilde{V}_k^1 = 0, \qquad \text{for } k = 1, \ldots, N_z \tag{2.56}$$

$$I_k^1 = 0, \qquad \text{for } k = 1, \ldots, N_z - 1 \tag{2.57}$$

For $n \geq 2$,

$$\tilde{V}_1^n = (1 - \beta_1)\, \tilde{V}_1^{n-1} - 2I_1^{n-1} + \frac{2}{R_S} V_0(t_{n-1}) \tag{2.58}$$

$$\tilde{V}_k^n = \tilde{V}_k^{n-1} - \left(I_k^{n-1} - I_{k-1}^{n-1}\right), \qquad \text{for } k = 2, \ldots, N_z - 1 \tag{2.59}$$

$$\tilde{V}_{N_z}^n = (1 - \beta_2)\, \tilde{V}_{N_z}^{n-1} + 2I_{N_z-1}^{n-1} \tag{2.60}$$

$$I_k^n = I_k^{n-1} - r\left(\tilde{V}_{k+1}^n - \tilde{V}_k^n\right), \qquad \text{for } k = 1, \ldots, N_z - 1 \tag{2.61}$$

$$\beta_1 = \frac{2\Delta t}{R_S C \Delta z} \tag{2.62}$$

$$\beta_2 = \frac{2\Delta t}{R_L C \Delta z} \tag{2.63}$$

$$r = \frac{(\Delta t)^2}{LC(\Delta z)^2} \tag{2.64}$$

Obtaining and evaluating preliminary results

As commented in the opening of this chapter, the FDTD computes results in the *time* domain. The analytical (phasor – steady state) solution is in the *frequency*

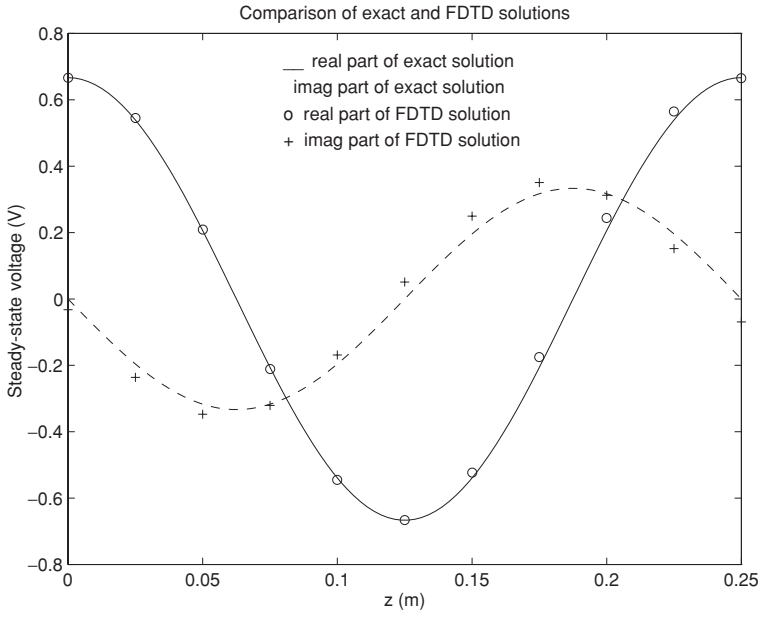


Figure 2.6 Solution for $N_z = 11$, $M = 64$, and $\epsilon = 0.002$.

domain. As is well known from circuit theory, the response of a system is the superposition of the transient and steady-state responses. By Fourier transforming the solution at different times, and noting the change in the solution, we can effectively eliminate the transient part of the solution. An estimate of the phasor representation of the steady-state response is obtained at the end of each period (of the sinusoid of 4 Hz frequency, i.e. every 250 ms) by evaluating the Fourier coefficient at frequency $\omega = 8\pi$ rad/s of the time domain data from the FDM solution. In computing the response, time domain data are stored for one period, and then overwritten. Steady state is taken to be achieved when the normalized RMS discrepancy between consecutive estimates is less than some positive error bound, i.e., $D_{\text{rms}} \leq \epsilon > 0$. As a measure of accuracy, we evaluate the normalized RMS error of Yee's algorithm with respect to the exact solution, viz. (E_{rms}). The result of this is given in Fig. 2.6. This particular solution required $N = 6$ for convergence with $E_{\text{rms}} = 0.0432$. Note that since this is a phasor, the result is of course complex, with both real and imaginary parts.

2.4.3 Accuracy, convergence, consistency and stability of the method

For any numerical method, important questions which one must pose include the following.

Accuracy The degree to which the numerical solution *to the approximate field problem* approximates the exact solution *to the approximate field problem*.

Consistency A finite difference equation is said to be *consistent* with a PDE provided that the local discretization error tends to zero as the mesh density increases, or alternatively, the mesh increment decreases. This is another statement of *convergence*: the numerical solution should converge to the exact solution as the mesh is refined (for this FDTD problem, this implies $\Delta z \to 0$).

Stability A process (e.g., a finite difference scheme) is said to be *stable* if and only if errors introduced at any stage in the process remain bounded throughout the entire evolution of the process.

The Lax equivalency theorem states that if a finite difference scheme is consistent with a properly posed[2] linear problem, then stability is the necessary and sufficient condition for convergence. (Stability proofs can fortunately be readily obtained.)

With regard to accuracy, we will first investigate this by numerical experimentation. In short, we are verifying the FDTD scheme. We have discussed this topic in Chapter 1; it is so important that further comments are in order at this point. Right at the start, we must stress that we can only meaningfully talk about accuracy of our numerical model with respect to the field problem which we posed – what we defined as the approximate field problem in Chapter 1. This problem is almost always a simplified version of the real-world problem. For instance, in our transmission line model, we assume no loss; no matter how good our FDTD solution, if the transmission line we are modelling has significant loss, our solution cannot be an accurate simulation of the real problem. Hence, to verify a numerical model, results are often compared to a known analytical solution of the same approximate field problem.

In practice of course, we want to use EM simulators to model the real world, and for this, comparison with measured data is highly desirable or even essential in many cases. Good agreement between measured data and numerically computed results indicates that all the important physics of the problem has been captured in the field description of the problem; the numerical approximation of the field problem is accurate and reliable; and (a point all too often overlooked) that reliable measurements on properly calibrated equipment have been made.

In Fig. 2.7, the effect of decreasing the size of time-step (or as plotted, by equivalently increasing the number of time points per period) is investigated. In Fig 2.8,

[2] A problem is said to be *properly posed* if: (i) a unique solution exists; (ii) the solution depends continuously on the initial and/or boundary conditions.

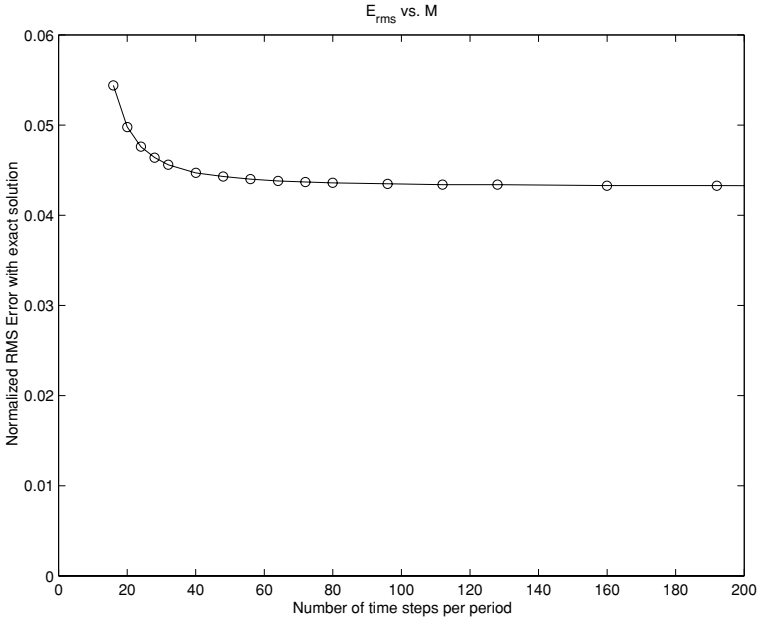


Figure 2.7 Normalized RMS error with exact solution versus number of time points per period (M) with $N_z = 11$. Note that Yee's algorithm is unstable in this case for $M < 16$.

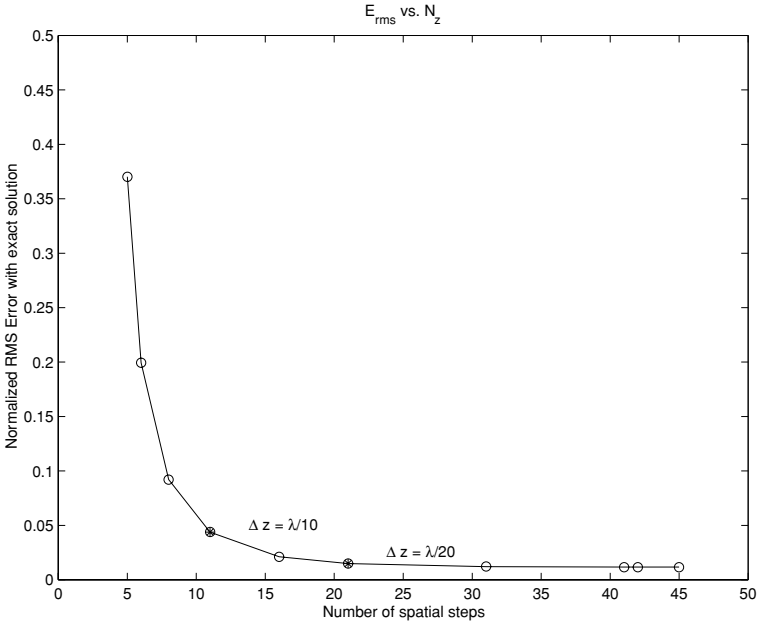


Figure 2.8 Normalized RMS error with exact solution versus number of spatial points (N_z) with $M = 64$. Note that Yee's algorithm is unstable in this case for $N_z > 45$.

the effect of decreasing the size of spatial step is investigated (again, the plot shows the equivalent effect of increasing the number of spatial points along the length of the line).

In both cases, one notes that stability imposes limits on the discretization. As already hinted at, with an explicit method such as the FDTD, there will be some maximum time-step size. As Fig. 2.7 implies, this will also be linked to spatial step size. The oft quoted stability criterion for Yee's algorithm in one, two, or three dimensions is

$$\frac{u\Delta t}{\Delta s} \leq \frac{1}{\sqrt{n}} \tag{2.65}$$

where Δs is the length of a side of a uniform cell and n is the number of space dimensions in the problem. For our one-dimensional problem, the above becomes $r \leq 1$ or

$$\frac{u\Delta t}{\Delta z} \leq 1 \tag{2.66}$$

where $u = 1/\sqrt{LC}$ is the velocity of propagation on the line.

The above stability criterion is also called the *Courant condition*, and it can be derived using Von Neumann's method applied to Yee's algorithm. The essential idea is to discretize a known plane wave in the algorithm, and require that its amplitude remain bounded as time-stepping progresses.

A physical interpretation of the *Courant condition* may be obtained by considering both the *numerical domain of dependence* and the *physical domain of dependence* for an arbitrary point in the grid. This is illustrated in Fig. 2.9. The region within the solid lines is the *numerical domain of dependence* and the region within the dashed lines is the *physical domain of dependence*. The solid lines have slopes of magnitude $\Delta t/\Delta z$ and the dashed lines have slopes of magnitude $1/u$. Yee's algorithm is stable provided that the physical domain of dependence is contained within the numerical domain of dependence. If this is not the case, then grid points outside of the numerical domain of dependence should be influencing the solution but cannot. Hence, instability is the result. The physical domain of dependence is contained within the numerical domain of dependence provided $1/u \geq \Delta t/\Delta z$, which is the Courant condition.

The Courant condition guarantees the stability of the basic update equations derived from the transmission line equations. However, it does not guarantee stability of the overall algorithm. Additional stability criteria exist for the update equations at the boundaries. Unfortunately, these are not usually known analytically, and numerical experimentation is often required. In practice, many FDTD simulations use either perfect electrical conductors (PECs) or absorbing boundary conditions on the exterior boundaries; the former simply zero the tangential fields, the latter

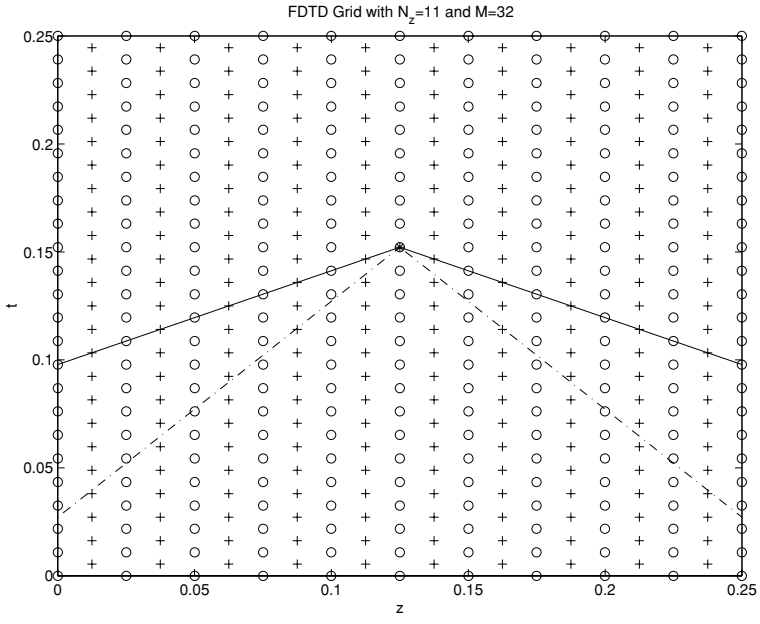


Figure 2.9 Physical interpretation of the Courant limit.

aim to match the interior wave properties as far as possible. Hence, this is generally not as serious a problem as this example might lead one to believe. Nonetheless, it is a point worth bearing in mind.

We have already seen that for our particular example, we experience instability for values of r greater than about $1/2$ (r is the fraction of the Courant limit; $r = 1$ implies one is at the limit). Consider the transmission line circuit problem shown in Fig. 2.10. Assume $L = 1$ H/m, $C = 1$ F/m, $h = 0.25$ m, and R_L is allowed to vary. Figure 2.11 shows the number of periods required for convergence of the solution and normalized RMS error with the exact solution versus reflection coefficient at the load.[3] Computations were made with $N_z = 11$, $M = 64$, and $\epsilon = 0.002$. The algorithm is found to be unstable for values of R_L equal to or less than about $0.15\ \Omega$, in spite of the very small value of $r = 0.0252$.

Further on the topic of stability, consider the normalized RMS error with the exact solution versus the number of periods used in the calculation for $R_L = 200\ \Omega$, as show in Fig. 2.12. Note that in the context of a $1\ \Omega$ system, this load is almost an open circuit. So-called "*late time instabilities*" in Yee's algorithm are rumored to manifest themselves when dealing with high Q structures – such as this example – that require a large number of time-steps for convergence to the steady state. These instabilities are usually attributed to the accumulation of round-off errors.

[3] For $R_L < 1\ \Omega$, the value of E_{rms} is normalized by dividing by the maximum value of voltage on the line.

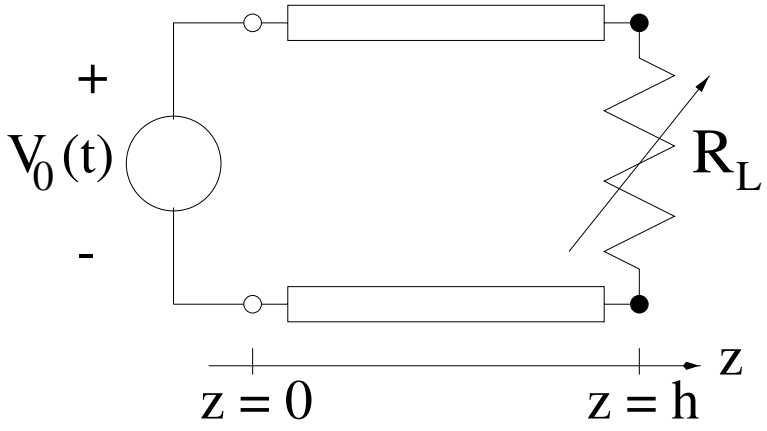


Figure 2.10 Transmission line circuit problem illustrating effect of load on stability.

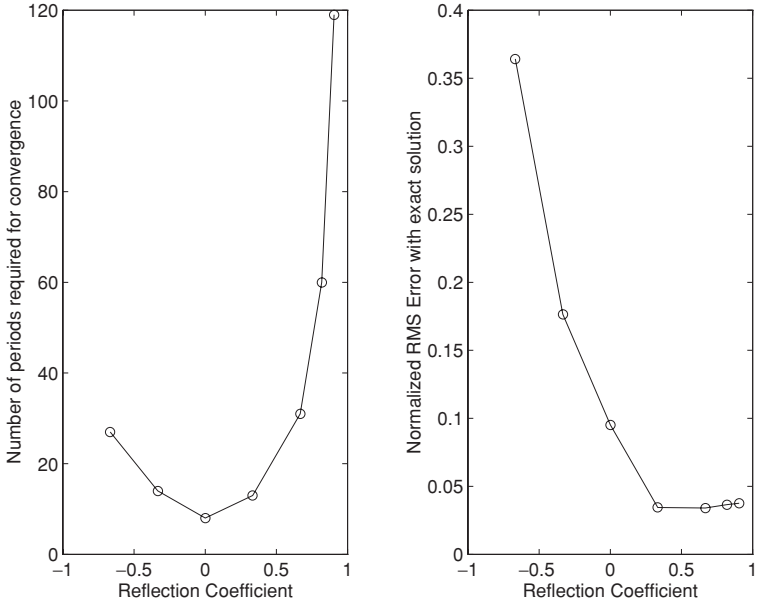


Figure 2.11 Number of periods required for convergence of the solution and normalized RMS error with the exact solution versus reflection coefficient at the load.

2.5 Obtaining wideband data using the FDTD

The transmission line example we have discussed follows the same historical path as the first FDTD work, by using a single-frequency excitation, waiting for the transients to die out, and then using the Fourier transform to give the frequency domain solution. It also connects elegantly with phasor circuit theory as taught worldwide at undergraduate level. However, this is a very inefficient use of the

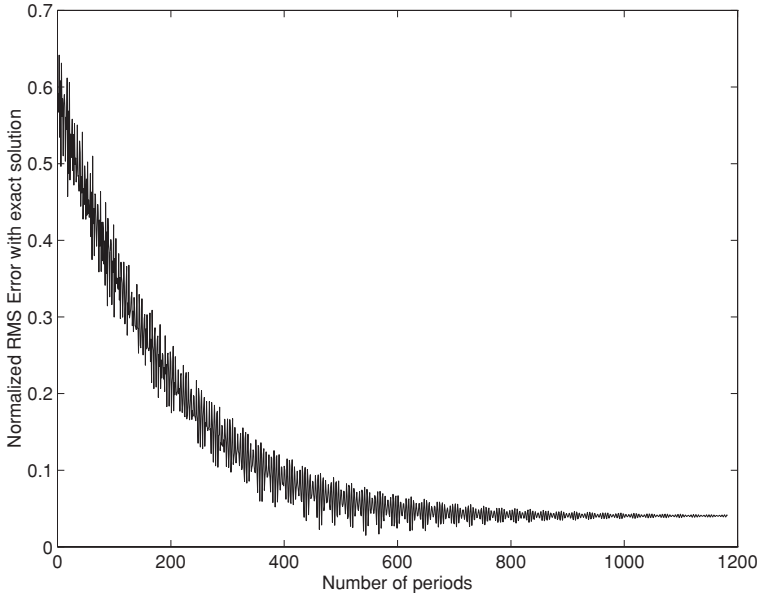


Figure 2.12 Illustration of late-time instabilities.

FDTD. The frequency spectrum associated with an excitation can directly produce the desired system response using some elementary concepts from system theory. Given an input signal and its $s(= j\omega)$ domain transform $x(t) \Leftrightarrow X(j\omega)$, a transfer function $h(t) \Leftrightarrow H(j\omega)$ and output signal $y(t) \Leftrightarrow Y(j\omega)$, we can find the transfer function as

$$H(j\omega) = \frac{Y(j\omega)}{X(j\omega)} \tag{2.67}$$

In introductory courses in circuit theory, one may have been asked to measure a transfer function in the laboratory, using a signal generator and an oscilloscope, with one channel monitoring the input and the other the output; in this case, $H(j\omega)$ has to be computed point by point across the required spectrum (a very painful process, not least since the signal generator needs to be continually re-set to a constant amplitude and phase as its frequency is changed, or these data must be noted for subsequent processing). What we have just done with our transmission line problem is the same, although done computationally. However, by using sources with more than just one frequency component, we can readily evaluate a number of points simultaneously. Ideally, we would like signal containing all possible frequencies (the Dirac delta function, of course, with spectrum $X(s) = 1$); for reasons we will appreciate shortly, this is neither practical nor desirable in real FDTD code (although it is possible in the very special case of a 1D code running at the "magic time-step," to be discussed subsequently).

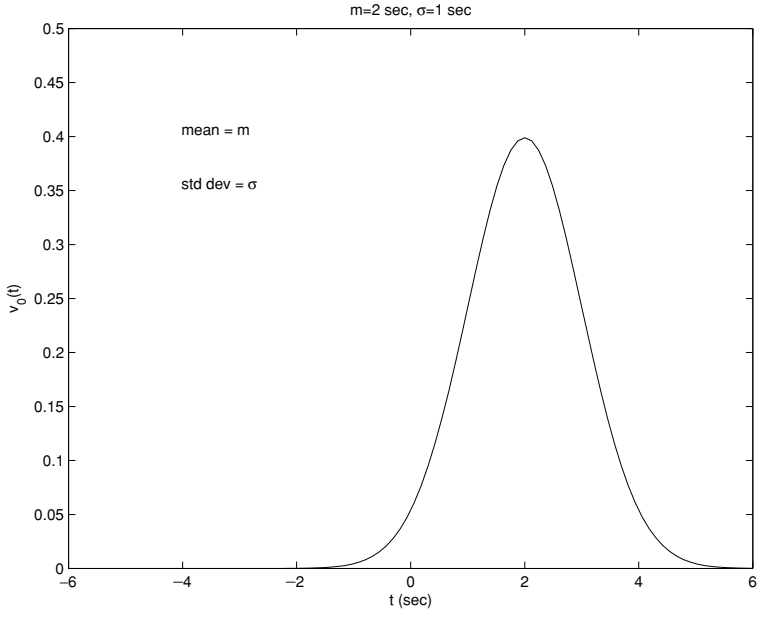


Figure 2.13 A Gaussian pulse.

Examples of wideband sources used in FDTD simulation include the following forms: Gaussian, Gaussian derivative, Rayleigh, chirp and wavelet pulses. The properties of the first two, perhaps the most popular in introductory FDTD work, are discussed in the following sections, as well as another interesting polynomial pulse.

2.5.1 The Gaussian pulse

The Gaussian pulse (Fig. 2.13) is popular in FDTD simulations:

$$v_0(t) = \frac{1}{\sqrt{2\pi}\sigma}\, e^{-(t-m)^2/2\sigma^2} \tag{2.68}$$

It has the advantage of having an analytically known spectrum – one of the peculiarities of the Fourier transform is that the spectrum of a Gaussian pulse is also a Gaussian (Fig. 2.14):

$$V_0(\omega) = e^{-j\omega m}\, e^{-\omega^2\sigma^2/2} \tag{2.69}$$

The energy contained in the pulse is also readily obtained:

$$\begin{aligned} E &= \int_{-\infty}^{\infty} v_0^2(t)\, dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} |V_0(\omega)|^2\, d\omega \\ &= \frac{1}{2\sigma\sqrt{\pi}} \end{aligned} \tag{2.70}$$

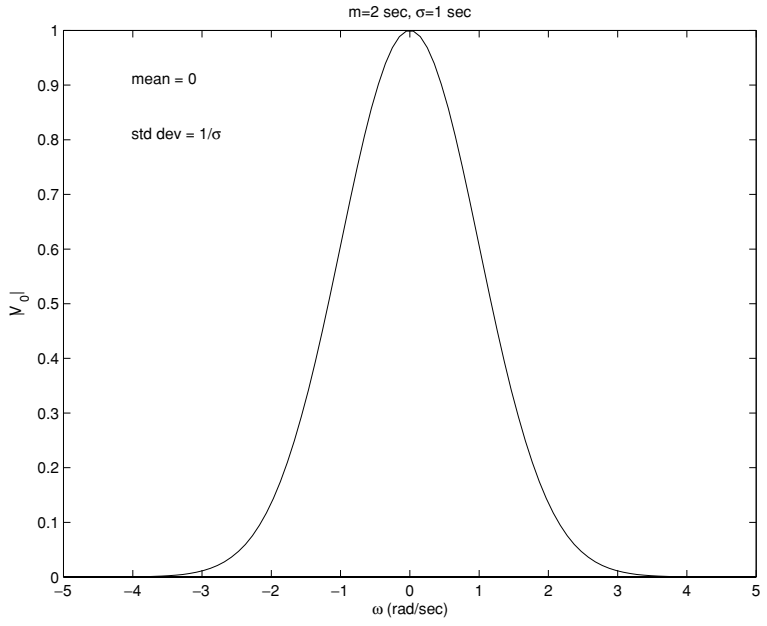


Figure 2.14 Spectrum of a Gaussian pulse.

However, the Gaussian pulse has some significant disadvantages. The most important are:

- it exists for all time, including $t < 0$,
- it has a strong frequency component at $\omega = 0$, i.e. DC.

The former requires that the pulse be windowed at some time (i.e. set to zero) which means there is a slight discontinuity of switch-on. The latter is a more subtle point; it turns out the static (DC) component can cause problems with charge build-up in FDTD grids[4] and it is better to avoid strong DC spectral components in FDTD simulations.

2.5.2 The Gaussian derivative pulse

A simple variant on the Gaussian, namely its derivative (Fig. 2.15), is also very popular in FDTD simulations, since it removes the DC component. It is defined as follows:

$$v_0(t) = \frac{-1}{\sqrt{2\pi}} \frac{(t-m)}{\sigma^3} e^{-(t-m)^2/2\sigma^2} \tag{2.71}$$

[4] Showing this is beyond the scope of this introductory discussion – for a detailed analysis, refer to [5].

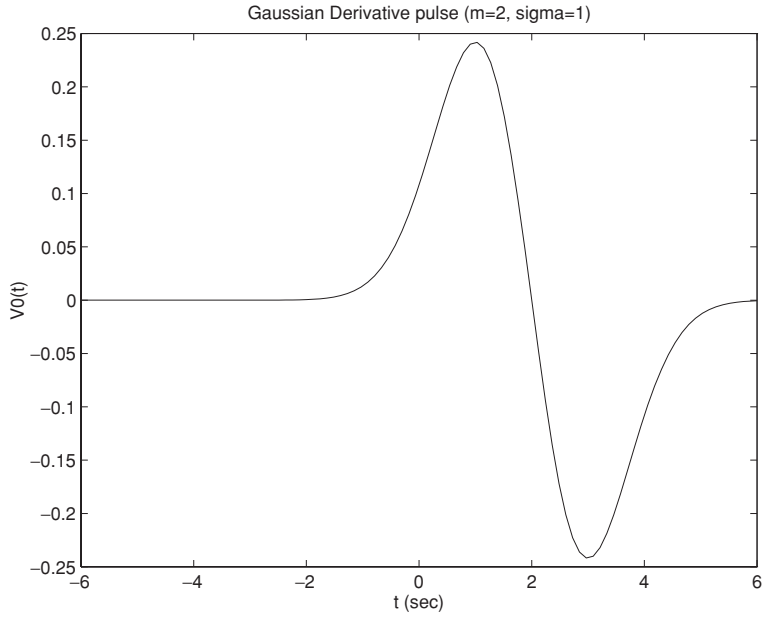


Figure 2.15 A Gaussian derivative pulse.

The spectrum of the Gaussian derivative pulse is (Fig. 2.16):

$$V_0(\omega) = j\omega e^{-j\omega m}\, e^{-\omega^2\sigma^2/2} \tag{2.72}$$

The energy of a Gaussian derivative pulse is also easily computed:

$$\begin{aligned} E &= \int_{-\infty}^{\infty} V_0^2(t)\, dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} |V_0(\omega)|^2\, d\omega \\ &= \frac{1}{4\sigma^3\sqrt{\pi}} \end{aligned} \tag{2.73}$$

2.5.3 A polynomial pulse

A pulse with finite support and interesting properties is the following, of quartic polynomial form:

$$f(t) = \begin{cases} (1-t^2)^4 & \forall |t| \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.74}$$

This pulse does not appear to have a specific name. Its derivative has the important

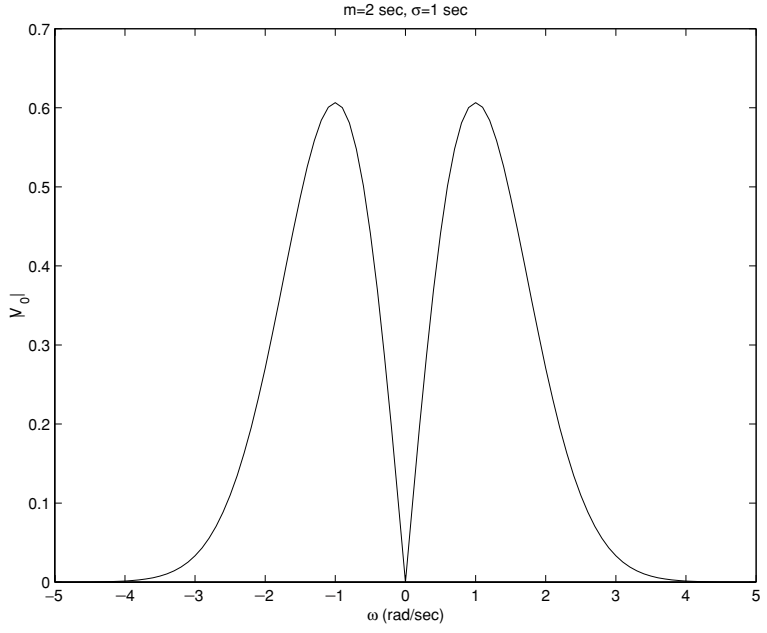


Figure 2.16 Spectrum of a Gaussian derivative pulse.

property of *also* being zero at $|t| = 1$:

$$\frac{df(t)}{dt} = \begin{cases} -8t(1-t^2)^3 & \forall |t| \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.75}$$

The first derivative has no DC content.

Interestingly, its *second* derivative has the *same* zero property at $|t| = 1$:

$$\frac{d^2f(t)}{dt^2} = \begin{cases} 48t^2(1-t^2)^2 - 8(1-t^2)^3 & \forall |t| \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.76}$$

Thus, the pulse has extremely smooth switch-on and switch-off characteristics, with the pulse and both its first and second derivatives all being zero at $|t| = 1$. These properties are clearly visible in Fig. 2.17. (Although not shown, this property even extends to the *third* derivative.)

By replacing t with

$$\tau(t) = 1 - 2(t/T) \tag{2.77}$$

in the above, a pulse is obtained with switch-on time $\tau = 0$ and duration T.

The Fourier transform of these pulses must be computed numerically.

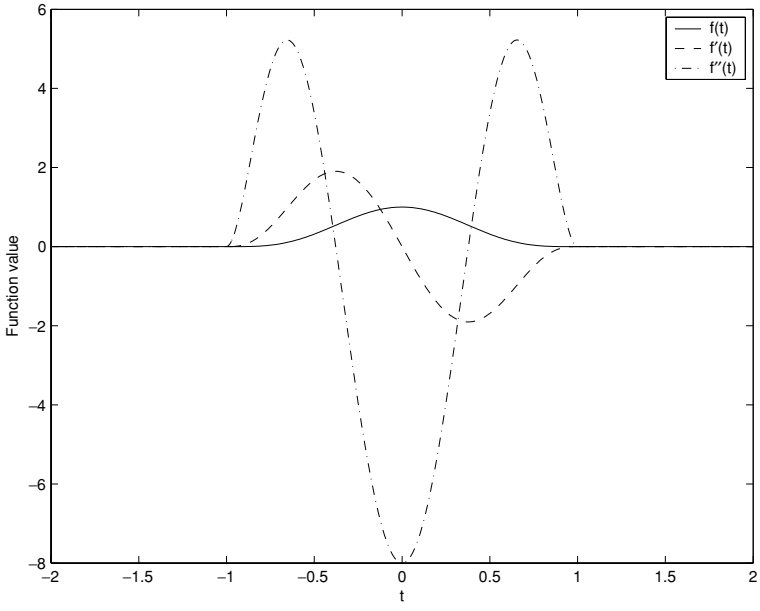


Figure 2.17 The $(1 - t^2)^4$ pulse, and its first and second derivatives.

This pulse is especially suitable for use as a windowed sinusoid (continuous wave):

$$F_{\mathrm{CW}} = [1 - \tau(t)^2]^4 \ \sin[m(2\pi t/T)] \tag{2.78}$$

with integer m controlling the number of cycles in the pulse. Clearly, $m = 1$ corresponds to one cycle only, since the windowing function is non-zero only in the interval $t = [0, T]$. An example of a ten cycle windowed sinsusoid is shown in Fig. 2.18.

This specific pulse, and its use as a window, appear to have been introduced in [6], although windowed sinusoids have been quite widely used in FDTD analysis.

2.5.4 The 1D transmission line revisited from a wideband perspective

We will now revisit our model 1D transmission line problem, and pose a slightly different question.

Find the frequency response $V_L(\omega)/V_0(\omega)$ of the transmission line circuit shown in Fig. 2.19 from $0 \leq \omega \leq 16\pi$ rad/s. Assume $L = 1$ H/m, $C = 1$ F/m, $h = 0.25$ m, $R_S = 0.5\,\Omega$, and $R_L = 2\,\Omega$.

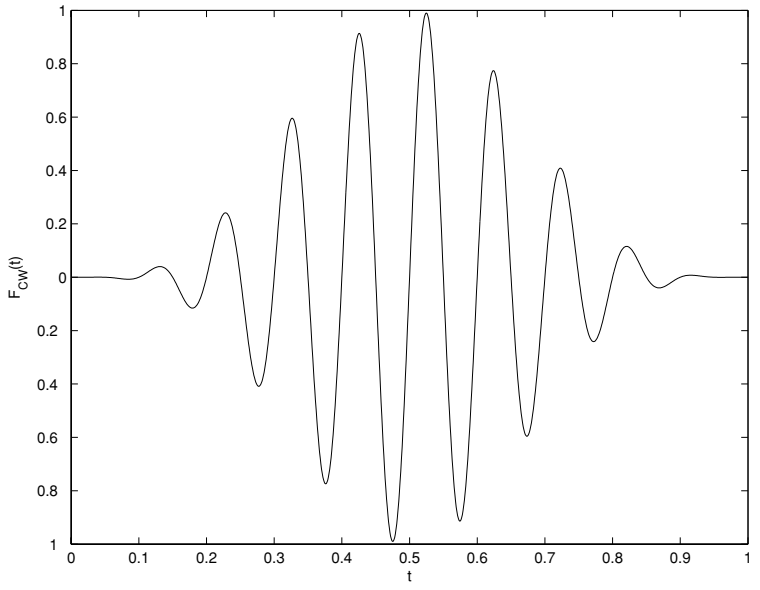


Figure 2.18 A ten cycle windowed sinusoid, defined over the interval [0, 1].

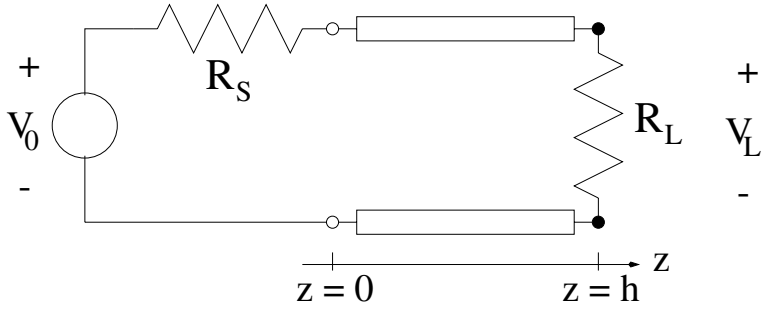


Figure 2.19 One-dimensional transmission line.

Using standard frequency domain transmission line analysis, we can obtain an exact solution for this problem as

$$\frac{V_L(\omega)}{V_0(\omega)} = \frac{Z_0}{Z_0 + R_S} \frac{1 + \Gamma_L}{e^{j\beta h} - \Gamma_L \Gamma_S e^{-j\beta h}} \tag{2.79}$$

where

$$\Gamma_L = \frac{R_L - Z_0}{R_L + Z_0} \tag{2.80}$$

$$\Gamma_S = \frac{R_S - Z_0}{R_S + Z_0} \tag{2.81}$$

$$Z_0 = \sqrt{\frac{L}{C}} \tag{2.82}$$

$$\beta = \omega\sqrt{LC} \tag{2.83}$$

To obtain the transfer function of the circuit using a Gaussian pulse source, we need to do the following.

- Set the bandwidth of the source to be wide enough to cover the frequencies of interest by choosing the standard deviation of the frequency spectrum of the source to be equal to $\omega_{\max}$, the maximum radian frequency of interest, i.e.

$$1/\sigma = \omega_{\max} \tag{2.84}$$

  In the results to be shown, $\omega_{\max} = 16\pi$ was used. This is sufficient to demonstrate the behavior of the transfer function over frequency, as well as the dispersive nature of the FDTD algorithm (more on this subsequently).
- Set the mean value of the time domain source function to be equal to four standard deviations so that the source can safely be assumed to be zero for $t \leq 0$, i.e.,

$$m = 4\sigma = 4/\omega_{\max} \tag{2.85}$$

- Choose a space step such that

$$\Delta z \leq \frac{\lambda_{\min}}{10} \tag{2.86}$$

  or equivalently,

$$\Delta z \leq \frac{\pi}{5\omega_{\max}\sqrt{LC}} \tag{2.87}$$

- Choose a time-step which satisfies both the stability criterion for Yee's algorithm (*Courant condition*) and the required *Nyquist sampling rate* for the highest frequency in the pulse.

$$\Delta t \leq \min\left(\Delta z\sqrt{LC}, \frac{\pi}{4\omega_{\max}}\right) \tag{2.88}$$

  where we assumed that the highest frequency in the pulse is $4\omega_{\max}$. (All finite-time sources have a theoretically infinite spectrum; we have to decide some reasonable upper limit on the spectrum. Recall that the Nyquist theorem states that a signal with maximum frequency content f_m must be sampled at at least twice this frequency, i.e. $\Delta t = 1/2f_m$, or $\Delta t = \pi/\omega_m$ in terms of radian frequency. In this case, we chose $\omega_m = 4\omega_{\max}$. Be careful not to confuse $\omega_{\max}$, the maximum frequency of interest, with ω_m, the maximum frequency present in the simulation!) Remember that the Courant condition does not guarantee the stability of the update equations at the boundaries, as the inequality in Eq. (2.88) reminds us.
- Use the FDTD update equations to let the system evolve during the source "on" time, which can be taken to be $0 \leq t \leq m + 4\sigma$. At the end of this time, compute the total

energy of the source as[5]

$$E_{\text{source}} = \int_0^{m+4\sigma} v_0^2(t)\,dt \tag{2.89}$$

Also, compute the Fourier transform of the response and the first estimate of its total energy as

$$E_L^{(1)} = \int_{-\infty}^{\infty} |V_L(\omega)|^2\,d\omega \tag{2.90}$$

- Allow the time evolution of the system to proceed. Periodically interrupt the time evolution to compute the Fourier transform of the response and a new estimate of the total energy of the response. Stop the time evolution of the system when the difference between the Kth and $(K-1)$th estimates of the total energy of the response normalized to the total energy of the source are less than or equal to some positive error bound, i.e.,

$$\frac{\left|E_L^{(K)} - E_L^{(K-1)}\right|}{E_{\text{source}}} \leq \epsilon > 0 \tag{2.91}$$

and the total energy of the response is greater than some small fraction of the total energy of the source.

2.5.5 Estimating the Fourier transform

The Fourier transform $X(\omega)$ of a time domain signal $x(t)$, for angular frequencies $\omega = 2\pi f$ is defined as

$$X(\omega) = \int_{-\infty}^{\infty} x(t)\,e^{-j\omega t}\,dt \tag{2.92}$$

and inverse transform

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(f)\,e^{j\omega t}\,d\omega \tag{2.93}$$

The pair are also often written as

$$X(f) = \int_{-\infty}^{\infty} X(t)\,e^{-j2\pi f t}\,dt \tag{2.94}$$

and inverse transform

$$x(t) = \int_{-\infty}^{\infty} X(f)\,e^{j2\pi f t}\,df \tag{2.95}$$

[5] This can be done conveniently in MATLAB using the `trapz` function.

We will approximate the Fourier transform using the discrete Fourier transform (DFT) defined by

$$X(k) = \sum_{n=1}^{N} x(n)\, e^{-j2\pi(k-1)(n-1)/N}, \qquad 1 \le k \le N \tag{2.96}$$

Signal processing experts sometimes view the two as entirely different transforms, and indeed, there are significant differences: the Fourier transform is defined for aperiodic signals, whereas the DFT automatically renders the signals periodic (at the Nyquist frequency); the Fourier transform is continuous, the DFT is discrete. However, we can very usefully approximate the Fourier transform with the DFT if we bear this in mind, ensure that we satisfy the sampling theorem and note that the DFT as defined above is missing the correct normalization. By replacing the infinite limits in Eq. (2.92) with 0 and $T = N\Delta t$, and then approximating the integral as a finite sum with $\Delta t = T/N$, we see that the DFT approximates the Fourier transform, but with a Δt scale factor missing, and also with the signal repeated with period T.

The DFT can be confusing when first used in this context, since the DC component is *not* in the middle as one might expect, but is rather the first component $k = 1$. Some definitions of the DFT include a $1/N$ scaling factor in the forward transform; other include this in the inverse transform. The DFT implementation in MATLAB (`fft`) uses the latter convention. The DFT yields N discrete frequency samples with a spacing $\Delta f = 1/T = 1/N\Delta t$. The number of samples N is usually taken to be a power of 2 (also sometimes known as radix-2) so that efficient algorithms, specifically the FFT, can be used to compute the DFT.[6] For an even number of samples N, the actual frequencies are defined as:

$$f_k = (k-1)\Delta f \tag{2.97}$$

for $k = 1, 2, \ldots, N/2$ and

$$f_k = (k-N-1)\Delta f \tag{2.98}$$

for $k = N/2+1, \ldots, N$ (the negative frequencies). The frequency at $k = N/2+1$ (which can equally validly be viewed as a positive frequency) $\mp\frac{N}{2}\Delta f$ is also known as the *folding frequency* or the *Nyquist frequency*, and the Fourier transform is symmetric about this.

An aside – gaining confidence with the DFT (and FFT)

Despite undergraduate exposure, the DFT can remain rather mysterious to many students. One way to gain confidence with the DFT is to Fourier transform

[6] In Chapter 6, the fast Fourier transform (FFT) is discussed in some detail.

simple signals, whose transforms are known. Consider a cosine signal of angular frequency 1 rad/s. Its period is 2π s, and its frequency $1/(2\pi)$ Hz. (From elementary courses on signal theory, it will be recalled that its Fourier transform is $\pi\{\delta(\omega+1)+\delta(\omega-1)\}$.) Let us take eight samples over one period (remember that we must take more than two to satisfy the sampling theorem!). These should be equally spaced from $t=0$ to $t=(7/8)2\pi$. (Including the sample at $t=2\pi$ would be incorrect, since this point has already been included at $t=0$.) This can be achieved very simply in MATLAB by using the command:

```
t=linspace(0,(7/8)*2*pi,8).
```

Now we create the cosine signal:

```
x=cos(t)
```

and apply the DFT (implemented as the FFT) to this:

```
X=fft(x).
```

The result is the following vector:

```
X=[0 4 0 0 0 0 0 4]
```

Inserting the $\Delta t = T/N$ scale factor, with $T=2\pi$ and $N=8$ in this case, which MATLAB omits, this vector is

```
X=pi*[0 1 0 0 0 0 0 1]
```

and we immediately recognize the positive frequency component $X(k=1)$ at $f_1 = 1/(2\pi)$, and negative frequency component $X(k=8)$ at $f_8 = -1/(2\pi)$. (Note that the FFT is complex, but by choosing a signal with even time symmetry, only the real parts of the Fourier transform are non-zero.)

Since most of our applications of the Fourier transform will be in computing *ratios* of spectra, the constants are not of great concern, but should be included for completeness.

The highest (non-aliased) frequency in the spectrum produced by the FFT is $F_{\max} = 1/2\Delta t$ and the frequency points are spaced by $\Delta f = 1/N\Delta t$. Additional frequency points (i.e., smaller values of Δf) can be obtained by *zero-padding* of the time domain data. The spectrum obtained by zero-padding of the time domain data is equivalent to that obtained by sinc-interpolation of the frequency domain data. (As an aside, we note that zero-padding to improve frequency resolution is a questionable practice, since no additional real data have been added to the system.)

To compare the FDTD solution to the exact solution, define the normalized RMS error with respect to the exact solution as

$$E_{\text{rms}} = \sqrt{\frac{1}{N}\sum_r \left| \frac{V_L^{(\text{FDTD})}(\omega_r)}{V_0^{(\text{FDTD})}(\omega_r)} - \frac{V_L^{(\text{exact})}(\omega_r)}{V_0^{(\text{exact})}(\omega_r)} \right|^2} \tag{2.99}$$

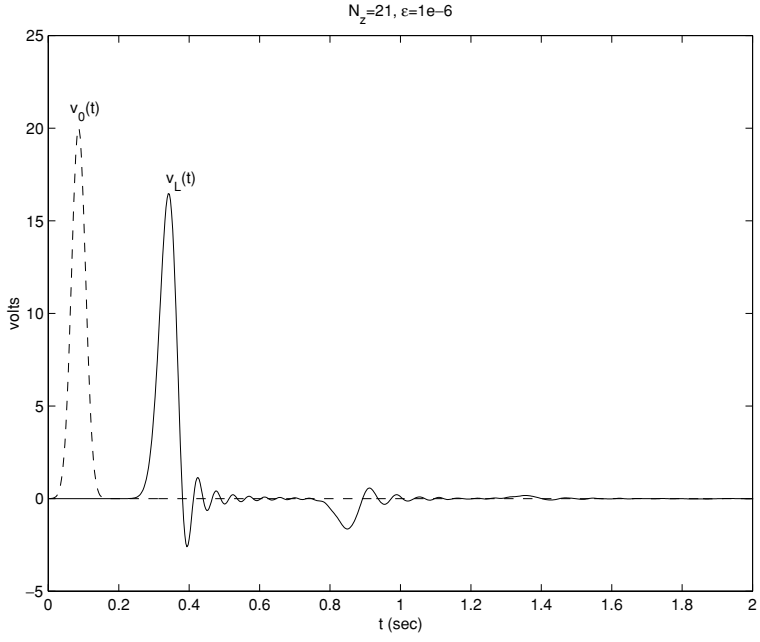


Figure 2.20 Generator and load voltages in the time domain for a Gaussian pulse.

2.5.6 Simulation using Gaussian and Gaussian derivative pulses

The FDTD solution with Gaussian pulse excitation required 694 total time-steps for convergence and results in $E_{\text{rms}} = 0.189$. Results are shown in Figs. 2.20, 2.21 and 2.22. The "ringing" in Fig. 2.20 is characteristic in telecommunications theory of a wideband signal on a dispersive channel, and we will see shortly that the FDTD indeed has dispersive properties.

The FDTD solution with Gaussian derivative pulse excitation requires 820 total time-steps for convergence and results in $E_{\text{rms}} = 0.190$. The generator and load voltages in the time domain for a Gaussian derivative pulse are shown in Fig. 2.23.

2.6 Numerical dispersion in FDTD simulations

2.6.1 Dispersion

Dispersion is the phenomenon of signal distortion caused by the dependence of phase velocity (v_p) on frequency. In a *dispersive medium*, either ϵ or μ or both are frequency dependent. The resulting dispersion is called *natural dispersion*. In general, *normal dispersion* occurs when $dv_p/d\omega < 0$ and *anomalous dispersion* occurs when $dv_p/d\omega > 0$. Numerical solutions (such as the FDTD) can also introduce *numerical* dispersion – we will return to this shortly.

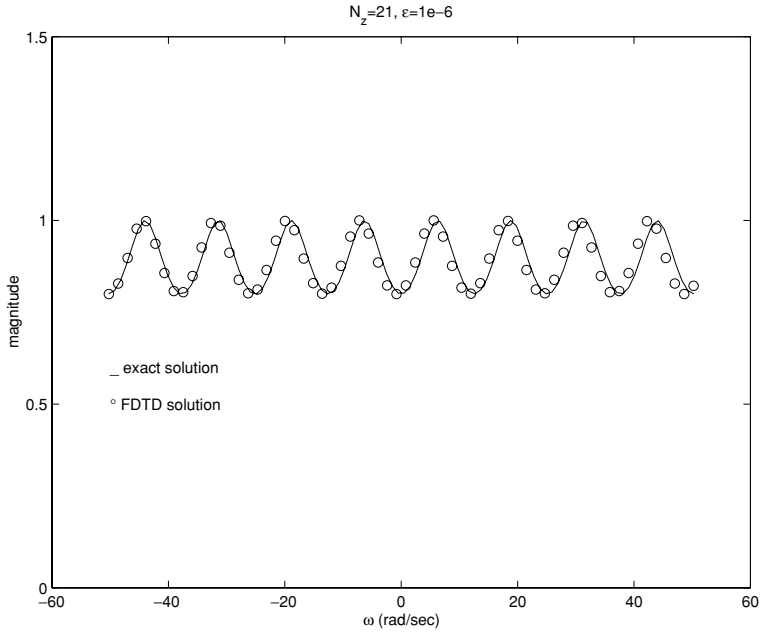


Figure 2.21 Magnitude of the transfer functions for exact and FDTD solutions.

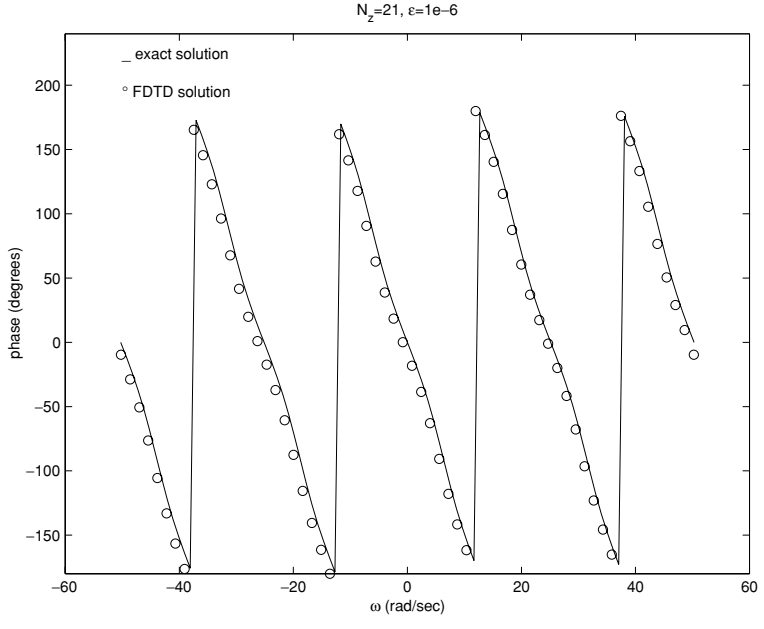


Figure 2.22 Phase of the transfer functions for exact and FDTD solutions.

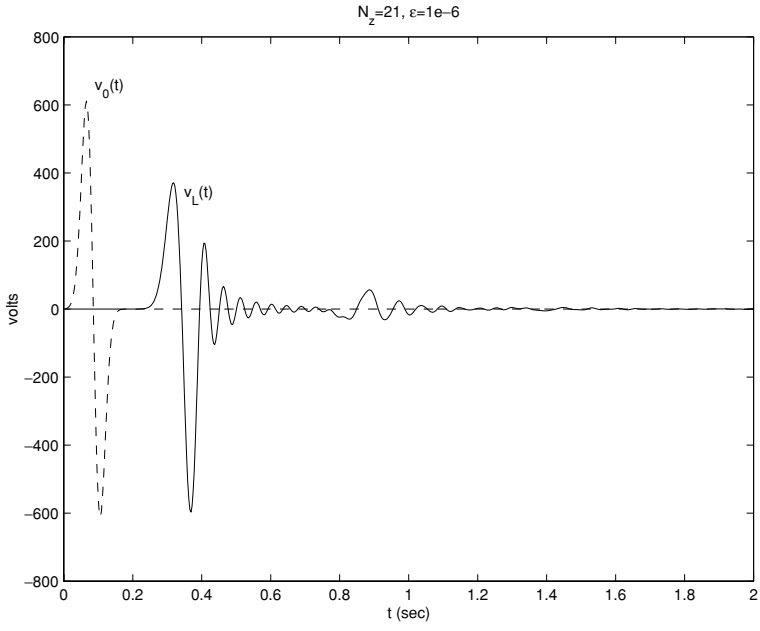


Figure 2.23 Generator and load voltages for a Gaussian derivative pulse.

The phase velocity in a medium is given by:

$$v_p = \frac{\omega}{\beta} \tag{2.100}$$

A *dispersion relation* describes the relationship between β and ω. In a *distortionless medium*, β is a linear function of ω and hence the phase velocity is constant with frequency. The *signal (group) velocity* is the velocity with which the signal (i.e., information) moves. It is the signal velocity which can never exceed the speed of light in a vacuum. The signal velocity can be computed as:

$$v_g = \frac{h}{T_d} \tag{2.101}$$

where T_d is the delay time experienced by the signal in traveling over the distance h.

Assume that the signal at $z = 0$ is given by $v_0(t)$, and that the signal at $z = h$ is given by $v_L(t)$. The delay time in traveling from $z = 0$ to $z = h$ (T_d) is the value of τ which maximizes the cross-correlation between $v_L(t)$ and $v_0(t)$,

$$\chi_L(\tau) = \int_{-\infty}^{\infty} v_L(t)\, v_0(t-\tau)\, dt \tag{2.102}$$

In a distortionless medium, the group velocity is equal to the phase velocity.

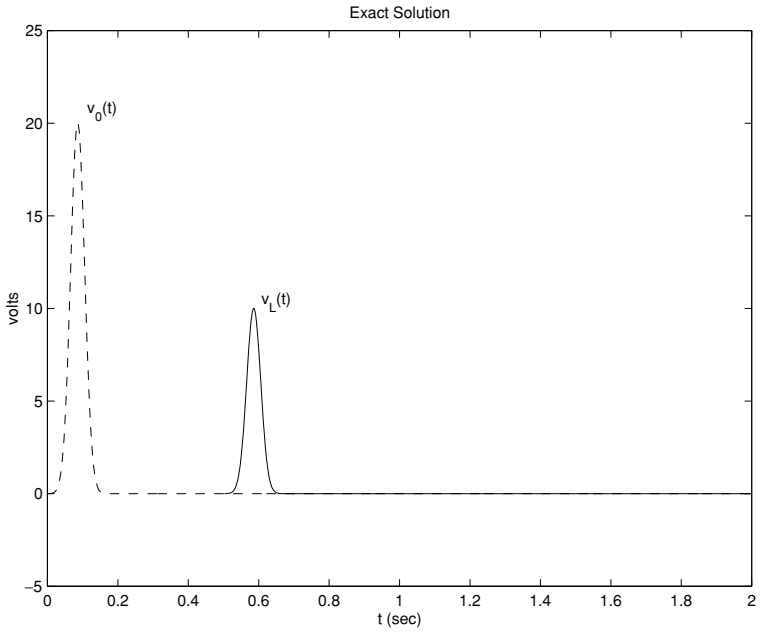


Figure 2.24 Theoretical generator and load voltages in the time domain for a Gaussian pulse.

A numerical algorithm can introduce *numerical dispersion*, even when waves are propagating in a distortionless medium. Yee's FDTD algorithm causes numerical dispersion. We will illustrate this by comparing theoretical and FDTD results for our simple transmission line circuit shown in Fig. 2.19. Assume $L = 1$ H/m, $C = 1$ F/m, $h = 0.5$ m, $R_S = 1\,\Omega$, and $R_L = 1\,\Omega$. This transmission line is distortionless with $\beta = \omega\sqrt{LC}$. The phase velocity on the line (v_p) is a constant versus frequency and is equal to 1 m/s. The source and load impedances are equal to the characteristic impedance of the line. Hence, there are no reflections at either end of the line. The theoretical generator and load voltages in the time domain for the following Gaussian pulse excitation are shown in Fig. 2.24

$$v_L(t) = \frac{1}{2} v_0\,(t - 0.5) \tag{2.103}$$

Compare these with the results computed in the time domain using the FDTD shown in Fig. 2.25. The ringing clearly visible on the load voltage is the result of numerical dispersion.

2.6.2 Derivation of the dispersion equation

To obtain the numerical dispersion relation resulting from Yee's algorithm, we assume monochromatic plane-wave trial solutions. Substituting these trial solutions

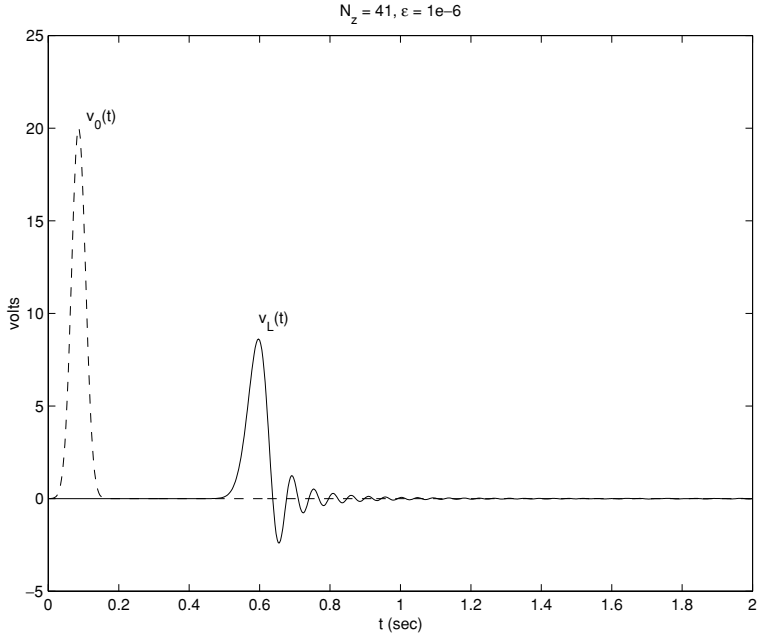


Figure 2.25 FDTD generator and load voltages in the time domain for a Gaussian pulse.

into the update equations and performing some straightforward algebraic manipulations yields the numerical dispersion relation. The procedure is as follows.

Firstly, assume trial solutions of plane-wave form. In continuous space-time, a z-propagating plane wave has the form $e^{j\omega t}e^{-j\beta z}$. In discretized form, and allowing for arbitrary amplitude, this becomes:

$$V_k^n = A\, e^{j\omega n\Delta t}\, e^{-j\beta k\Delta z} \tag{2.104}$$

A similar equation can be written for the discretized current:

$$I_k^n = B\, e^{j\omega(n+1/2)\Delta t}\, e^{-j\beta(k+1/2)\Delta z} \tag{2.105}$$

noting the offset between voltage and current.

These are now substituted into the update equations (2.29) and (2.32) to obtain the expression for the *next* time-step:

$$V_k^{n+1} = A\, e^{j\omega n\Delta t}\, e^{-j\beta k\Delta z} - \frac{\Delta t\beta}{c\Delta z} \times \left(e^{j\omega n\Delta t}\, e^{j\omega\Delta t/2}[e^{-j\beta(k+1/2)\Delta z} - e^{-j\beta(k-1/2)\Delta z}]\right) \tag{2.106}$$

Obviously, the last exponential term can be simplified as a sinusoid.

The *crucial* step in the derivation is to recognize that the discretized plane wave can *also* be written as:

$$V_k^{n+1} = A\, e^{j\omega(n+1)\Delta t}\, e^{-j\omega k \Delta z} \tag{2.107}$$

Since these two equations represent the same wave (albeit via the FDTD update and the analytical solution respectively), we can equate them. Thus, equating Eqs. (2.106) and (2.107), noting that for a plane wave the ratio of voltage to current is $Z_0 = \sqrt{L/C}$, and simplifying Eq. (2.106), we obtain the *dispersion equation*:

$$\sin\left(\frac{\omega\Delta t}{2}\right) - \frac{\Delta t}{\sqrt{LC}\Delta z}\sin\left(\frac{\beta\Delta z}{2}\right) = 0 \tag{2.108}$$

In the limit as $\Delta z \to 0$ (and thus, from the Courant limit, $\Delta t \to 0$), the small argument approximation (Taylor series expansion) of the sine function can be applied, and the expression becomes the exact (dispersionless) relation for the transmission line. This is important, because it indicates that *dispersion in an FDTD mesh can be controlled by making the mesh sufficiently fine.* This is a general result, and applies in 2D and 3D (although the dispersion equation is more complex, of course).

2.6.3 Some closing comments on dispersion in FDTD grids

Given ω, L, C, Δt, and Δz, the above non-linear equation can be solved numerically for β, allowing us to determine the phase velocity as a function of frequency. This is shown graphically in Fig. 2.26.

The exact group velocity is $v_g^{\text{exact}} = 1$ m/s, and the group velocity resulting from using Yee's algorithm varies over the range of frequencies simulated in our model problem from this value to around 0.984 m/s, a difference exceeding 10%.

As a closing comment on the subject of dispersion, it is interesting to note, from Eq. (2.108), that if the FDTD simulation is run *at the Courant limit*, viz. $\Delta t = \Delta z/c$, with $c = 1/\sqrt{LC}$, the term in front of the second sinusoid becomes unity, hence the sinusoids are equal and hence their arguments, thus $\omega/\beta = \Delta z/\Delta t \equiv v$, in other words, *there is no dispersion*. This is also sometimes known as the "magic" time-step. This implies that an FDTD simulation run at this time-step can (in theory at least) handle Dirac delta functions (of infinitely wide bandwidth). Unfortunately, this does *not* extend to two or three dimensions, and is thus just a curiosity of no practical value. In 2D and 3D, it turns out that dispersion is minimized (but not eliminated) by operating at the Courant limit. FDTD beginners often run their codes well below the Courant limit, believing that their results will be better with a smaller time-step, but due to numerical dispersion, this is *not* the case.

We can summarize this rather counter-intuitive fact as follows: *FDTD codes should be run as close to the Courant limit as possible*. It should also be noted that

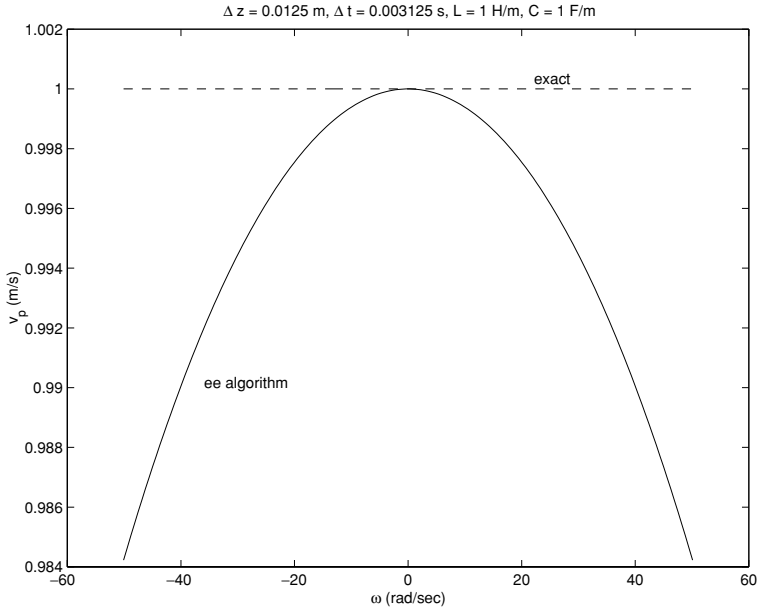


Figure 2.26 Phase velocity as a function of frequency.

numerical dispersion is frequency dependent, and worsens rapidly above a certain frequency. As such, when using a wideband source, we should be careful to ensure that we use a source whose spectrum does not have significant frequency content in this region. This is where rules-of-thumb such as 10 cells per wavelength criteria used earlier in this chapter arise; we appreciate here that the concept of "wavelength" is rather nebulous in the case of a wideband simulation, and we should rather interpret this as the wavelength corresponding to the maximum frequency of interest – often chosen as the point where the spectrum of the source is $1/e$ of its maximum value (this is -8.6859 dB; -10 dB is also sometimes used). It must be appreciated that these are guidelines rather than exact rules. It should also be appreciated that these rules arose in an era when structures being simulated where at most a wavelength or two in size; for larger structures, as can now be undertaken, a *finer* discretization is required since dispersion accumulates over the length of the simulation.

2.7 Conclusion

In this chapter, we have used a very simple one-dimensional transmission line example to introduce the FDTD algorithm. We have seen from first principles how to derive the update equations; this has also given us a handle of the accuracy

of the method. Due to its second-order nature, the Yee algorithm is surprisingly accurate. The important issue of stability has been discussed, and we have seen that the Courant stability criterion is a necessary, but not sufficient, condition for stability – the boundary conditions can also cause instabilities, although as we have commented, in most FDTD simulations, this is not usually a major cause of concern.

Although the FDTD method can be used in the frequency domain, by simply waiting for the transients to die out – and indeed, our first example did just that – this is an inefficient use of the method, which is capable of generating wideband data in one run. This has been discussed in depth in this chapter.

Finally, the fact that the FDTD method has numerical dispersion has been discussed, as well as the implications. Importantly, and perhaps counter-intuitively, FDTD codes should be run as close to the stability limit as possible to minimize dispersion.

With some very simple substitutions, one can solve one-dimensional TEM field problems using the same theory that we have introduced. However, we prefer now to move into two dimensions, and immediately address field problems there. This is the topic of the next chapter.

References

[1] K. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equation in isotropic media," *IEEE Trans. Antennas Propag.*, **14**, 302–307, May 1966.
[2] A. Taflove, *Advances in Computational Electrodynamics: the Finite Difference Time Domain Method*. Boston, MA: Artech House, 1998.
[3] R. P. Feynman, R. B. Leighton, and P. Sands, *The Feynman Lectures on Physics*, vol. 1. Reading, MA: Addison-Wesley, 1963.
[4] R. P. Feynman, R. B. Leighton, and P. Sands, *The Feynman Lectures on Physics*, vol. 2. Reading, MA: Addison-Wesley, 1963.
[5] C. L. Wagner and J. B. Schneider, "Divergent fields, charge, and capacitance in FDTD simulations," *IEEE Trans. Microwave Theory Tech.*, **46**, 2131–2136, December 1998.
[6] D. Davidson and R. W. Ziolkowski, "Body-of-revolution finite-difference time-domain modelling of space-time focusing by a three-dimensional lens," Special Issue on 3D Scattering, *J. Opt. Soc. Am. A*, **11**, 1471–1490, April 1994.

3

The finite difference time domain method in two and three dimensions

3.1 Introduction

In the previous chapter, the basic concepts of the finite difference time domain method were introduced via a one-dimensional example. We will briefly reprise the issues one must attend to when doing an FDTD simulation, as follows.

- An FDTD mesh (or grid) must be created for the problem. (This is trivial in 1D, requires a little thought in 2D, and becomes quite a major problem in 3D.)
- This mesh must be fine enough – i.e. Δs must be no more than perhaps one-tenth of the minimum wavelength (i.e. maximum frequency) of interest (Δs represents the spatial step size; quite often, Δx, Δy and Δz are chosen equal and Δs is used as shorthand for this).
- The time step Δt must satisfy the Courant limit (but be as close to this as possible to minimize dispersion).
- Boundary conditions (the source and load resistors in our 1D example) must be specified.
- An appropriate signal shape (e.g. differentiated Gaussian) with suitable time duration for the desired spectral content must be chosen. Also, in general, its spatial position must be specified. (In the transmission line example, it was fixed as the source voltage generator.)

In this chapter, we will study the FDTD method in two and three dimensions. Firstly, we will develop a 2D simulator for a problem of scattering in free space. Following this, a very important development, the perfectly matched layer absorbing boundary condition, will be discussed and implemented. This is followed by a brief discussion of the extension to three dimensions. We conclude the chapter with a discussion of the use of CST MICROWAVE STUDIO$^{\text{TM}}$, a commercial electromagnetics simulation package which includes an FDTD solver.

3.2 The 2D FDTD algorithm

We will now apply these ideas to a free-space scattering problem in two dimensions. Firstly, we remind the reader that although the real world is obviously three dimensional, many useful problems can be solved when one of the dimensions is much longer than the other two. In this case, we generally assume that the field solution does not *vary* in this dimension – often arbitrarily chosen to be the z-direction, which allows us to simplify the analysis greatly. (A note: assuming that there is no *variation* in z, for instance, does *not* preclude $\hat{z}$-*directed* fields; this point can sometimes cause confusion.) In electromagnetics, this assumption permits us to *decouple* the Maxwell equations into two sets of fields or modes, as they are often called: transverse magnetic and transverse electric.[1] Any field *subject to the assumption of no variation in* z can be written as the sum of these modes:

Transverse magnetic TM, often written TM_z, modes contain the following field components: $E_z(x, y, t)$, $H_x(x, y, t)$ and $H_y(x, y, t)$.

Transverse electric TE, often written TE_z, modes contain the following field components: $H_z(x, y, t)$, $E_x(x, y, t)$ and $E_y(x, y, t)$.

At the risk of repetition, there is *no* z variation in any of the above fields.

3.2.1 Electromagnetic scattering problems

When an electromagnetic field encounters a target,[2] currents are excited on it, which in turn re-radiate. This process is called "electromagnetic scattering." Obvious applications are in radar, and also in multi-path analysis for radio-wave propagation. Since the Maxwell equations are linear, the fields are often decomposed into an *incident* field $\vec{E}^{\mathrm{inc}}$ and a *scattered* field $\vec{E}^{\mathrm{scat}}$. The overall field, called the *total* field $\vec{E}^{\mathrm{tot}}$, is then:

$$\vec{E}^{\mathrm{tot}} = \vec{E}^{\mathrm{inc}} + \vec{E}^{\mathrm{scat}} \tag{3.1}$$

By definition, the incident field is the field which would exist if the scatterer were absent. This is very useful; often, this will be a plane wave which can easily be expressed mathematically in closed form. We will see shortly how useful this idea can be when studying scattering.

[1] Readers who have previously studied waveguide analysis will immediately recognize these concepts.
[2] Because most of the original work was done for radar applications, the military term "target" is frequently used for describing the scatterer in such circumstances.

3.2.2 The TE_z formulation

At this stage, we could solve either (or both) transverse modes; the FDTD process is essentially identical. We will chose the TE_z formulation, because TE_z waves exhibit interesting behavior when scattering off circular targets – *creeping waves* are excited on the structure, i.e. a wave "attaches" itself to the cylinder, goes around the target and then comes back towards the source, potentially in or out of phase with the incident field. (TM_z waves do not do this; the reason is that the boundary conditions are different.)

The TE_z mode set is described by the following parts of Maxwell's equations:

$$\frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_z}{\partial y} - \sigma E_x\right) \tag{3.2}$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\varepsilon}\left(-\frac{\partial H_z}{\partial x} - \sigma E_y\right) \tag{3.3}$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x}\right) \tag{3.4}$$

We will simplify these further by assuming that the materials are lossless:

$$\frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon}\frac{\partial H_z}{\partial y} \tag{3.5}$$

$$\frac{\partial E_y}{\partial t} = -\frac{1}{\varepsilon}\frac{\partial H_z}{\partial x} \tag{3.6}$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x}\right) \tag{3.7}$$

In the transmission line case of the previous chapter it will be recalled that we chose "half-step" increments for the current. We will apply the same idea to developing a 2D FDTD solution of the above equations. We will make the following choices:

$$\begin{aligned} x_i &= (i-1)\Delta x, \qquad i = 1, 2, \ldots, N_x \\ \Delta x &= \frac{X}{N_x - 1}, \qquad N_x \geq 2 \end{aligned} \tag{3.8}$$

$$\begin{aligned} y_k &= (j-1)\Delta y, \qquad j = 1, 2, \ldots, N_y \\ \Delta y &= \frac{Y}{N_y - 1}, \qquad N_y \geq 2 \end{aligned} \tag{3.9}$$

$$t_n = (n-1)\Delta t, \qquad n = 1, 2, 3, \ldots$$
$$\Delta t = \frac{T}{M-1}, \qquad M \geq 2 \tag{3.10}$$

Here, X and Y are the dimensions of the region we will be gridding (in the x and y directions) and N_x and N_y obviously are the number of cells in each dimension. It is traditional, but certainly not essential, to associate the indices i, j and k in an FDTD code with x, y and z, and m or n with t.

Coding hints – the indices i, j and MATLAB

At this point, it is worth sounding a warning that using these traditional indices can cause very frustrating problems in MATLAB, where i, and j, are usually defined as $\sqrt{-1}$. A useful programming habit to develop is instead to use ii and jj as indices.

A similar array of half-index points will also be defined:

$$x_{i+1/2} = (i-1/2)\Delta x, \qquad i = 1, 2, \ldots, N_x \tag{3.11}$$
$$y_{j+1/2} = (j-1/2)\Delta y, \qquad j = 1, 2, \ldots, N_y \tag{3.12}$$
$$t_{n+1/2} = (n-1/2)\Delta t, \qquad n = 1, 2, 3, \ldots \tag{3.13}$$

Following Yee's choice, we will locate $H_z(i, j, n)$ at $x_{i+1/2}; y_{j+1/2}; t_{n+1/2}$. $E_x(i, j, n)$ will be located at $x_{i+1/2}; y_j; t_n$ and $E_y(i, j, n)$ at $x_i; y_{j+1/2}; t_n$. This choice is far from random; it provides a spatial grid with the magnetic field H_z surrounded in space by the electric fields – $E_x(i, j, k)$ and $E_x(i, j+1, k)$, $E_y(i, j, k)$ and $E_y(i+1, j, k)$ – and offset in time by $\Delta t/2$. The spatial locations are indicated in Fig. 3.1.

Now we will turn our attention to the discretization of the FDTD TE_z modes. Consider Eq. (3.7). We apply central differencing to both time and space, producing:

$$\begin{aligned}
&\frac{H_z(i+\frac{1}{2}, j+\frac{1}{2}, n+\frac{1}{2}) - H_z(i+\frac{1}{2}, j+\frac{1}{2}, n-\frac{1}{2})}{\Delta t} \\
&= \frac{1}{\mu}\left[\frac{E_x(i+\frac{1}{2}, j+1, n) - E_x(i+\frac{1}{2}, j, n)}{\Delta y}\right] \\
&- \frac{1}{\mu}\left[\frac{E_y(i+1, j+\frac{1}{2}, n) - E_y(i, j+\frac{1}{2}, n)}{\Delta x}\right]
\end{aligned} \tag{3.14}$$

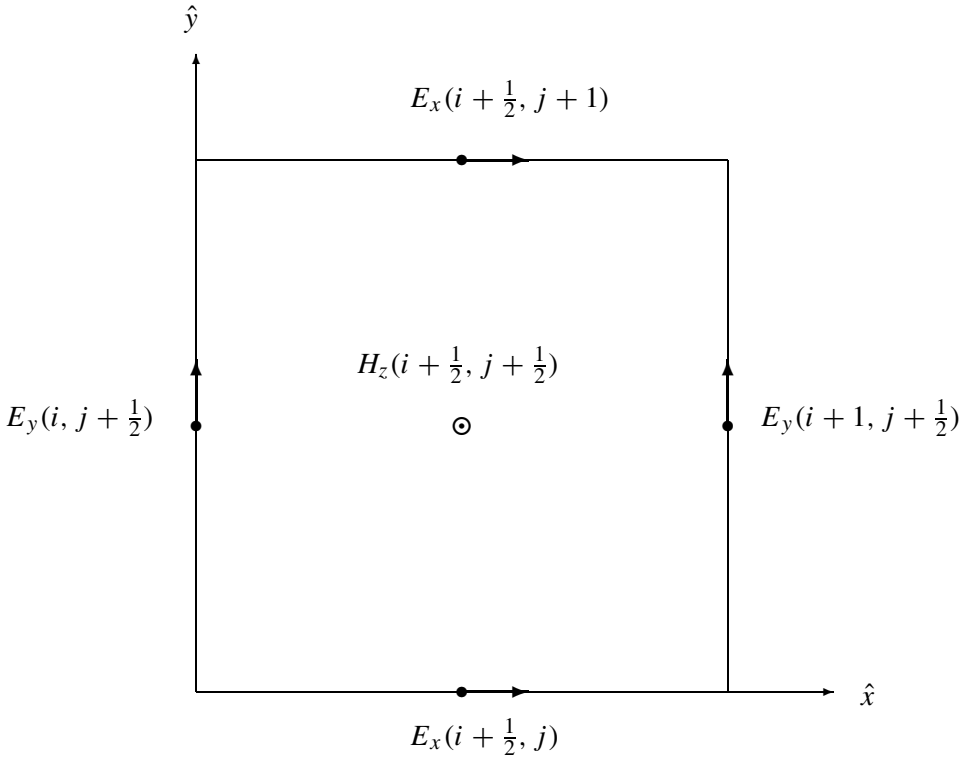


Figure 3.1 The Yee grid for the i, jth cell for the FDTD 2D TE$_z$ mode.

Now, keeping only $H_z(i+\frac{1}{2}, j+\frac{1}{2}, n+\frac{1}{2})$ on the left-hand side of the equation, we rewrite this as:

$$H_z\left(i+\frac{1}{2}, j+\frac{1}{2}, n+\frac{1}{2}\right) = H_z\left(i+\frac{1}{2}, j+\frac{1}{2}, n-\frac{1}{2}\right) + \frac{\Delta t}{\mu \Delta y}\left[E_x\left(i+\frac{1}{2}, j+1, n\right) - E_x\left(i+\frac{1}{2}, j, n\right)\right] - \frac{\Delta t}{\mu \Delta x}\left[E_y\left(i+1, j+\frac{1}{2}, n\right) - E_y\left(i, j+\frac{1}{2}, n\right)\right] \tag{3.15}$$

Similar procedures, applied to Eqs. (3.5) and (3.6), produce the update equations for the E-field components:

$$E_x\left(i+\frac{1}{2}, j, n+1\right) = E_x\left(i+\frac{1}{2}, j, n\right) + \frac{\Delta t}{\varepsilon \Delta y}\left[H_z\left(i+\frac{1}{2}, j+\frac{1}{2}, n+\frac{1}{2}\right) - H_z\left(i+\frac{1}{2}, j-\frac{1}{2}, n+\frac{1}{2}\right)\right] \tag{3.16}$$

$$E_y\left(i, j+\frac{1}{2}, n+1\right) = E_y\left(i, j+\frac{1}{2}, n\right) - \frac{\Delta t}{\varepsilon \Delta x}\left[H_z\left(i+\frac{1}{2}, j+\frac{1}{2}, n+\frac{1}{2}\right) - H_z\left(i-\frac{1}{2}, j+\frac{1}{2}, n+\frac{1}{2}\right)\right] \tag{3.17}$$

Just as in the 1D case, the half space and time increments are inconvenient to program, and we will refer simply to i, j, n for E_x, E_y and H_z, but keeping in mind the actual locations. We will also assume $\Delta x = \Delta y = \Delta s$. This allows us the simplify the above to the following:

$$H_z(i, j, n) = H_z(i, j, n-1) + \frac{\Delta t}{\mu \Delta s}\left[E_x(i, j+1, n) - E_x(i, j, n) + E_y(i, j, n) - E_y(i+1, j, n)\right] \tag{3.18}$$

$$E_x(i, j, n+1) = E_x(i, j, n) + \frac{\Delta t}{\varepsilon \Delta s}\left[H_z(i, j, n) - H_z(i, j-1, n)\right] \tag{3.19}$$

$$E_y(i, j, n+1) = E_y(i, j, n) - \frac{\Delta t}{\varepsilon \Delta s}\left[H_z(i, j, n) - H_z(i-1, j, n)\right] \tag{3.20}$$

Note that when the electric fields are updated, the magnetic field values used are the newly updated ones.

We now have our update equations, and the Courant limit for two dimensions:

$$\Delta t \leq \frac{\Delta s}{\sqrt{2}c} \tag{3.21}$$

where c is the (largest) speed of light in the FDTD region (in non-vacuum regions, the speed of light is of course slowed). We are not quite ready to program, however. There are two things we still need to consider: injecting a source, and terminating the mesh.

3.2.3 Including a source: the scattered/total field formulation

If we want to study scattering, we need a method for simulating a plane wave. (Usually, scattering problems assume that whatever source setup the incident field is far removed from the scatterer, and hence the field incident on the target is a uniform plane wave.) The simplest method for doing this is to exploit the concepts of incident, scattered and total fields introduced in Section 3.2.1. Since the Maxwell equations are linear, and we will only work with linear materials here, we can use the FDTD to solve for either the scattered or total fields. (Remember that the incident field is assumed known in this type of formulation.) We will split the computational area into two zones using a (non-physical) line, as in Fig. 3.2: in one region, we will have only scattered fields, and in the other, total fields. For

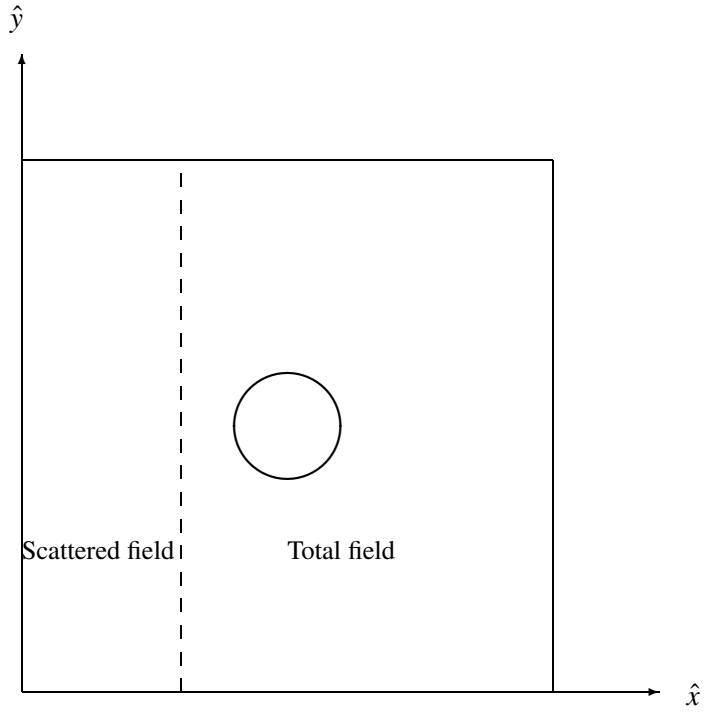


Figure 3.2 The scatterer and surrounding FDTD zones, showing scattered field and total field regions.

convenience, we will choose a constant x coordinate; we will assume this corresponds to index $i_L = L$ (the subscript L short for left; we could also position another scattered field zone to the right of the scatterer, etc.).

Now we note one of the points which can sometimes cause problems with the FDTD algorithm. Do we interpret i_L as being on a spatial step or half-step? There is no correct answer to this, we need to make a decision and then work consistently with this. Since three of the five field components in the two-dimensional Yee cell are located at half-step values of x, let us choose this. Hence our scattered/total field demarcation is located at $x_L = (L - \frac{1}{2})\Delta$. Fields located *on* and to the right of this line this will be chosen as total fields. Fields to the left will be scattered fields.

Clearly, we cannot simultaneously work with scattered and total fields in the update equations. However, because we know the incident field, we can add or subtract this as necessary. Let us consider the update equation for H_z. Here, we will retain the full notation (including half-steps) to avoid confusion. For $i < i_L$, we use Eq. (3.15), with all the fields *scattered* fields. For $i > i_L$, we use the same Eq. (3.15), but now all the fields are *total* fields.

On the zoning interface, $i = i_L$, we have a total $H_z^{\text{tot}}(i + \frac{1}{2}, j + \frac{1}{2})$ field, total E_x^{tot} fields, a total $E_y^{\text{tot}}(i + 1, j + \frac{1}{2})$ field and a scattered $E_y^{\text{scat}}(i, j + \frac{1}{2}, n)$ field. We can make this last consistent by adding the known incident field $E_y^{\text{inc}}y(i, j + \frac{1}{2}, n)$. The update equation for $i = i_L$ becomes:

$$\begin{aligned} H_z^{\text{tot}}&\left(i_L + \frac{1}{2}, j + \frac{1}{2}, n + \frac{1}{2}\right) \\ &= H_z^{\text{tot}}\left(i_L + \frac{1}{2}, j + \frac{1}{2}, n - \frac{1}{2}\right) \\ &\quad + \frac{\Delta t}{\mu \Delta y}\left[E_x^{\text{tot}}\left(i_L + \frac{1}{2}, j + 1, n\right) - E_x^{\text{tot}}(i_L + \frac{1}{2}, j, n\right)\right] \\ &\quad - \frac{\Delta t}{\mu \Delta x}\left[E_y^{\text{tot}}\left(i_L + 1, j + \frac{1}{2}, n\right)\right. \\ &\quad \left. - E_y^{\text{scat}}\left(i_L, j + \frac{1}{2}, n\right) - E_y^{\text{inc}}\left(i_L, j + \frac{1}{2}, n\right)\right] \end{aligned} \tag{3.22}$$

For the E_y component located at $x_L = (L - 1)$, i.e. just to the left of the interface, all the fields in the update equations are scattered, except for the H_z field located at $x_L = (L - \frac{1}{2})\Delta_x$. The update equation for this becomes:

$$\begin{aligned} E_y^{\text{scat}}&\left(i_L, j + \frac{1}{2}, n + 1\right) \\ &= E_y^{\text{scat}}\left(i_L, j + \frac{1}{2}, n\right) - \frac{\Delta t}{\varepsilon \Delta x}\left[H_z^{\text{tot}}\left(i_L + \frac{1}{2}, j + \frac{1}{2}, n + \frac{1}{2}\right)\right. \\ &\quad \left. - H_z^{\text{inc}}\left(i_L + \frac{1}{2}, j + \frac{1}{2}, n + \frac{1}{2}\right) - H_z^{\text{scat}}\left(i_L - \frac{1}{2}, j + \frac{1}{2}, n + \frac{1}{2}\right)\right] \end{aligned} \tag{3.23}$$

The update equation for the other component, E_x, involves either only total fields (for $i \geq i_L$) or only scattered fields (for $i < i_L$) and hence can be used without change.

As an example, if the incident field is a plane wave, propagating in the x-direction, in free space, with time history $x(t)$, with a z-polarized magnetic field, the expressions for the incident fields are:

$$E^{\text{inc}} = x(t - t_{D_E})\hat{y} \tag{3.24}$$

$$H^{\text{inc}} = \frac{1}{\eta_0} x(t - t_{D_H})\hat{z} \tag{3.25}$$

$\eta_0 = \sqrt{\mu_0/\varepsilon_0}$ is the wave impedance of free space. t_D is the delay time from some arbitrary start location. For the problem shown in Fig. 3.2, this could conveniently

be taken as $x = 0$. For the magnetic fields located at $x_L = (L - \frac{1}{2})\Delta x$, the delay time is $t_{D_H} = (L - \frac{1}{2})\Delta x/c$; for the electric fields located at $x_L = (L - 1)\Delta x$, the delay time is $t_{D_E} = (L - 1)\Delta x/c$. In short, the half-delta difference in spatial position of the fields must be taken into account. Note that these delay times are only valid for the specific case of a field propagating only in the $\hat{x}$-direction. Formulas are easily derived for plane waves propagating in other directions, but the above is sufficient for now.

Considering how simple it was to include the 1D source, one might wonder why this apparently much more complex approach is necessary in 2D. It is possible to include a simple line source in 2D in much the same way as in 1D, by simply specifying the value of the source at a particular point in the mesh. This however radiates *cylindrical*, not plane, waves; hence, this approach is not useful for most scattering problems. However, it is convenient for initial code testing, and also for checking the operation of absorbing boundary conditions. The next idea that springs to mind is simply to drive a *line* of points in the mesh with some source function. The problem with this is more subtle; suffice it to say for now that although this seems like a simple approach, it does not give good results in practice.

3.2.4 Meshing the scatterer

The process of generating a suitable FDTD grid for a problem is often called "meshing." As already indicated, this can be a formidable problem in general. We will be using a very simple test problem – a circular cylinder.[3] This will allow us to make a very simple "mesher." We will place the cylinder, radius a, at a convenient location in the mesh and then simply compute the distance to a point in the mesh; if this distance exceeds a, the point lies outside the cylinder, if it is less than or equal to a, it lies inside or on the surface. Since the E_x and E_y field components are offset in space, we must do this for each component. As a first pass, we will make the cylinder highly conducting, indeed perfectly conducting, so that the (total) fields inside the cylinder are zero. The appropriate boundary condition will be to zero the fields tangential to the cylinder.

The above sounds very straightforward. It is only when coding that a whole number of problematic issues suddenly appear. The first is that we have spoken about "tangential" fields. With a round cylinder, the tangent will only lie in the $\pm\hat{x}$ or $\pm\hat{y}$ directions at four points (top, bottom, right and left in Fig. 3.2). Elsewhere, in all the other FDTD cells which the boundary of the cylinder passes through,

[3] "Cylinder" is the general mathematical description of any object generated by translating a two-dimensional cross-section along its normal. For instance, a "cylinder" may be square. (In normal English usage, a cylinder is round.) The full mathematical term for what is commonly called a cylinder is a "right circular cylinder."

we are only going to be able to *approximate* the boundary, and because we use a rectangular grid, the resulting approximation is often called a "stair-step" approximation.

This problem emerged because we are modelling round (or more generally, curvilinear) structures with a rectangular grid. But even if we only model rectangular structures which can be aligned to the FDTD grid, another problem still remains. Refer back to Fig. 3.1. Now, instead of modelling a PEC (perfect electrical conductor) scatterer, let us rather model a cylinder made of some dielectric material, with permittivity ε_R. In the update equations, we need to specify the value for $\varepsilon = \varepsilon_R \varepsilon_0$. Assume we do this for $E^y(i+1, j+\frac{1}{2})$. Now, what do we do with the two E^x components located $\Delta x/2$ to the left of this interface? If we set ε_R for them as well, the interface has effectively been "moved" slightly to the left, and now we have the same problem with $E^y(i, j+\frac{1}{2})$ ... If we do not, the interface is then located somewhere between $(i+\frac{1}{2})\Delta x$ and $(i+1)\Delta x$. Again, this is a problem without a simple answer. Due to the half-step offsets in the FDTD Yee grid, there is an uncertainty about the precise position of material interfaces in the basic Yee algorithm. Since it is a maximum of a half-cell, and the cells are usually quite small, it is normally acceptable, but can be problematic. ("Averaging" methods have been used successfully to correct this, and to improve the modelling of curvilinear structures, but we will not consider these at present.)

One final issue still remains to be solved before we develop a 2D FDTD code for scattering off a cylinder: how do we terminate the mesh? The problem is the following: we want to simulate a free-space environment, which means that waves scattered off the target should radiate radially away to infinity, diminishing in strength and eventually disappearing. Clearly, we cannot make an FDTD grid sufficiently large to simulate this. If one has seen an anechoic chamber used for antenna measurements, one will know that antenna designers have a similar problem; they have solved this by coating the walls of the anechoic chamber with an absorbing material. This, effectively, is what we will attempt to do now.

3.2.5 Absorbing boundary conditions

The field of absorbing boundary conditions (ABCs) attracted much research throughout the 1980s and early 1990s. Two methods have historically been pursued: radiation BCs and absorbing BCs. The term ABC is also used more generally for both. The former modifies the FDTD update equations; the latter modifies the material properties in the mesh.

Having really good ABCs, and here is meant ABCs with a reflection coefficient less than -60 or -70 dB, means that it is possible to bring the ABC close to the radiating/scattering structure, "wasting" as few Yee cells as possible meshing

up free space. Due to the great interest in the field, one will find a large number of references on the topic. Later in this chapter, we will introduce a revolutionary boundary condition, the perfectly matched layer, but for the time being, we will use a very simple ABC. The idea is the following, for a $-x$ traveling wave on plane $x = 0$. It uses the concept of one-way wave equation, also known as the advective equation, with a wave solution $f(x + ct)$, traveling only in the $-\hat{x}$ direction:

$$\left[\frac{\partial}{\partial x} - \frac{1}{c}\frac{\partial}{\partial t}\right]\phi(x,t) = 0 \tag{3.26}$$

$\phi(x,t)$ represents one of the components of the wave. This leads then to a 1D ABC, as follows. We impose this one-way wave equation on a wave incident on a surface normal to $\hat{x}$:

$$\left.\frac{\partial}{\partial x}\phi(x,t)\right|_{x=0} = \left.\frac{1}{c}\frac{\partial}{\partial t}\phi(x,t)\right|_{x=0} \tag{3.27}$$

Applying *forward differencing* in x and t, one obtains:

$$\phi_1^n - \phi_0^n \approx \frac{\Delta x}{c\Delta t}(\phi_0^{n+1} - \phi_0^n) \tag{3.28}$$

Finally, rewrite this to give the desired ABC:

$$\phi_0^{n+1} = \phi_0^n\left(1 - \frac{c\Delta t}{\Delta x}\right) + \frac{c\Delta t}{\Delta x}\phi_1^n \tag{3.29}$$

This analysis must be repeated at the boundary $x = x_{\max}$. In this case, the relevant one-way wave equation, with solution in this case $f(x - ct)$, traveling only in the $+\hat{x}$ direction, is

$$\left[\frac{\partial}{\partial x} + \frac{1}{c}\frac{\partial}{\partial t}\right]\phi(x,t) = 0 \tag{3.30}$$

Imposed on a wave incident on a surface normal to $\hat{x}$, the wave is again "absorbed." This leads then to the other 1D ABC:

$$\left.\frac{\partial}{\partial x}\phi(x,t)\right|_{x=x_{\max}} = -\left.\frac{1}{c}\frac{\partial}{\partial t}\phi(x,t)\right|_{x=x_{\max}} \tag{3.31}$$

Applying *backward differencing* in x and forward differencing in t as before, one obtains:

$$\phi_{N_x}^n - \phi_{N_x-1}^n \approx -\frac{\Delta x}{c\Delta t}(\phi_{N_x}^{n+1} - \phi_{N_x}^n) \tag{3.32}$$

Finally, rewrite this to give the desired ABC:

$$\phi_{N_x}^{n+1} = \phi_{N_x}^n\left(1 - \frac{c\Delta t}{\Delta x}\right) + \frac{c\Delta t}{\Delta x}\phi_{N_x-1}^n \tag{3.33}$$

Interestingly, the equation is identical in form to Eq. (3.29). The extension to $\pm\hat{y}$ propagating waves on the planes $y = 0$ and $y = y_{\max}$ is obvious.

As noted, ϕ was used here; clearly, we need to apply this to the various tangential field components at each boundary. Note that we only need apply it to either $\vec{E}$ or $\vec{H}$; once we establish one of the fields "outside" the computational domain, the usual update equations, combined of course with the half-space step offset, establishes the other.

Because this ABC used forward differencing, it is only accurate to first order. (Remember that the Yee scheme has second-order accuracy.) It is "exact" in 1D; in 2D and 3D, for paraxial incidence, reflection coefficients of $\Gamma < -25$ dB may be obtained, but it degrades rapidly off-normal. Mur, in 1981, published a more complete first-order ABC, as well as a second-order one. Details are available in [1, Chapter 6]. These first- and second-order Mur ABCs are still widely used, owing to their simplicity and reasonable effectiveness; however, commercial codes should also offer perfectly matched layers.

We now have all the tools needed to produce a 2D FDTD simulation of electromagnetic scattering from a cylinder in free space – we already have suitable wideband pulses from our 1D work. We will now proceed to develop the simulator.

3.2.6 Developing the simulator

There are a number of issues to consider when turning this algorithm into code. Although we will not be excessively concerned with computational efficiency initially, it is good practice nonetheless to consider some issues. Firstly, division is a much more expensive process in terms of computing time than multiplication. Equation (3.18) contains a term $\Delta t/\mu\Delta s$, and Eqs. (3.19) and (3.20) both contain the term $\Delta t/\varepsilon\Delta s$. Usually, there will only be a few different material regions in an FDTD code. So, it would be better to store these as an array representing material properties, perform the division once before the time-stepping starts, and then simply use the relevant value of this array at each stage. One of these is needed per field component:

$$H_z(i, j, n) = H_z(i, j, n-1) + D_{Hz}(i, j)\,[E_x(i, j+1, n) - E_x(i, j, n) + E_y(i, j, n) - E_y(i+1, j, n)] \tag{3.34}$$

$$E_x(i, j, n+1) = E_x(i, j, n) + C_{Ex}(i, j)\,[H_z(i, j, n+1) - H_z(i, j-1, n+1)] \tag{3.35}$$

$$E_y(i, j, n+1) = E_y(i, j, n) - C_{Ey}(i, j)\,[H_z(i, j, n+1) - H_z(i-1, j, n+1)] \tag{3.36}$$

with

$$C_{Ex}(i, j) = \frac{\Delta t}{\varepsilon([i - 1/2]\Delta s, [j - 1]\Delta s)} \tag{3.37}$$

$$C_{Ey}(i, j) = \frac{\Delta t}{\varepsilon([i - 1]\Delta s, [j - 1/2]\Delta s)} \tag{3.38}$$

$$D_{Hz}(i, j) = \frac{\Delta t}{\mu([i - 1/2]\Delta s, [j - 1/2]\Delta s)} \tag{3.39}$$

where the (x, y) coordinates at which ε and μ are to be evaluated are explicitly indicated. The previous discussion in Section 3.2.4 regarding the exact position of material interfaces refers again.

Coding hints – programming the update equations efficiently

The obvious way of programming Eqs. (3.34)–(3.36) is to use a double-loop (a DO-loop in FORTRAN, a FOR-loop in many other languages, including MATLAB). However, with MATLAB, this is not a good idea. The problem is that MATLAB is an interpreted language, as opposed to a compiled one, and only runs efficiently when its (highly optimized) vector commands can be used by the interpreter. So, an update such as Eq. (3.36) is best programmed as in Fig. 3.3 – note that the ... is the MATLAB line continuation character.

```
E_y_n(2:N_x,2:N_y) = E_y_nmin1(2:N_x,2:N_y) ...
    - C(2:N_x,2:N_y).*(  H_z_n(2:N_x,2:N_y)  - H_z_n(1:N_x-1,2:N_y) )
```

Figure 3.3 MATLAB code stub for updating E_y.

This looks somewhat cryptic on a first reading: the key operation is `H_z_n(2:N_x,2:N_y) - H_z_n(1:N_x-1,2:N_y)` which effectively shifts the second occurrence of the `H_z_n` array along its first dimension (corresponding to x) and permits the difference to be formed as a vector operation. It is also clear why the indices must run from 2 to N_x, rather than from 1 to N_x (and similarly along the second dimension); otherwise, the operation would refer to non-existing array elements at 0 when shifted. These, the boundary values, must be computed separately. The `.*` operation in MATLAB denotes element-by-element multiplication (also sometimes known as the *outer* product of two matrices).

A point to note when coding is that because the FDTD algorithm is *explicit*, the new values that we compute at a point are *not* affected by the new values at any other points. Hence, we do not need to take particular care at the line

```
% Update H fields:
  H_z_n(1:N_x-1,1:N_y-1) = H_z_nmin1(1:N_x-1,1:N_y-1) ...
      + D(1:N_x-1,1:N_y-1).*(  E_x_nmin1(1:N_x-1,2:N_y)   - E_x_nmin1(1:N_x-1,1:N_y-1) ...
                             + E_y_nmin1(1:N_x-1,1:N_y-1) - E_y_nmin1(2:N_x,1:N_y-1) ) ;
% Special update on scat/tot field boundary
  E_y_nmin1_inc = ones(1,N_y)*gaussder((m-1)*delta_t - (L-1)*delta_s/c,m_offset,sigma) ;
  H_z_n(L,1:N_y-1) = H_z_nmin1(L,1:N_y-1) ...
      + D(L,1:N_y-1).*(  E_x_nmin1(L,2:N_y)     - E_x_nmin1(L,1:N_y-1) ...
                       + E_y_nmin1(L,1:N_y-1) + E_y_nmin1_inc(1:N_y-1) - E_y_nmin1(L+1,1:N_y-1)
```

Figure 3.4 MATLAB code stub for updating H_z.

corresponding to scattered/total field interface $i = L$. We can update values at this point as usual with a vector operation, and then overwrite them with the correct values. (Obviously, the values of H_z, for instance, must be correct before we start the updates of the electric fields, and vice versa.) Although this involves a small amount of unnecessary computation – in this case, we compute the values along the line separating the scattered/total field twice – the savings in code complexity are so significant that this is almost universal practice in FDTD codes. In the code stub shown in Fig. 3.4, we show the update for the $\vec{H}$ field, demonstrating this idea. The semicolons at the end of each line prevent the results being written to the command window, which is essential with the large datasets which the FDTD can easily generate. `gaussder` is a function which returns a suitable differentiated Gaussian.

With the 1D FDTD, the algorithm is simple enough that it is relatively easy to program correctly. However, our 2D FDTD simulator is already sufficiently complex that to try to program it in its entirety in one go is likely to lead to great frustration. There are no less than three major, different types of errors that can be made. How to test the code systematically, and locate likely errors, will now be discussed.

Coding hints – frequently made errors in MATLAB

MATLAB is an excellent environment for quickly testing and demonstrating algorithms. However, from the viewpoint of programming, it has a number of "features" which would be seen as deficiencies in most programming languages. The most prominent of these is that it is *not* a strictly typed language – indeed, MATLAB has many properties of a scripting language. This means that variables do not need to be declared before they are used. The advantage is convenience; the drawback is reliability. Firstly, one can accidentally overwrite an existing variable; in particular, `i` and `j` offer suffer this fate. A variant of this is that a subtle spelling error creates a different (and usually undefined) variable. Some other errors frequently made in MATLAB, in particular by programmers used to other languages, include:

Indices in `for` *loops* The correct format for the `for` loop indices is `for ii=1:N_x`, for example. FORTRAN programmers in particular are inclined to code this as `for ii=1,N_x`, which is incorrect in MATLAB.

Testing equality versus assignment The correct logical expression to test if `ii` is equal to `jj` is `if ii == jj` (as in C). Again, FORTRAN programmers often code this as `if ii = jj`, which *assigns* the value of `jj` to `ii`.

Both these errors are especially frustrating to locate; MATLAB executes the former incorrectly (or at least incorrectly in terms of the programmer's expectations), and earlier versions also executed the latter (later versions issue a warning).

Implementing the update equations

The easiest mistakes to make here are with the indices. In particular, the repetitiveness of FDTD equations encourages cutting-and-pasting, and one has be *very* careful to correct all the indices (and also field subscripts) when doing this. A simple test which can be used is to note that an FDTD update equation involving (say) the x component of a field on the right-hand side *never* involves a partial derivative (which is of course a difference equation in the code) in x (i.e. the first index). For instance, look at the term in the update for H_z (Fig. 3.4):

```
E_x_nmin1(1:N_x-1,2:N_y)   - E_x_nmin1(1:N_x-1,1:N_y-1)
```

Clearly, the following would be incorrect:

```
E_x_nmin1(2:N_x,2:N_y)   - E_x_nmin1(1:N_x-1,2:N_y) % THIS IS WRONG!
```

It is essential to check the update equations by very carefully reading through each one as programmed.

To check that the update equations are working, a very simple source at one point can be used. Physically, this represents an infinitely long line source. Instead of the full scattered/total field approach shown in Fig. 3.4, the code in Fig. 3.5 injects a source of cylindrical waves in the center of the mesh. Again, note that the source update at $(N_x/2, N_y/2)$ simply overwrites the just updated value. Note also that the E field update equations in this case are simply those of free space. Also combined with this, the outer boundaries at this stage can simply be set as PECs by zeroing the relevant tangential electric field components; see Fig. 3.6 for an example.

```
% Update H fields:
  H_z_n(1:N_x-1,1:N_y-1) = H_z_nmin1(1:N_x-1,1:N_y-1) ...
      + D(1:N_x-1,1:N_y-1).*(  E_x_nmin1(1:N_x-1,2:N_y)   - E_x_nmin1(1:N_x-1,1:N_y-1) ...
                             + E_y_nmin1(1:N_x-1,1:N_y-1) - E_y_nmin1(2:N_x,1:N_y-1) ) ;
% Drive a test line source - used to check basic operation
  H_z_n(N_x/2,N_y/2) = gaussder((m-1)*delta_t,m_offset,sigma);
```

Figure 3.5 MATLAB code stub for updating H_z, using a point (line) source.

```
% Fix outer values of E_tangential as PEC:
E_y_n(1,:) = 0;
E_y_n(N_x,:) = 0;
E_x_n(:,1) = 0;
E_x_n(:,N_y) = 0;
```

Figure 3.6 MATLAB code stub for setting PEC boundaries.

Implementing the plane-wave source

Once one has confidence that the update equations are working, one can proceed to test the full scattered/total field formulation, incorporating the plane-wave source. Now, one needs to start thinking about the electromagnetics of the problem. In the 1D case, we simplified matters by using a set of equations with the speed of light set to 1 m/s. Now, we are working with the real world, and $c \approx 3 \times 10^8$ m/s. Since we are primarily interested in radio-frequency (RF) problems, we will select an RF source, with Gaussian derivative shape, with frequency content in the gigahertz range. It turns out to be convenient to select a signal with $\sigma \approx 1 \times 10^{-10}$; this produces a signal with peak spectral amplitude at about 1.5 GHz; reference to Fig. 2.16[4] shows that at around twice the frequency of peak spectral amplitude, the spectrum has decayed to around 30% of the peak value. In the present case, this is 3 GHz; the wavelength in free space is 10 cm (0.1 m) and now we have some guidelines to setting Δs: we should make this around 1/10 of the wavelength at 3 GHz, viz. $\Delta s = 0.01$ m. (Note that we must be careful to work in SI units!) Δt will be set by the Courant limit (a maximum of 23.587 ps, when using the exact value for c).

For testing the code, it is tempting to set N_x and N_y quite small, for instance, 5 or 10. Whilst this is occasionally necessary when something is really wrong and one is having to step through the code, it is actually a bad idea in general. The reason is that the absorbing boundary conditions are not included yet, and the temporary PEC boundaries suggested above result of course in the wave reflecting back. With small domains, these reflections mean that it is not possible to observe the field develop and propagate properly. A good test uses $N_x = 200$ and $N_y = 100$ (corresponding physically with $\Delta s = 0.01$ m to an area 0.2×0.1 m^2). The scattered/total field zone is placed at $L = 50$.

[4] That plot was normalized to $\sigma = 1$; the extension is obvious.

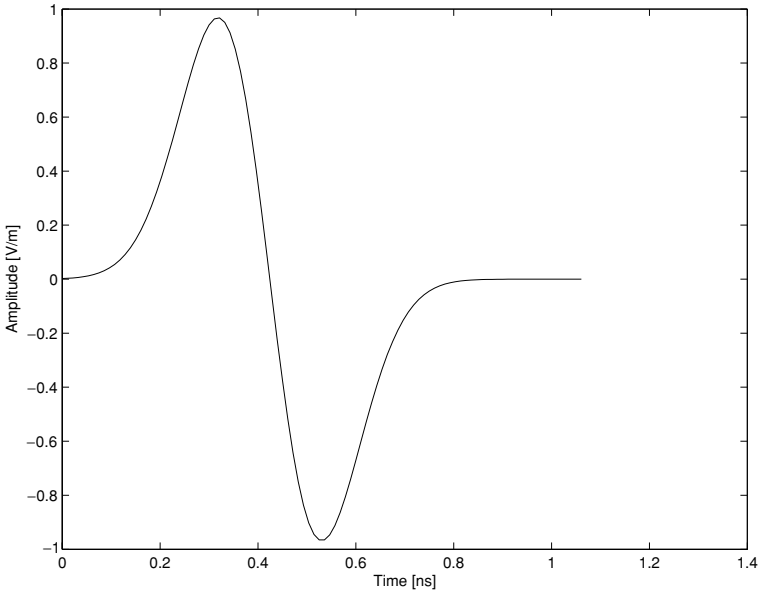


Figure 3.7 Gaussian derivative pulse used for 2D FDTD simulation.

The Gaussian derivative pulse defined in Eq. (2.71) was obtained by differentiating Eq. (2.68), and has inconvenient amplitude behavior (being proportional to $1/\sigma^2$). The following pulse has a far more convenient, almost normalized amplitude:

$$v_0(t) = \frac{-4}{\sqrt{2\pi}} \frac{(t-m)}{\sigma} e^{-(t-m)^2/2\sigma^2} \tag{3.40}$$

Its time history for $\sigma = 1 \times 10^{-10}$, and with $m = 4\sigma$, is shown in Fig. 3.7. The peak amplitude is 0.9670, at 0.3322 ns.

Coding hints – a normalized Gaussian derivative pulse

The following equation defines a properly normalized Gaussian derivative pulse:

$$v_0(t) = -\frac{e^{1/2}}{\sigma} (t-m) e^{-(t-m)^2/2\sigma^2} \tag{3.41}$$

The normalizing constant $e^{1/2}/\sigma$ provides a unit peak amplitude at $t - m = \pm\sigma$. Since the results in this chapter do not require this, the signal in Eq. (3.40) is used in the following discussion.

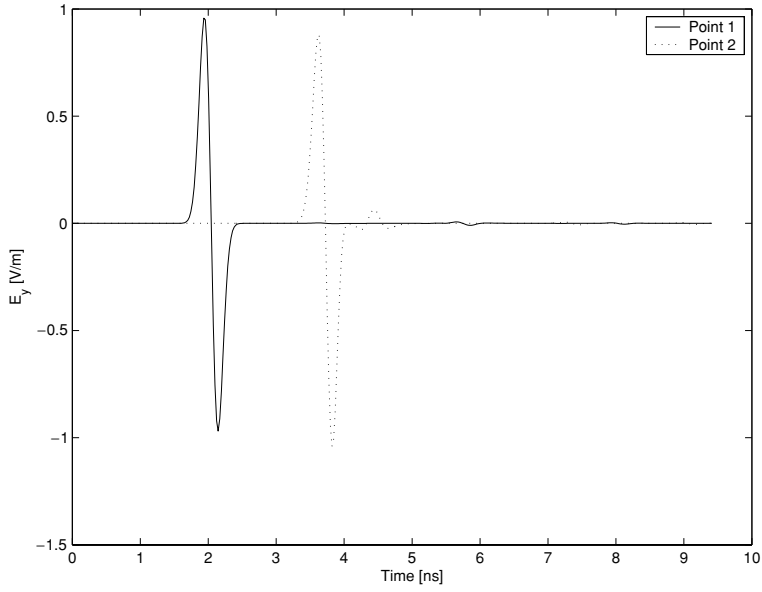


Figure 3.8 Gaussian derivative pulse at a point just to the right of the scattered/total field zone.

This can now be injected into the scattered/total field code. We will monitor the H_z field (scaled by η_0, to give a peak value close to unity). We will do this at point 1, with indices $(L+1=51;50)$, just to the right of the scattered/total field interface, and at point 2, with indices $(101;50)$. The result, for $M=400$ time-steps, in shown in Fig. 3.8. The first peak value is at 1.9577 ns, the second, at 3.6559 ns. Now, we establish whether this checks with basic physics. The time difference between the peaks of these pulses is 1.6982 ns. In free space, it should take 1.6678 ns propagating at the speed of light to cover the distance of $50\Delta x = 0.5$ m. This is a difference of around 1.8%. This is very probably due to numerical dispersion. To confirm this, the problem should be rerun, using a finer mesh. If this is done with Δs reduced by half to 0.005 m, the time difference reduces to 1.6746 ns, corresponding to an error of around 0.41%, and confirming that numerical dispersion was indeed the cause of the problem.

The above results demonstrate a working code. If, however, one is not this fortunate, where does one look for the errors? The first thing to do is to ensure that the source really is working correctly. In MATLAB, the source was implemented as a function, in a separate m-file. This allows one to write a short test routine to see what the signal looks like. If this is correct, then the likely errors are in the scattered/total field equations. Be especially careful to ensure that the half-step offsets are correctly taken care of *in both space and time*.

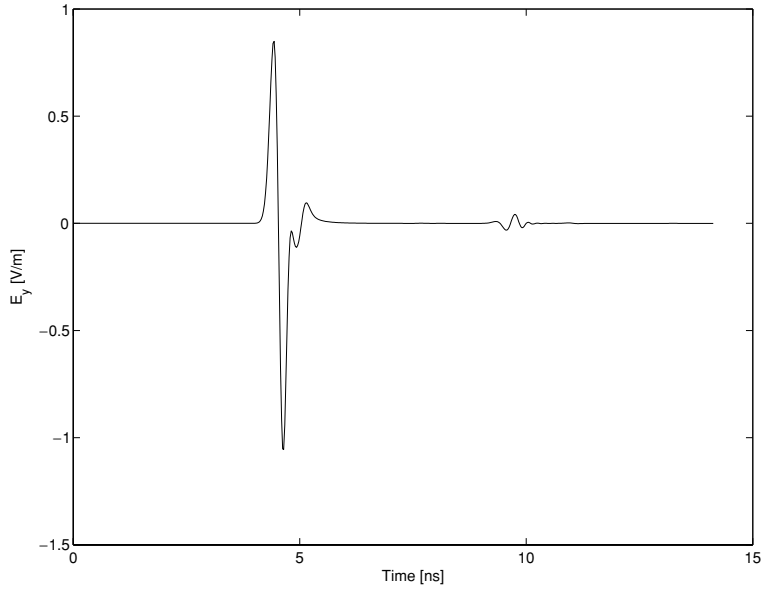


Figure 3.9 Gaussian derivative pulse and reflection.

Implementation of the ABC

Now that we have a code with working update equations, and can inject a plane wave into it, the PEC boundaries must be replaced with ABCs. An implementation note in passing: in order to test the ABCs, it is sufficient initially, using the plane-wave source in the previous section, to implement only the ABC at N_x, keeping PEC ABCs at the other boundaries. This permits one to get one set of ABCs working first.

Monitoring the signal at a location mid-way between the zone interface at $i = L$ and the right-hand boundary at $i = N_x$, the signal shown in Fig. 3.9 is recorded for $M = 600$. "Zooming-in" on the reflection, we see Fig. 3.10. The first (negative) peak, with a value of around -0.03 V/m, corresponds to the reflection of the first (positive) peak, which was around 0.8 (see Fig. 3.9), so the reflection coefficient of the ABC is around -30 dB.[5]

3.2.7 FDTD analysis of TE scattering from a PEC cylinder

Now that the basic FDTD code is working, we are in position to study TE_z scattering from a PEC cylinder. Again, we will work in the microwave region. A

[5] One could compute this more accurately but all that is needed at present is a "ball-park" figure. The correct method for numerically evaluating reflection off ABCs is to run two simulations, one using a reference solution computed on a very large grid, and the other a much smaller grid using the ABC. The reflection is then computed by subtracting the reference solution from the ABC-corrupted solution. We will do this when we evaluate the PML ABC later in this chapter.

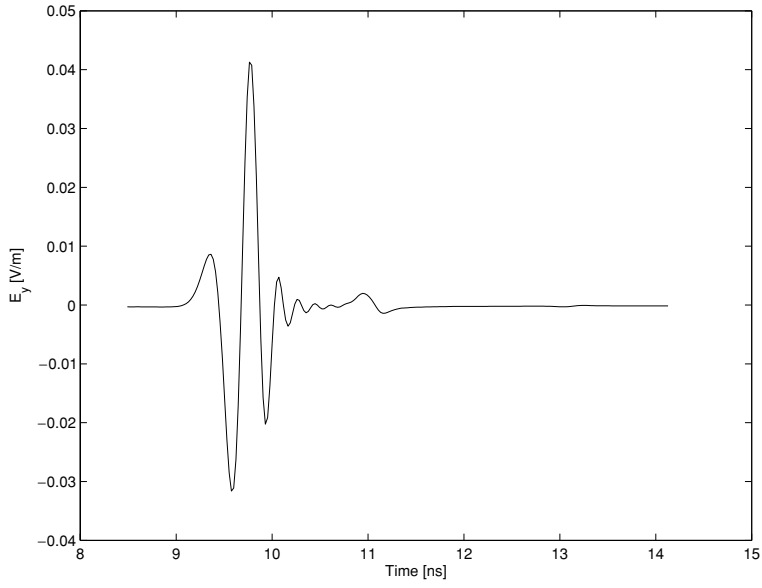


Figure 3.10 An enlargement of the signal in the region of the reflection.

convenient dimension will be a radius of $a = 0.03$ m. As a first pass, we will choose a rectangular domain, 2 m $\times$ 1 m (we will see shortly why we chose this). We will choose $\Delta s = 0.005$ m; this will allow a moderate approximation of the curvature of the cylinder. Even so, this means that the stair-step approximation of the cylinder will be quite crude – across the diameter of the cylinder there are only six cells – and we should bear this in mind when interpreting the results we will generate.

The simplest method of introducing the cylinder into the mesh is by simply zeroing the relevant $C(i, j)$ coefficients, see Eqs. (3.37) and (3.38). This ensures that the relevant electric fields inside and on the surface of the cylinder are zero. It is tempting to do the same with the magnetic fields; this however is incorrect, since it effectively also forces the tangential magnetic fields to zero at the cylinder's "surface," which is not the correct boundary condition.

We want to compute the echo width of the target, usually abbreviated σ_w. It is defined as follows:

$$\sigma_w = \lim_{\ell \to \infty} 2\pi \ell \frac{|E^{\text{scat}}|^2}{|E^{\text{inc}}|^2} \tag{3.42}$$

In an FDTD simulation, some finite limit on ℓ is essential. The conventional 3D criterion for establishing the onset of the far-field, viz. $\ell > 2D^2/\lambda$, where D is the largest dimension of the target, $D = 2a$ in this case, can be used. If we set $\ell \approx 1$ m, the minimum wavelength (and hence maximum frequency) at which this

still satisfies the far-field criterion is around 7 mm, or over 40 GHz, so this is more than adequate for our purposes.

We also now appreciate how convenient the scattered/total field formulation is; we can immediately obtain the scattered field by placing our sample point in the scattered field zone. Here, we have the following considerations: we would like to be as far away from the cylinder as possible, but since the reflected signal can be expected to be quite small, we should also be far away enough from the left-hand wall that we can "gate out" unwanted reflections – remember that our ABC is far from perfect. Since we are only going to look at *back-scattered* fields, we can place the scatterer to the right in our grid.

With these considerations in mind, then, we make the following choices.

- Locate the cylinder at $x = 1.5$ m, $y = 0.5$ m.
- Place the scattered/total field boundary at $x = 1$ m.
- Record the scattered field at $x = 0.5$ m, $y = 0.5$ m (i.e. 1 m away from the target, and 0.5 m away from the closest walls).

We will see shortly (Fig. 3.13) that TE_z back-scattering from a PEC cylinder increases rapidly with frequency up to a first resonance. This occurs when $ka \approx 0.8$, which for our cylinder with $a \approx 0.03$ corresponds to a frequency of just over 1 GHz. We also want to be able to capture the next resonances, so we need a signal with significant frequency content in this region. A differentiated Gaussian pulse with $\sigma = 5.0 \times 10^{-11}$ has a spectrum peaking at around 3.2 GHz, which will be adequate here. A longer pulse would work from the viewpoint of spectral content, but this shorter pulse is convenient for another reason we will see shortly.

Finally, we note that Eq. (3.42) is a frequency domain expression. The Fourier transforms of both the scattered and the incident fields must be computed, and divided pointwise.[6] Note also that this expression, being a power ratio, requires squaring the magnitude of the resultant transforms. (The phase information is irrelevant here.)

The back-scattered signal computed with the FDTD, with grid and problem as set up above, is shown in Fig. 3.11. Although we can go ahead and transform this, we should note that the main signal lies in the region 8–11 ns; the signal at 12 ns is almost certainly an unwanted reflection of some type. Similarly, the small "glitch" at 6 ns is also very likely to be some form of computational artifact. It is usual practice to remove these by "windowing" – although quite sophisticated windows exist, here it is sufficient simply to zero the signal outside this window. This is shown in Fig. 3.12. Finally, the echo width is plotted in Fig. 3.13, normalized by πa and compared to results computed using an exact eigenfunction solution [2, Figs. 12–34]. (The frequency axis is also normalized; this is usual practice with

[6] In MATLAB, the ./ operation.

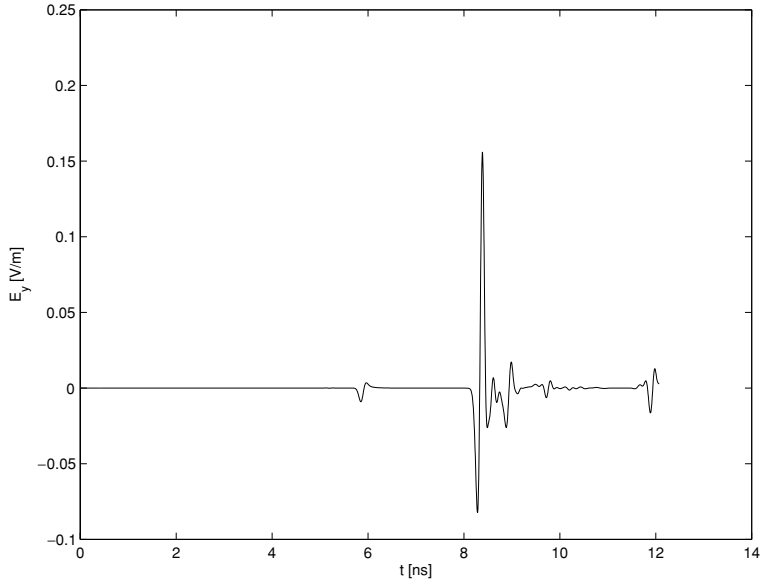


Figure 3.11 Back-scattered signal from the PEC cylinder. Medium mesh, $\Delta s = 0.0025$ m, $\sigma = 5 \times 10^{-11}$.

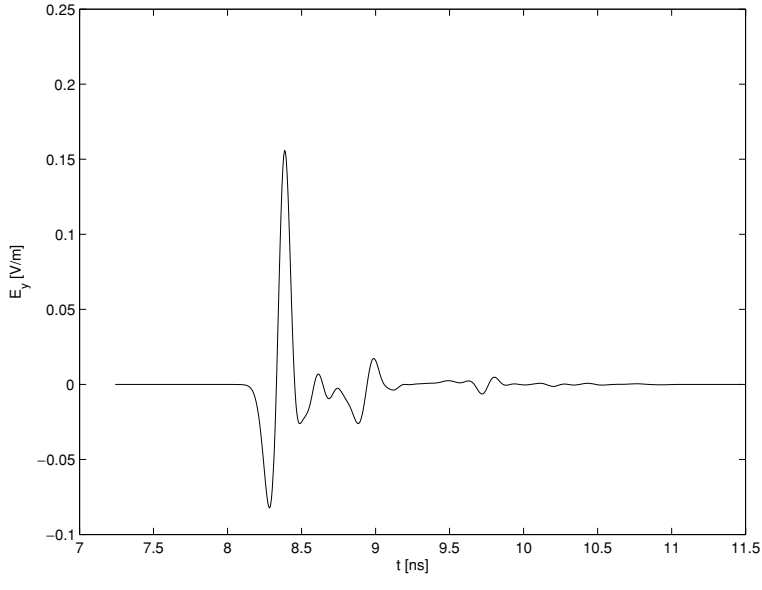


Figure 3.12 Windowed back-scattered signal from the PEC cylinder. Medium mesh, $\Delta s = 0.0025$ m, $\sigma = 5 \times 10^{-11}$.

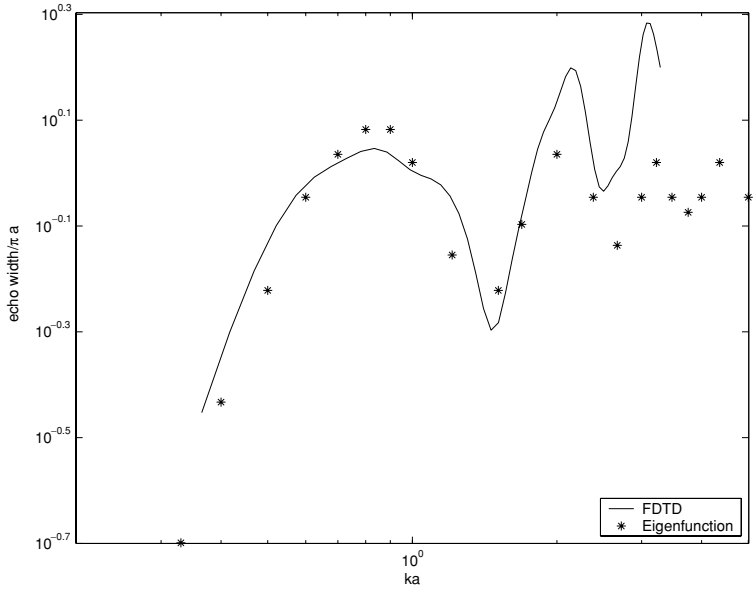


Figure 3.13 Normalized echo width for the PEC cylinder: FDTD results and eigenfunction solution. Medium mesh, $\Delta s = 0.0025$ m, $\sigma = 5 \times 10^{-11}$.

canonical shapes such as cylinders. k is the free-space wavenumber, and a the cylinder radius.)

The results in Fig. 3.13 show reasonable agreement at the first resonance, but the comparison is quite poor for the next resonances. To improve this, we first need to understand the *physics* of the scattering process. The first peak is simply energy which reflects directly off the cylinder, back in the direction of propagation. (This is the reflection which asymptotic methods, such as geometrical optics, would compute.) The next peak is due to energy which attaches itself to the top (bottom) of the cylinder, and "creeps" around the shadowed side of the cylinder before detaching itself from the bottom (top). Clearly, this signal travels a longer distance than the direct reflection; depending on the cylinder's size, it may reinforce the direct reflection or partially cancel it. This then accounts for the peak at around $ka \approx 2$. The extra distance traveled is $a + \pi a + a = (2 + \pi)a$; this signal travels at the speed of light which should result in a delay of about 514 ps. If we inspect Fig. 3.12, we can see these two signals; the (negative) peak of the direct reflection is at around 8.3 ns and the (same) peak of the creeping wave is at around 8.8 ns, i.e. around 500 ps apart. On this figure, there is another rather smaller signal, approximately 800 ps later; this is the creeping wave which has gone right around the cylinder for a second time. It travels an extra $2\pi a$, a slightly longer distance.

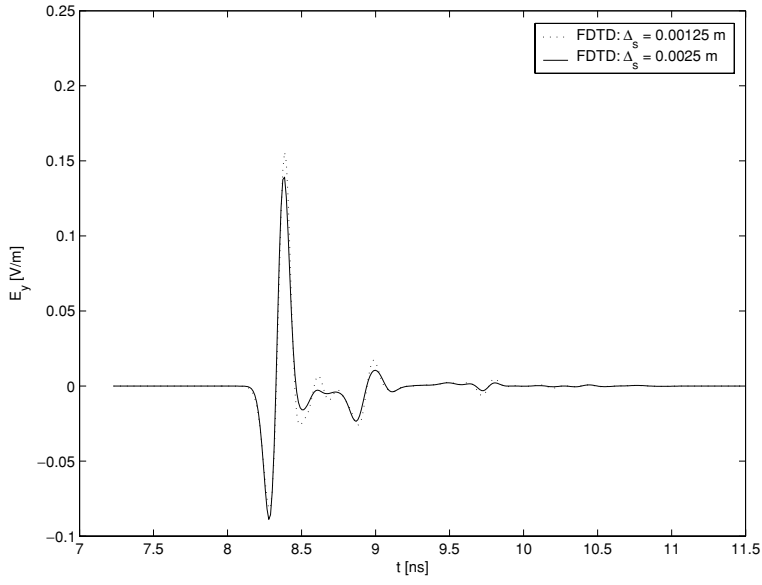


Figure 3.14 Windowed back-scattered signal from the PEC cylinder, comparing the medium and fine mesh solutions. For both solutions, $\sigma = 5 \times 10^{-11}$.

The problem is that the approximation of the round cylinder with the FDTD stair-step approximation is inadequate at higher frequencies; we need to refine the mesh. Time domain results comparing a finer mesh (with $\Delta s = 0.00125$) with the medium mesh above are shown in Fig. 3.14; note the better pulse shape in the finer mesh case. Results for both this finer mesh and for a coarser mesh ($\Delta s = 0.005$, and using a longer signal with $\sigma = 1 \times 10^{-10}$) are shown in Fig. 3.15. The eigenfunction data have been interpolated to make it easier to compare the respective results. The agreement is satisfactory, bearing in mind that although we satisfy the far-field criterion, ideally one should be a much larger distance from the scatterer. It is also clear that the solution will require an even finer mesh to get good agreement at the higher frequencies. We also note, perhaps surprisingly, that the coarse mesh solution appears to give a more accurate solution for the amplitude of the first resonance. However, we should bear in mind that we used a longer pulse, with lower spectral content, for this solution; the other meshes used pulses with peak spectra somewhat higher than this.

3.2.8 Computational aspects

One last aspect we should at least get an appreciation of before finishing this introduction to the two-dimensional FDTD is the question of the amount of computation required – and also the amount of computer storage needed. The most

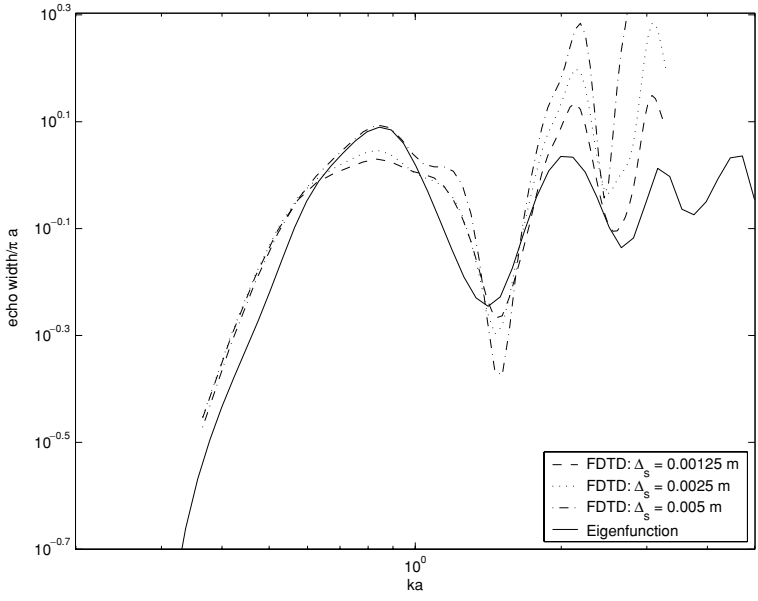


Figure 3.15 Normalized echo width for the PEC cylinder showing three different FDTD results compared with the eigenfunction solution.

computation is required in Eqs. (3.18)–(3.20), because each of these needs to be updated at all $N_x \times N_y$ points at each of M time-steps. All other operations, such as boundary conditions, sources via the scattered/total field interface etc., involve only *either* N_x or N_y points. Counting the number of operations, we see that to update the H_z field at each point requires five floating-point operations ($\pm$, $\times$), usually abbreviated as *flops*. (The shift operations on the field components are ignored in this type of count. The reason is that efficient computer languages recognize this type of operation and perform the shift indirectly by an offset in memory access.) The E_x and E_y field updates each require three flops. Thus, in total, the number of operations required per time step is approximately $11 \times N_x \times N_y$ flops. The overall number of operations is thus $11 \times N_x \times N_y \times M$ flops. If we also keep track of the run-time (which MATLAB allows us to do, the `cputime` command being one way of doing this, or one can simply use a stopwatch), the speed of the computer for floating point operations can be computed – often known as the floprate, and given as megaflops,[7] a million floating-point operations per second, or gigaflops (10^9 flops per second). Some very fast supercomputers are specified in terms of

[7] The results in this book were originally prepared largely on an IBM A31 notebook computer in 2003. The machine had a Pentium® 4, 1.8 GHz, 512 MB RAM. According to this test, the computer produced around 11.7 megaflops, which is quite slow; the clock speed on its own of course says little about especially floating-point speed. However, it is quite possible that an implementation in FORTRAN or C would be much faster; factors of two orders of magnitude are not unusual when converting code which does not readily vectorize in MATLAB to FORTRAN etc. In the present context, one would expect a less dramatic speed-up, given the vector nature of the update equations as coded.

teraflops (10^{12} flops per second) or even petaflops (10^{15} flops per second). Now it can be appreciated that halving the mesh size will increase the run time by a factor of $2^3 = 8$; to put this into practical terms, a run which took perhaps some minutes with one mesh may take an hour or so with a mesh twice as fine.

The analysis just performed leads to a field of computational science known as complexity analysis. What is of interest is the *asymptotic* computational cost of the algorithm. For CEM algorithms, this is usually performed on a square region, dimension d per side, with $\Delta x = \Delta y = \Delta s$ (or in 3D, a cube), thus $N_x = N_y = N$; furthermore, we note that the number of time-steps M is also essentially proportional to N. Hence the run-time is proportional to N^3 for the 2D FDTD algorithm; alternatively, we describe this as an $\mathcal{O}(N)^3$ algorithm.

This analysis in terms of number of unknowns is correct. Since N is inversely proportional to Δs, which in turn is often assumed to be inversely proportional to frequency f (via rules of thumb such as $\Delta s < 1/\lambda_{\min}$), the 2D FDTD algorithm is also often viewed as $\mathcal{O}(f)^3$ or equivalently, noting that $k_{\max}d$ is the size of the region in wavelengths,[8] $\mathcal{O}(k_{\max}d)^3$. This, however, is optimistic. The problem is that the assumption that Δs, and hence N, is directly proportional to $\lambda_{\min}$ is incorrect as the electromagnetic size of the problem increases. The reason is numerical dispersion in the FDTD grid. As an example, a phase error of 5% over a region of one wavelength results in around an 18° cumulative error, probably acceptable; the same percentage error over a region ten wavelengths in length will produce a cumulative error of 180°, clearly *unacceptable*. The dispersion error can be reduced by using a finer mesh. A more realistic assumption is that $N \propto (k_{\max}d)^{1.5}$; hence the 2D FDTD algorithm has an asymptotic complexity of $\mathcal{O}(k_{\max}d)^4$–$\mathcal{O}(k_{\max}d)^{4.5}$, depending on whether the number of time steps is assumed proportional to N or $k_{\max}d$. (One will find both in the literature.)

Regarding storage, the 2D FDTD does not make especially heavy demands on modern computers. The amount of storage required is the following, per cell:

- three field components – times two, for past and present;
- three material constants.

There are $N_x \times N_y$ cells, so the total storage required is $9N_x \times N_y$. In MATLAB, each real number is stored in double precision, requiring 8 bytes (most conventional languages, such as C and FORTRAN, permit the user to choose single or double precision). The storage in bytes is $72N_x \times N_y$. Of course, there are some other variables to store as well, but these are generally the largest. The fine mesh solution of the PEC discussed in the previous section required around 92 Mbytes of storage.

[8] Since $k_{\max} = 2\pi/\lambda_{\min}$.

3.3 The PML absorbing boundary condition

3.3.1 An historical perspective

By the early 1990s, the FDTD method had become very popular. However, the problem of terminating the mesh remained problematic. As we have seen, simple ABCs such as the first-order one already outlined only provide -20 to -30 dB of absorption, and then only close to normal incidence; whilst there were already better ones available, they were non-trivial to implement, and battled to provide more than -50 dB or so. By comparison, good anechoic chambers were able to provide 70 dB or more of dynamic range. Most of the work on ABCs had concentrated on analytical ABCs, using the properties of the wave operators. However, another type of absorber had also been experimented with – perhaps inspired by the pyramidal absorbers used in anechoic chambers. This was the use of absorbing material at the periphery of the mesh. As we will shortly see, a material with both electric and magnetic loss (carefully chosen in the correct ratios) can provide a perfect match, but only at normal incidence. The advantage of this is that the update equations do not need to be modified. Early efforts had achieved some success, but only worked well near normal incidence.

In 1994, Berenger published a truly seminal paper[9] [3]. His idea, like most really good ones, was in essence quite simple. He noted that the problem with artificial absorbers was their inability to operate over a wide range of incidence angles, and proposed that the solution was to increase the degrees of freedom available to provide the match. He proposed a method to do this in two dimensions, by "splitting" one of the field components in two – in the case of the TE_z problem we have investigated, it is H_z which is thus treated, viz. $H_z = H_{zx} + H_{zy}$ – and assigning *different* electric and magnetic loss to each component. Despite the initially worrisome nature of the split field, he showed that the result was what he called a perfectly matched layer (PML) which, in theory at least, absorbed incident waves of *all* polarizations, at *all* frequencies, and at *all* angles of incidence. Furthermore, the wave transmitted into the PML had the *same* wave speed as the incident wave, the *same* characteristic impedance, but *attenuated* (potentially rapidly) in the normal direction. All that was needed to implement the absorber was to modify the FDTD update equations in the PML region to accommodate the split field. Perhaps even more incredibly, "corner regions" of a mesh, which had long caused problems, could be treated by simply overlapping an x-attenuating and a y-attenuating PML.

This almost appeared to good to be true in 1994; within an extremely short time, the entire FDTD community identified the crucial importance of Berenger's work,

[9] In retrospect, ideas in CEM can often be attributed to several independent inventors, but his invention was unique and certainly deserving of the subsequent accolades. It is interesting that he appears to have published nothing on the FDTD in English language journals prior to this, although he had worked on ABCs before, publishing in French.

validated it independently, and quickly extended it to three dimensions. Furthermore, two different approaches were quickly introduced to avoid the split field formulation, whilst retaining the superb performance of the PML. The one approach used "stretched coordinates," and was independently introduced by Chew and Weedon [4] and Rappaport [5]; the other used an anisotropic medium with uniaxial permittivity and permeability tensors, and was introduced by Sacks *et al.* [6]; the latter approach is generally known as the UPML formulation (Uniaxial PML). The stretched coordinate formulation is rather mathematical in nature, but is very useful for other coordinate systems; the UPML, due to its physical plausibility (usually described as Maxwellian), is probably the most popular contemporary approach. Note that even the UPML material is nonetheless fictitious; however, Ziolkowski has investigated the physical realizabilty of such material [7].

In this chapter, we are going to use Berenger's original split field formulation. The reason is that it is both the simplest and also the most efficient approach *in two dimensions*. Using the UPML, for instance, requires introducing the electric and magnetic flux vectors, $\vec{D}$ and $\vec{B}$, which doubles the amount of storage required in the UPML region, whereas the split field formulation requires only one extra field component. Additionally, using the UPML requires that we deal with dispersive materials: although this is not too difficult to implement, it is additional complexity we choose to avoid now. It is important to note that this benefit accrues only in two dimensions; in three dimensions, there is little to choose between the formulations from the viewpoint of efficiency, since *all* fields must then be split in the Berenger approach. Furthermore, dispersive materials with the specific form required by the UPML can be quite efficiently handled by the FDTD. Should the reader want to undertake a three-dimensional implementation, a detailed discussion of the UPML approach is available in [8, Chapter 7] and would be the present author's recommendation.

3.3.2 A numerical absorber – pre-Berenger

Before discussing Berenger's contribution, we will review the case of a normally matched numerical absorber. Our presentation is based on Gedney and Taflove's approach [1, Chapter 7] and we very largely use their notation here. Firstly, we consider the case of a TE_z wave $\vec{H}^{\mathrm{inc}} = H_0 e^{-j(\beta_{1x}x+\beta_{1y}y)}\hat{z}$ incident on a half-space interface with an absorber at $x = 0$. Importantly, the (fictitious) absorber has *both* electrical (σ) and magnetic (σ^*) loss. The fields on the incident ($x < 0$) side, region 1, are the usual free-space fields:

$$\vec{H}_1 = H_0(1 + \Gamma\, e^{j2\beta_{1x}})\, e^{-j(\beta_{1x}x+\beta_{1y}y)}$$

$$\vec{E}_1 = \left[-\frac{\beta_{1y}}{\omega\epsilon_1}(1 + \Gamma\, e^{j2\beta_{1x}})\hat{x} + \frac{\beta_{1x}}{\omega\epsilon_1}(1 - \Gamma\, e^{j2\beta_{1x}})\hat{y} \right] H_0\, e^{-j(\beta_{1x}x+\beta_{1y}y)}$$

The fields on the transmitted ($x > 0$) side are:

$$\vec{H}_2 = H_0\tau\, e^{-j(\beta_{2x}x+\beta_{2y}y)}$$

$$\vec{E}_2 = \left[-\frac{\beta_{2y}}{\omega\epsilon_2(1+\frac{\sigma}{j\omega\epsilon_2})}\hat{x} + \frac{\beta_{2x}}{\omega\epsilon_2(1+\frac{\sigma}{j\omega\epsilon_2})}\hat{y} \right] H_0\tau\, e^{-j(\beta_{2x}x+\beta_{2y}y)}$$

Here, Γ and τ are the usual plane-wave reflection and transmission coefficients at the interface. These equations follow simply from the Maxwell equations, if a (fictitious) magnetic current and hence loss term are included in Faraday's law; they are generalizations of the case discussed by Balanis in [9, Section 5.4.2].

Similarly, the dispersion relationships are:

$$\beta_{1x} = k_1\cos\theta_i, \qquad \beta_{1y} = k_1\sin\theta_i, \qquad \forall x < 0$$

$$\beta_{2x} = \sqrt{k_2^2\left(1+\frac{\sigma}{j\omega\epsilon_2}\right)\left(1+\frac{\sigma^*}{j\omega\mu_2}\right) - (\beta_{2y})^2}, \qquad \forall x > 0 \tag{3.43}$$

with $k_i = \omega\sqrt{\epsilon_i\mu_i}$, $i = (1, 2)$.

Enforcing continuity of the tangential fields at the interface, $x = 0$, one obtains:

$$\Gamma = \frac{\frac{\beta_{1x}}{\omega\epsilon_1} - \frac{\beta_{2x}}{\omega\epsilon_2(1+\sigma/j\omega\epsilon_2)}}{\frac{\beta_{1x}}{\omega\epsilon_1} + \frac{\beta_{2x}}{\omega\epsilon_2(1+\sigma/j\omega\epsilon_2)}}$$

$$\tau = 1 + \Gamma$$

$$\beta_{2y} = \beta_{1y} = k_1\sin\theta_i$$

For normal incidence ($\theta_i = 0$), this simplifies to:

$$\Gamma = \frac{\eta_1 - \eta_2}{\eta_1 + \eta_2}$$

with

$$\eta_1 = \sqrt{\frac{\mu_1}{\epsilon_1}}\ , \quad \eta_2 = \sqrt{\frac{\mu_2(1+\sigma^*/j\omega\mu_2)}{\epsilon_2(1+\sigma/j\omega\mu_2)}}$$

Now, the core idea: *set* $\mu_2 = \mu_1, \epsilon_2 = \epsilon_1$ *and further, enforce*

$$\frac{\sigma^*}{\mu_1} = \frac{\sigma}{\epsilon_1} \Rightarrow \sigma^* = \sigma\mu_1/\epsilon_1 = \sigma(\eta_1)^2 \tag{3.44}$$

Then, $k_1 = k_2$, $\eta_1 = \eta_2$, and thus we obtain perfect absorption: $\Gamma = 0$. Also, very importantly,

$$\beta_x^t = \left(1+\frac{\sigma}{j\omega\epsilon_1}\right)k_1 = k_1 - j\sigma\eta_1 \tag{3.45}$$

and the transmitted fields in region 2 are

$$\vec{E}_2 = \eta_1 H_0 \, e^{-jk_1 x} e^{-\sigma \eta_1 x} \hat{y}$$
$$\vec{H}_2 = H_0 \, e^{-jk_1 x} e^{-\sigma \eta_1 x} \hat{z}$$

In summary, note the following important features of this solution.

- At normal incidence, there is *no* reflection at the interface: hence (at this angle at least) we have a *perfectly matched layer* (*PML*).
- The transmitted wave in the absorber has the same velocity as in region 1, but *attenuates* in the normal direction.
- Although lossy, the absorbing material is *dispersionless* (that is, the wave speed is independent of frequency).

3.3.3 Berenger's split field PML formulation

The previous fictitious absorber exhibits PML behavior *only at normal incidence*; its properties degrade rapidly off-normal. As discussed in the introductory comments, Berenger recognized that an additional degree of freedom would permit a match off-normal *as well*. He did this by "splitting" the transverse fields into two orthogonal components, for example $H_z = H_{zx} + H_{zy}$ in his notation. Associated with these were *two* components[10] of σ^* (σ_x^* and σ_y^*) and similarly, *two* components of σ (σ_x and σ_y).

Applying this to our previous two-dimensional TE problem, instead of the usual three equations in E_x, E_y and H_z – for example, as in Eqs. (3.5)–(3.7) – we now have four:

$$j\omega\epsilon_2 \left(1 + \frac{\sigma_y}{j\omega\epsilon_2}\right) E_x = \frac{\partial (H_{zx} + H_{zy})}{\partial y} \tag{3.46}$$

$$j\omega\epsilon_2 \left(1 + \frac{\sigma_x}{j\omega\epsilon_2}\right) E_y = -\frac{\partial (H_{zx} + H_{zy})}{\partial x} \tag{3.47}$$

$$j\omega\mu_2 \left(1 + \frac{\sigma_x^*}{j\omega\mu_2}\right) H_{zx} = -\frac{\partial E_y}{\partial x} \tag{3.48}$$

$$j\omega\mu_2 \left(1 + \frac{\sigma_y^*}{j\omega\mu_2}\right) H_{zy} = \frac{\partial E_x}{\partial y} \tag{3.49}$$

Introducing the variables

$$s_k = (1 + \sigma_k / j\omega\epsilon_2), \qquad s_k^* = \left(1 + \sigma_k^* / j\omega\mu_2\right), \qquad k = x, y \tag{3.50}$$

[10] In retrospect, *this* was the crucial idea, and the split field simply a mathematical device to accomplish this: clearly this defines an anisotropic medium of some type.

it may be shown that:

$$H_z = H_0 \tau \, e^{-j\sqrt{s_x s_x^*}\beta_{2x} x - j\sqrt{s_y s_y^*}\beta_{2y} y} \tag{3.51}$$

$$E_x = -H_0 \tau \frac{\beta_{2y}}{\omega \epsilon_2} \sqrt{\frac{s_y^*}{s_y}} e^{-j\sqrt{s_x s_x^*}\beta_{2x} x - j\sqrt{s_y s_y^*}\beta_{2y} y} \tag{3.52}$$

$$E_y = H_0 \tau \frac{\beta_{2x}}{\omega \epsilon_2} \sqrt{\frac{s_x^*}{s_x}} e^{-j\sqrt{s_x s_x^*}\beta_{2x} x - j\sqrt{s_y s_y^*}\beta_{2y} y} \tag{3.53}$$

with

$$(\beta_{2x})^2 + (\beta_{2y})^2 = (k_2)^2 \tag{3.54}$$

Clearly, these can be discretized using the central-differenced leapfrog Yee approach.

The phase-matching condition at the interface requires that the propagation constants in the y-direction are identical; this can be achieved if $s_y s_y^* = 1$, or equivalently $\sigma_y = \sigma_y^* = 0$. Thus, $\beta_{2y} = \beta_{1y} = k_1 \sin\theta_i$. Further, the H-field reflection coefficient may be shown to be:

$$\Gamma = \frac{\frac{\beta_{1x}}{\omega\epsilon_1} - \frac{\beta_{2x}}{\omega\epsilon_2}\sqrt{\frac{s_x^*}{s_x}}}{\frac{\beta_{1x}}{\omega\epsilon_1} + \frac{\beta_{2x}}{\omega\epsilon_2}\sqrt{\frac{s_x^*}{s_x}}}, \qquad \tau = 1 + \Gamma \tag{3.55}$$

Now, let $\epsilon_1 = \epsilon_2$, $\mu_1 = \mu_2$, and $s_x = s_x^*$. This is equivalent to $k_1 = k_2$, $\eta_1 = \sqrt{\mu_1/\epsilon_1} = \sqrt{\mu_2/\epsilon_2} = \eta_2$ and $\sigma_x/\epsilon_1 = \sigma_x^*/\mu_1$. Thus, from Eq. (3.54), $\beta_{1x} = \beta_{2x}$, and from Eq. (3.55), $\Gamma = 0$. The resultant TE_z field transmitted into the PML is then:

$$H_z = H_0 \, e^{-j\beta^{1x} x - j\beta_{1y}} \, e^{-\sigma_x \eta_1 \cos\theta_i x} \tag{3.56}$$

and similar expressions for E_y and E_z.

These have the same behavior as the previous normal-only PML, but attenuate *without dispersion* for *all* incident angles.

These results are so important that we will highlight them again in summary form.

- Theoretically, the PML absorbs incident waves of *all* polarizations, at *all* frequencies, and at *all* angles of incidence.
- Further, the wave transmitted into the PML has the *same* wave speed as the incident wave, the *same* characteristic impedance, but *attenuates* (potentially rapidly) in the normal distance.
- All that is needed to implement the absorber is to modify the FDTD update equations in the PML region. (Again, in retrospect what is required is the ability to handle a certain

type of lossy anistropic material; this at heart is why the update equations need to be modified.)

- Although perhaps not immediately clear from the above, a "corner region" of a mesh can be treated by simply overlapping an x-attenuating and a y-attenuating PML. This had long been a very troublesome problem with analytical ABCs.

We have already discussed the alacrity with which Berenger's idea was adopted in the FDTD community; within a few months, the PML had been extended to three dimensions by Katz, Thiele and Taflove [10]; Berenger himself also extended his formulation to three dimensions [11].

3.3.4 The FDTD update equations for a PML

With the theoretical background in place, we turn our attention to implementing and then testing a split field PML. The time domain equivalents of Eqs. (3.46)–(3.49) are

$$\left(\epsilon_2 \frac{\partial}{\partial t} + \sigma_y\right) E_x = \frac{\partial (H_{zx} + H_{zy})}{\partial y} \tag{3.57}$$

$$\left(\epsilon_2 \frac{\partial}{\partial t} + \sigma_x\right) E_y = -\frac{\partial (H_{zx} + H_{zy})}{\partial x} \tag{3.58}$$

$$\left(\mu_2 \frac{\partial}{\partial t} + \sigma_x^*\right) H_{zx} = -\frac{\partial E_y}{\partial x} \tag{3.59}$$

$$\left(\mu_2 \frac{\partial}{\partial t} + \sigma_y^*\right) H_{zy} = \frac{\partial E_x}{\partial y} \tag{3.60}$$

Compared to Eqs. (3.5)–(3.7), the loss terms bring a slight complication: we require the value of the electric field, for instance, at a half time-step, e.g. $E_x(i + \frac{1}{2}, j, n + \frac{1}{2})$, a point at which it is not available. (Note that this problem is due to the presence of loss, and not specifically because of the PML – even a normal material with finite electrical conductivity presents this problem.) A method widely used with success is the "semi-implicit"[11] approximation: the required value is computed as the arithmetic average of the previous (known) value and the as-yet-to-be-computed value, i.e.

$$E_x(i + \tfrac{1}{2}, j, n + \tfrac{1}{2}) = \frac{E_x(i + \frac{1}{2}, j, n + 1) + E_x(i + \frac{1}{2}, j, n)}{2} \tag{3.61}$$

[11] The FDTD method is an explicit method; "future" values are computed entirely from "present" and "past" ones. The approach discussed here uses a "future" value as unknown in the update equation, albeit itself, and hence the name "semi-implicit."

Using this approximation, and otherwise proceeding as before, the result is the following set of update equations:

$$H_{zx}(i,j,n) = D_{a_{Hzx}}(i,j)\cdot H_{zx}(i,j,n-1) - D_{b_{Hzx}}(i,j)\cdot [E_y(i+1,j,n) - E_y(i,j,n)] \tag{3.62}$$

$$H_{zy}(i,j,n) = D_{a_{Hzy}}(i,j)\cdot H_{zy}(i,j,n-1) + D_{b_{Hzy}}(i,j)\cdot [E_x(i,j+1,n) - E_x(i,j,n)] \tag{3.63}$$

$$E_x(i,j,n+1) = C_{a_{Ex}}(i,j)\cdot E_x(i,j,n) + C_{b_{Ex}}(i,j)\cdot [H_z(i,j,n+1) - H_z(i,j-1,n+1)] \tag{3.64}$$

$$E_y(i,j,n+1) = C_{a_{Ey}}(i,j)\cdot E_y(i,j,n) - C_{b_{Ey}}(i,j)\cdot [H_z(i,j,n+1) - H_z(i-1,j,n+1)] \tag{3.65}$$

where we have combined[12] the H field in Eqs. (3.64) and (3.65):

$$H_z(i,j,n) = H_{zx}(i,j,n) + H_{zy}(i,j,n) \tag{3.66}$$

and the material constants are defined as

$$C_{a_{Ex}}(i,j) = \frac{1 - \frac{\sigma_y(i,j)\Delta t}{2\epsilon_2(i,j)}}{1 + \frac{\sigma_y(i,j)\Delta t}{2\epsilon_2(i,j)}} \tag{3.67}$$

$$C_{b_{Ex}}(i,j) = \frac{\frac{\Delta t}{\epsilon_2(i,j)\Delta y}}{1 + \frac{\sigma_y(i,j)\Delta t}{2\epsilon_2(i,j)}} \tag{3.68}$$

$$C_{a_{Ey}}(i,j) = \frac{1 - \frac{\sigma_x(i,j)\Delta t}{2\epsilon_2(i,j)}}{1 + \frac{\sigma_x(i,j)\Delta t}{2\epsilon_2(i,j)}} \tag{3.69}$$

$$C_{b_{Ey}}(i,j) = \frac{\frac{\Delta t}{\epsilon_2(i,j)\Delta x}}{1 + \frac{\sigma_x(i,j)\Delta t}{2\epsilon_2(i,j)}} \tag{3.70}$$

$$D_{a_{Hzx}}(i,j) = \frac{1 - \frac{\sigma_x^*(i,j)\Delta t}{2\mu_2(i,j)}}{1 + \frac{\sigma_x^*(i,j)\Delta t}{2\mu_2(i,j)}} \tag{3.71}$$

$$D_{b_{Hzx}}(i,j) = \frac{\frac{\Delta t}{\mu_2(i,j)\Delta x}}{1 + \frac{\sigma_x^*(i,j)\Delta t}{2\mu_2(i,j)}} \tag{3.72}$$

[12] This is slightly more convenient to code. However, note that the split fields must be retained, and updated as usual before the next iteration.

$$D_{a_{Hzy}}(i,\,j) = \frac{1 - \frac{\sigma_y^*(i,j)\Delta t}{2\mu_2(i,j)}}{1 + \frac{\sigma_x^*(i,j)\Delta t}{2\mu_2(i,j)}} \tag{3.73}$$

$$D_{b_{Hzy}}(i,\,j) = \frac{\frac{\Delta t}{\mu_2(i,j)\Delta y}}{1 + \frac{\sigma_y^*(i,j)\Delta t}{2\mu_2(i,j)}} \tag{3.74}$$

As usual with an FDTD equation set, there are subtle differences between the otherwise very repetitive equations which one must be careful to code correctly. In particular, note that σ_x is associated with the E_y update (and vice versa), whereas σ_x^* and σ_y^* are associated with the H_{zx} and H_{zy} updates respectively.

3.3.5 PML implementation issues

One issue which one needs to decide upon when implementing an FDTD PML code is whether the PML update equations are going to be used throughout the entire computational domain, or whether different code will be written for each section. (By simply setting the conductivities to zero, the PML reduces to the usual update equations; alternatively, the electrical conductivity may be retained if required, etc.) The former has the advantage of being far simpler – and corner regions are very simply catered for automatically – but it does increase the memory requirement. The latter is far more tedious to code and the potential for coding error is much higher, but it is more memory efficient. In 2D, the overhead is only 33% in the non-PML regions, and since 2D FDTD codes are in any case not especially memory intensive, it is almost certainly better to use the PML update equations throughout. In 3D, however, the overhead is 100% in the non-PML regions, and the decision is not quite so straightforward. Bear in mind though that the PML works so well that the absorbing boundary can be brought quite close to the scatterer, reducing the memory required in any case.

Remember also that the exact positions of the material parameters are implied but not explicitly stated in the Eqs. (3.67)–(3.74); for example, in Eqs. (3.67) and (3.68), $\sigma_y(i,\,j)$ must be evaluated at $([i - \frac{1}{2}]\Delta_x, [j - 1]\Delta_y)$, the position of the relevant E_x field component; similarly, in Eqs. (3.69) and (3.70), $\sigma_x(i,\,j)$ must be evaluated at $([i - 1]\Delta_x, [j - \frac{1}{2}]\Delta_y)$; and in Eqs. (3.71)–(3.74), $\sigma_x^*(i,\,j)$ and $\sigma_y^*(i,\,j)$ must be evaluated at $([i - \frac{1}{2}]\Delta_x, [j - \frac{1}{2}]\Delta_y)$. (Note that H_{zx} and H_{zy} are located at the *same* grid point, the usual H_z location.) This implies of course that σ_x, σ_y, and the pair $\sigma_x^*; \sigma_y^*$ are always evaluated a half-grid point apart. Since the usual polynomial scaling results in quite rapidly changing conductivities, this is an important point to bear in mind for a high-performance PML.

Theoretically, the PML can be made as thin as desired by simply making the material extremely lossy. In practice, the FDTD discretization, with the accompanying half-cell offset, produces some "numerical" reflection. To ameliorate this, practical PML schemes use a number of FDTD cells to implement the absorber, with a "graded" loss profile, increasing from zero loss at the PML/free space interface to some maximum value at the boundary of the grid. A widely used profile is polynomial grading; for a PML of thickness d, the value of σ_x at depth x is

$$\sigma_x = (x/d)^m \sigma_{x,\max} \tag{3.75}$$

where $\sigma_{x,\max}$ is the maximum value attained at $x = d$. Typical practical PMLs are five to ten FDTD cells thick, with a polynomial order loss profile from two to four.

When discretized in an FDTD mesh, the discretization error produces a filtering effect, which produces some frequency dependence – typically low frequencies are not absorbed as well as higher frequencies.

Thus far, nothing has been said about suitable values for σ. An extensive series of numerical experiments has demonstrated that an optimal choice of this parameter for polynomial grading is

$$\sigma_{x,\max} = \frac{0.8(m+1)}{\eta \Delta s} \tag{3.76}$$

Usually, the external walls are treated as PECs for simplicity, i.e. the relevant tangential field is set to zero.

When implementing a PML, one needs to think carefully about the slight lack of symmetry in FDTD grids. As an example, consider σ_y in the layer of the cells with, on the one hand, $j = 1$ and on the other, $j = N_y$. Setting the tangential fields (E_x) to zero, the result is that in the layer of cells with $j = N_y$, there is no field, since the relevant E_x field component is "below" the last cell (in the geometry of Fig. 3.1). Thus, the value of σ_y in cell layer $j = 1$ actually corresponds to that in cell layer $j = N_y - 1$, rather than $j = N_y$. Also, once σ_y has been computed, it is tempting to find σ_y^* using $\sigma_y^* = \eta^2 \sigma_y$, but as we have already commented above, this is subtly incorrect due to the $\Delta s/2$ offset between electric and magnetic field points.

Coding hints – testing a PML

The first test to run with a PML is a free-space test: set all the conductivities to zero, which effectively reduces the PML to free space. Errors in the update equations will often quickly make themselves apparent without having to worry about whether conductivity profiles have been set correctly, for instance.

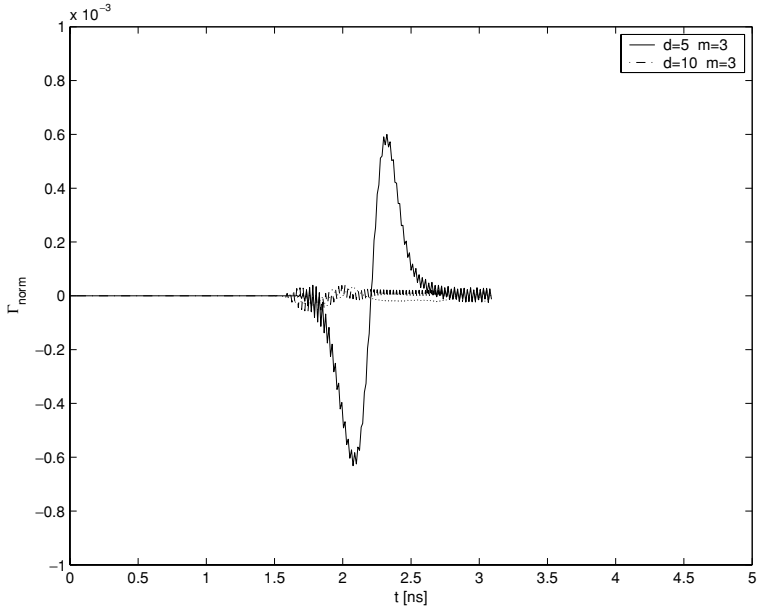


Figure 3.16 Normalized reflection from a split field PML.

3.3.6 Results for a split field PML

The PML performs so well as an absorber that trying to identify the reflection visually, as we did with the first-order ABC earlier, is impossible. The correct approach to testing a PML (or indeed any ABC) is to run two simulations, with identical discretization and source: one with the ABC under test, and another with a rather larger computational space. The signal is then compared at a point near the ABC. In this case, a 200×200 simulation was compared with a 400×400 simulation. The two signals cannot be distinguished on a graph, so on Fig. 3.16, the difference between the signals is shown – this is the reflection. Note the vertical scale. This has also been normalized by the signal peak, and further time-gated to remove double reflections, etc. When expressed in dBs in Fig. 3.17, the results are *deeply* impressive: the five cell thick, third-order polynomial grading PML has a maximum reflection of around -65 dB; the ten cell thick PML improves this to -85 dB.

Prior to the Berenger PML, the best ABCs were challenged to produce reflection coefficients significantly less than around -50 dB. As we have seen, the Berenger PML offers astounding performance – broadband reflection coeffecients *far* less than this are easily achieved, and with care (for example, optimized conductivity profile, double precision), absorptions of the order of -100 dB and significantly less have been obtained. The FDTD is in a position to out-perform very careful

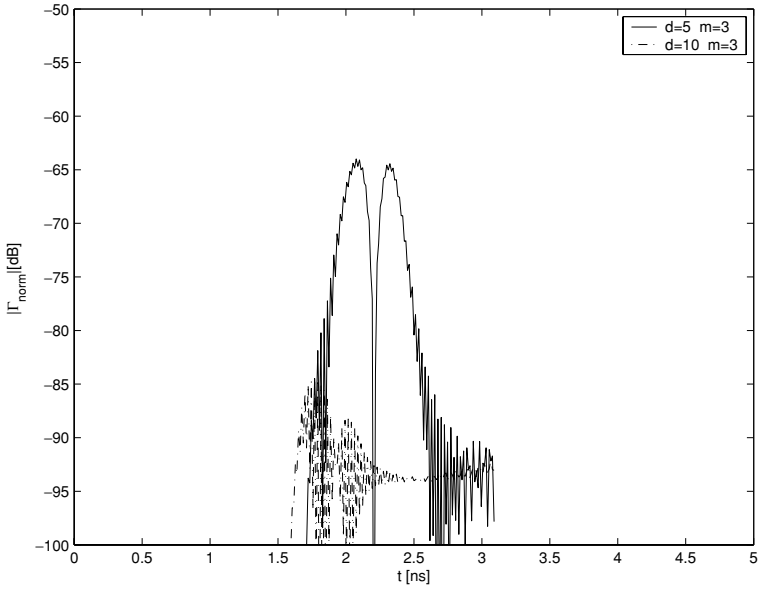


Figure 3.17 Normalized reflection from a split field PML, in dB.

measurements; as mentioned earlier, sophisticated anechoic chambers have dynamic ranges of around 70 dB but there is little prospect of dramatic improvements there.

3.3.7 Drawbacks of the Berenger PML

It may seem curmudgeonly to offer any criticism at all of such an innovation, but despite its superb performance, the PML has some drawbacks, especially in three dimensions. For 3D formulations, the PML requires that *all* field components be split, doubling the memory requirements in the absorbing region; with a 5–10 cell thick layer in 3D this can become a significant overhead. (Other formulations, such as the UPML, do not split the field but instead require the $\vec{D}$ and $\vec{B}$ fields to be stored as well in each cell, so the overhead is the same.) The Berenger PML is non-Maxwellian; the field splitting is a mathematical artifact which works very well but leaves niggling questions about physical reality. These drawbacks led to the investigation of other equivalent formulations, aiming to reproduce the superb performance of the PML with a (potentially) physical realizable material. This is also important for applications in FEM codes, where the splitfield formalism has no counterpart. Two approaches have emerged: the uniaxial anisotropic absorber and the stretched coordinate formulations. Although our implementation is the original split field one, we will briefly outline these other approaches.

3.3.8 Uniaxial absorber theory

A uniaxial material has the following tensor characterization:

$$\overline{\overline{\epsilon}} = \epsilon_1 \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & b \end{bmatrix}, \qquad \overline{\overline{\mu}} = \mu_1 \begin{bmatrix} c & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & d \end{bmatrix} \tag{3.77}$$

with $\vec{D} = \overline{\overline{\epsilon}}\vec{E}$ and $\vec{B} = \overline{\overline{\mu}}\vec{H}$. It has been shown that if the tensors are chosen as follows:

$$\overline{\overline{\epsilon}} = \epsilon_1 \overline{\overline{s}}, \qquad \overline{\overline{\mu}} = \mu_1 \overline{\overline{s}}, \qquad \overline{\overline{s}} = \begin{bmatrix} s_x^{-1} & 0 & 0 \\ 0 & s_x & 0 \\ 0 & 0 & s_x \end{bmatrix}$$

then a plane wave is completely transmitted (i.e. $\Gamma = 0$), independent of angle, frequency and polarization – a uniaxial PML (UPML).

The identity with Berenger's PML is reinforced with the choice:

$$s_x = 1 + \frac{\sigma_x}{j\omega\epsilon} \tag{3.78}$$

Note that this material is dispersive. This UPML and Berenger's split field PML have been shown to have the same propagation characteristics. The associated Gauss' laws are different (but irrelevant in an FDTD code, which discretizes only Ampère's and Faraday's laws).

The UPML can be discretized relatively simply in an FDTD fashion; the best source here is [1, Chapter 7]. However, instead of split fields, the $\vec{D}$ and $\vec{B}$ vectors must also be stored and updated in the PML region. As mentioned, the material is dispersive; fortunately, there are some elegant approaches available to deal with this [1].

3.3.9 Stretched coordinate theory

Another formulation shown to be equivalent is the "stretched coordinate" theory. The Cartesian coordinates (x, y, z) are mapped into complex space using

$$\tilde{x} \to \int_0^x s_x(x')\, dx' \tag{3.79}$$

and similarly y and z. Partial derivatives then become:

$$\frac{\partial}{\partial \tilde{x}} = \frac{1}{s_x}\frac{\partial}{\partial x} \tag{3.80}$$

and these are carried into the Maxwell equations. Stretched coordinates have been useful in extending the PML to cylindrical and spherical coordinate systems.

3.3.10 Further reading on PMLs

An excellent description by Gedney and Taflove may be found in [1, Chapter 7]. (A very similar treatment is also available in the slightly earlier [12, Chapter 5].) The treatment presented here is based on this approach. Gedney and Taflove well summarize the fervor with which the FDTD community adopted, expanded and generalized Berenger's work, and provide an extremely useful unified view of the original split field formulation, the UMPL and the stretched coordinate viewpoints, with a consistent notation. The original paper by Berenger remains interesting reading [3]. There are a very large number of papers on the subject of PML and the FDTD; the interested reader is referred to the extensive list of references in [1, Chapter 7]. One paper which is worth highlighting is Wittwer and Ziolkowski's contribution [13], since this discusses a number of practical issues in PML implementation.

3.3.11 Conclusions on the PML

Berenger's PML (and the related UPML) came close to putting the ABC "industry" out of business, at least in the FDTD community. Using the Berenger PML, a numerical absorber for the FDTD with essentially arbitrarily good performance can be produced. This has been extended to terminating conductive and/or dispersive regions, as well as half-spaces [1, Chapter 7]. There are still some detail issues to consider – although the basic formulation has been done, *details* for the PML in other coordinate systems are not always readily available.

The PML has some computational overhead and does complicate a code to some extent, whether one uses the split field, UPML or stretched coordinate formulations.

It should be commented that such superb absorption is not always required, and a simple ABC is sometimes sufficient, especially if combined with time-gating.

A final comment: the issue of high-performance numerical absorbers in FEM codes is *not* such a closed topic; UPML in an FEM mesh can wreck matrix conditioning and radically slow iterative solvers to the point of uselessness. With time domain FEM, the dispersive nature of the UPML is especially problematic.

3.4 The 3D FDTD algorithm

Extending the two-dimensional algorithm to three dimensions is straightforward from the viewpoint of the update equations. However, there is no analogy to the TM and TE modes, and all six field components must be updated. The field components are located on the full Yee cell. Again, the field components are offset in both space and time. Details are available in a number of texts. A good introduction is available in [2, Chapter 11]. For a very comprehensive study of the FDTD

method, including state-of-the art material, refer to [1]. We will not discuss the 3D FDTD algorithm further here, except to note the greatly increased computational cost associated with adding another dimension. The algorithm is now $\mathcal{O}(N)^4$, or $\mathcal{O}(k_{\max}d)^5$–$\mathcal{O}(k_{\max}d)^{5.5}$. Halving the mesh size increases the run time by a factor of 16, doubling the frequency, by between 32 and 45 or so (when numerical dispersion is correctly controlled as discussed previously).

In 3D, memory also starts becoming a serious issue; the storage requirements for the six field components (times two, for past and present) and the material arrays (in double precision) become $144N_x \times N_y \times N_z$ bytes. A computational volume with 100 cells on a side will require 144 MB. This will run on most contemporary personal computers (depending obviously on the amount of memory installed), but just doubling this to 200 cells increases the memory requirement to well over 1 Gbyte. This is well within the scope of most workstations, but beyond most PCs at the time of writing. Double precision is unnecessary for most applications, and one can save storage by storing an integer index rather than the material arrays as done here, but even so, the storage requirement grows very rapidly.

It is for these reasons that the development of efficient ABCs was so crucial as the enabling technology which permitted widespread adoption of the FDTD. Highly efficient ABCs permit one to place the scatterer very close to the boundary, and one can also obtain scattered fields very close to the boundary without unphysical reflections corrupting the fields.

We will not discuss the three-dimensional FDTD further, but rather turn now to the use of a commercial code which implements the FDTD.

3.5 Commercial implementations

Perhaps the most well-known commercial implementations of the FDTD are XFDTD and CST MICROWAVE STUDIO™ (MWS). The former is an implementation of the standard FDTD. The latter is actually a suite of codes, including a transient solver which uses the finite integration technique (FIT) [14, 15]; its predecessor was known as MAFIA and one may still encounter reference to this in the literature. Although apparently based on an integral equation approach to the Maxwell equations, for Cartesian grids the FIT can be rewritten as a standard FDTD method, and in the following we will use the term FDTD when discussing MWS.

It is worth commenting here that the FDTD method also sometimes uses finite integration methods, in particular for deriving subcellular models. The idea is the following. Referring back to Fig. 3.1, instead of writing the Maxwell equations in differential form, we will write them in integral form in this Yee cell. (As before, we will restrict ourselves to the TE_z mode here.) Specifically, we write Faraday's

Law on contour C, the boundary of the Yee cell:

$$\oint_C \vec{E} \cdot \vec{dl} = -\frac{\partial}{\partial t} \iint_A \mu \vec{H} \cdot \vec{dS} \tag{3.81}$$

Approximating the E_x and E_y components by their values at the Yee locations as in Fig. 3.1, and approximating H_z by its value in the center of the cell, one obtains:

$$\begin{aligned} -\frac{\partial}{\partial t}\mu H_z(i+\tfrac{1}{2}, j+\tfrac{1}{2})\Delta x \Delta y = {} & E_x(i+\tfrac{1}{2}, j)\Delta x + E_y(i+1, j+\tfrac{1}{2})\Delta y \\ & -E_x(i+\tfrac{1}{2}, j+1)\Delta x - E_y(i, j+\tfrac{1}{2})\Delta y \end{aligned} \tag{3.82}$$

Dividing by the area $\Delta x \Delta y$, and using the usual finite difference approximation in time for $\frac{\partial}{\partial t} H_z$, we obtain Yee's FDTD algorithm.

This form is especially useful when one wants to model fine geometrical features which are rather smaller than the Yee cell in the rest of the model, since the field behavior can be taken into account when performing the integral. (As a simple example, the quasi-static $1/r$ nature of the magnetic field near a thin wire is used to incorporate thin wires.) These are generally known as local subcell models.[13] Typical examples include thin sheets, better approximations of curved boundaries, thin wires, and thin cracks.

3.5.1 An introductory example – a waveguide "through"

The following is the first use of a commercial code in this book – in this case, MWS – and we will use this to highlight some important points about using an unfamiliar simulation tool.

Firstly, most packages nowadays ship with good documentation, usually with some form of "Getting Started" manual, or some variant on this theme, and time spent working through this type of manual is time very well spent indeed. Most simulators have some features and functions which are not immediately obvious, even if one is familiar with the method implemented, and the introductory manuals will often highlight these and save much time and subsequent frustration.

Secondly, even with the very best user interfaces – and MWS has a very impressive one – modelling complex three-dimensional geometries is not straightforward. One needs to try out simpler structures first, before attempting to model some complex device, quite possibly of unknown performance. Although MWS is at heart an FDTD code, the mesh is very largely invisible to the user. Model creation proceeds

[13] Another term often used is partially filled cells. Subcell is also sometimes used to describe submeshing, a method whereby a cell is divided into a number of smaller cells to improve accuracy.

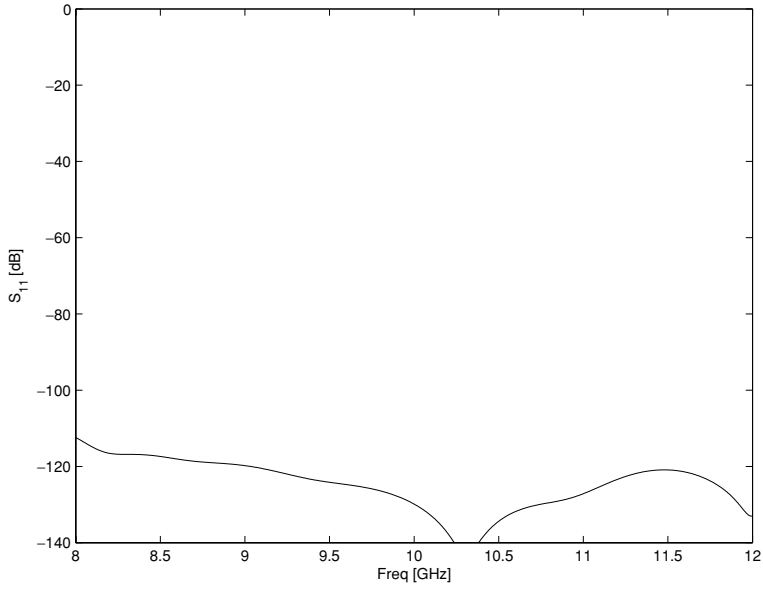


Figure 3.18 An MWS simulation of an empty piece of waveguide, showing an extremely low reflection coefficient as expected.

by defining geometrical primitives, which are then combined into more complex structures, before finally adding electrical parameters such as ports, field monitors etc. (One exception in MWS is the electrical and/or magnetic properties of materials, which are defined as needed during model building; in some other packages, this is only done once the geometrical model is finished.)

So, with the notes of caution in mind, before analyzing a real device, the first structure which we will simulate is an empty piece of waveguide. We will do this at X band (8.2–12.4 GHz), using a piece 40 mm long. (This is long enough to test the model without requiring a significant run-time.) In MWS, we create the waveguide using either of the pre-defined waveguide "templates." (Templates simplify generating particular types of frequently used models; in the case of a waveguide, for instance, the exterior region is set to PEC.) Then, the "brick" primitive is used to generate the length of waveguide (the standard cross-section inside dimensions are 22.86 mm $\times$ 10.16 mm). Finally, the "pick face" function is used twice, to assign waveguide ports to each end of the length of waveguide. Since the waveguide is empty, the magnitude of the transmission coefficient should be unity, and the reflection coefficient zero. A result is shown in Fig. 3.18.[14] The reflection coefficient

[14] Throughout this book, results have generally been plotted from MATLAB, using data computed by the relevant program, so as to provide some visual unity. Most programs provide a command to export data to some type of neutral file format.

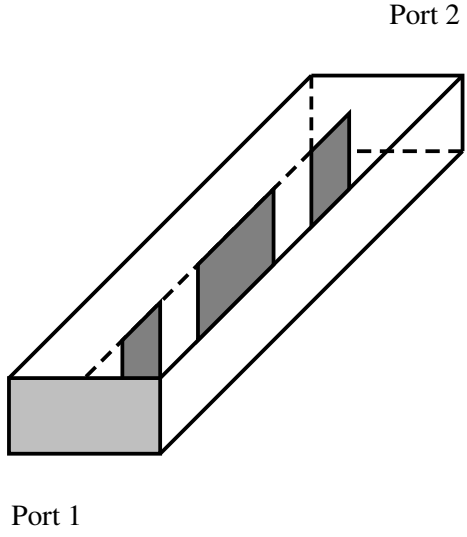


Figure 3.19 The waveguide filter geometry, showing the metallic septa (not to scale).

is less than −100 dB across the band, showing excellent performance, and giving confidence in basic modelling and simulation setup.

3.5.2 A waveguide filter

With some confidence that one has basic modelling skills with a particular package, one can turn to more interesting and challenging problems. Again, we will use a waveguide example, but now a more complex double-pole filter. The following example was originally designed by Meyer and van der Walt [16]. This X-band waveguide filter consists of three metal septa along its center, normal to the broad walls of the waveguide. The smaller septa are each 6.556 mm in length, and the longer is 16.788 mm. The inter-septa spacing is 12.148 mm. The septa are 0.2 mm thick. See Fig. 3.19 for a sketch of the filter.

When dealing with waveguide discontinuities, one of the first things one must note is that only the dominant waveguide mode should be present at the ports. In this case, an extra section of empty guide, 23.9 mm, was added, but any similar value would be acceptable. (The evanescent modes dampen exponentially, and at 10 GHz, the guide wavelength is around 40 mm, so the above length is around one-half a guide wavelength, more than sufficient.)

The modelling process in MWS is very similar to that already discussed in the previous introductory example, although here the “waveguide filter” template is chosen. (This sets some internal analysis parameters which are optimized for highly resonant structures.) The septa inside the waveguide are added quite easily

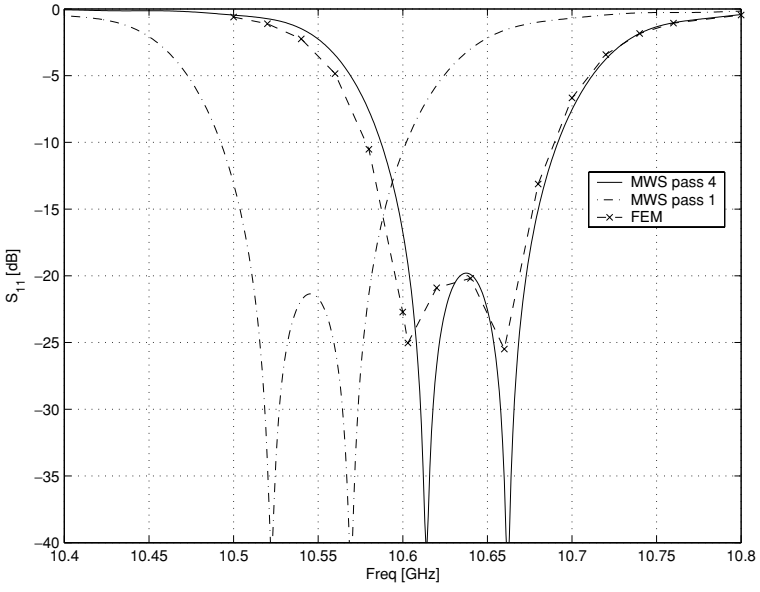


Figure 3.20 An MWS simulation of the waveguide filter in the text.

using the "brick" primitive. (There are various ways of doing this; using the "working coordinate system" – a local coordinate system which can be easily repositioned – which the package supports can simplify this.)

In this case, the initial results were not especially accurate. The reason is that a filter relies on resonances and anti-resonances for its operation, and these must be computed extremely accurately for good overall accuracy. MWS offers an adaptive mesh facility, which automatically refines the mesh in regions it determines. Using this option provides a much more accurate result in this case. In Fig. 3.20, three results are shown: MWS pass 1 is the result after one solution; MWS pass 4 is the result after four adaptive passes have been undertaken; and the FEM results were computed using FEMFEKO, an experimental FEM program that will be described in Chapter 10, using complete second-order vector elements.[15] Clearly, the FEM results and pass 4 are in excellent agreement. For this filter, measured data are also available; the measured center frequency was 10.47 GHz. This is an example of the difference we have already discussed between the approximate field problem (which these two different techniques have solved with great accuracy, the difference in center frequency being less than 0.1%) and the actual problem (both analyses differ from the measured result by about 2%); the difference is very likely due to manufacturing tolerances.

[15] More details on the FEM simulation may be found in Section 10.9; this solution had an average edge length of 3.0 mm, with 4968 tetrahedral elements and 41 526 degrees of freedom.

3.5.3 A microstrip patch antenna

FDTD codes can also be applied to antennas, provided a suitable ABC is available. MWS offers a PML-based ABC; as we have seen, this is a very accurate mesh truncation technique. An "antenna on planar substrate" template is available, although for an accurate model we will have to work a little harder. One important point which one must bear in mind is that with the FDTD, the substrate will not be of infinite extent, unless we use a suitable boundary condition to simulate this. This is different to the simulations we will discuss in Chapter 8, which use a form of the method of moments which includes stratified media in the formulation.

The particular patch we will analyze is discussed in some detail in Section 8.2; here, we will only give dimensions. It is 31.18 mm $\times$ 46.75 mm in size, on a substrate 2.87 mm thick with $\epsilon_r = 2.2$. The patch is fed via a pin (diameter 1.3 mm), offset by 8.9 mm from the center of the long edge, to provide a match close to 50 Ω.

In MWS, there are two ways to simulate such an antenna. The first uses a "discrete port." This is an approximation of a real feeding region, and implements either a voltage, current or "S-parameter" source (the last being a current source with internal impedance, which is needed when computing S-parameters). It amounts to forcing a field value at a point (or points) in the mesh. Since it is not a particularly accurate model of a physical source, there will be limitations on the accuracy expected, but it is fast to model and also more rapid to compute. If using a discrete port, the model is almost trivial to build: one defines the substrate using, once again, the "brick" primitive, then adds the patch, defines the discrete port at the appropriate offset location and runs the simulation. The only point which can cause some delay, in particular for users used to MoM codes, is that all structures in MWS have finite thickness – MoM codes usually work with infinitely thin metallic sheets. For the patch, a typical metalization thickness would be 25 μm, although the value is really not critical. MWS uses an elegant subcell model, known as the perfect boundary approximation [17] so that thin metal sheets do *not* have to comprise a full FDTD cell.

A more accurate model of the patch uses a coaxial feed and waveguide port. One way to do this is to add explicitly a ground plane of PEC of finite thickness, in which the coaxial feed will be embedded. (For reasons of internal code operation, MWS recommends that the length of the coaxial feed should be several times the thickness of the substrate; in this case, a length of 10 mm was chosen.) When adding the coaxial feed pin, one needs to be careful, since one is adding structures in regions where material already exists. In this case, it is easiest first to add the outer dielectric coaxial region cutting through the ground plane (and to use the same dielectric filler as the substrate material) and then to add the PEC inner conductor, which extends to become the feed pin. Although in general different

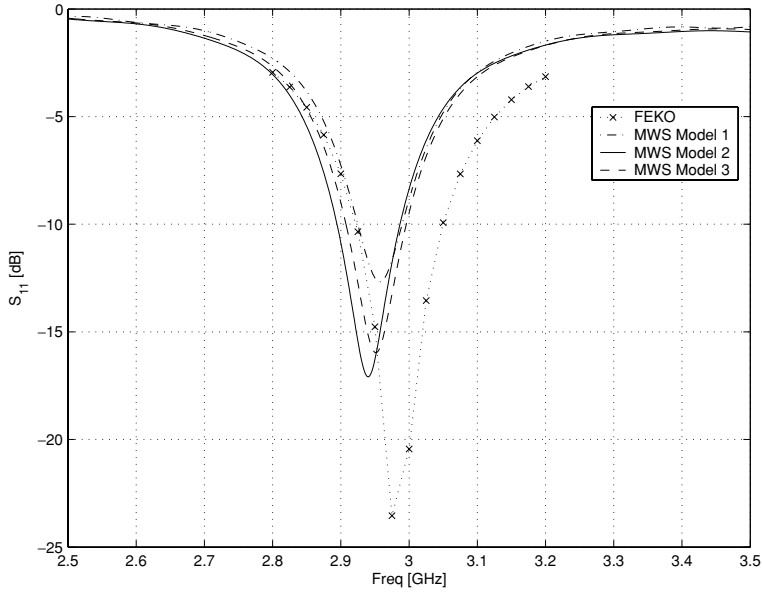


Figure 3.21 An MWS simulation of a microstrip patch antenna. FEKO results are also shown for comparison.

materials cannot be defined in the same geometrical region,[16] MWS permits PECs and dielectrics to coexist, but the region is effectively treated as perfectly conducting.

An alternative approach is to use a thin ground plane, and construct a coaxial cable on the reverse side.

Results for two such models are compared to a FEKO computation in Fig. 3.21. There are actually three MWS results in the plot. Model one used the discrete port approach, and a 100 mm $\times$ 100 mm substrate, using open boundaries on the substrate sides, an open boundary with additional space above the patch, and an electric boundary on the ground plane. Model two used the same substrate and boundary treatment, but a full coaxial feed model.[17] Model three used the same coaxial feed model and boundary treatment, but with a smaller substrate, 50 mm $\times$ 50 mm in size; the results are very similar to those of model two, indicating that the open boundary is simulating an infinite substrate quite well. All models were also run through the adaptive meshing process. The agreement between all three models and the FEKO computation is good; the discrete port model indicates the least

[16] One makes use of various Boolean operations to combine, intersect, etc. such overlapping regions to resolve this.

[17] The outer diameter of the coaxial feed, i.e. the region penetrating the ground plane, was chosen to give $Z_0 = 50\ \Omega$. For a coaxial cable of course, $Z_0 = \frac{60}{\sqrt{\epsilon_r}} \ln(b/a)$ with b and a the outer radius and inner radius respectively. In this case, with the same dielectric constant as the substrate, the outer radius was 2.24 mm.

good match, but the results are still quite usable. The difference in center frequencies between all four analyses is less than 1%. As we will see frequently in this book, this is a commonly encountered limit in CEM for resonant antenna models, unless tremendous care is taken with the model. (It is worth commenting that manufacturing and material tolerances will often render this moot in any case.)

Modelling hints – open boundaries and MWS

MWS has two types of open boundaries, both simulated using the PML, and the difference between them is subtle. Although we did not discuss this, PMLs can also terminate a region with two different materials; see, for example, [1, Section 7.10]. An *open boundary* in MWS places a PML at the plane indicated, permitting the code effectively to continue the substrate indefinitely. An *open boundary* (*add space*) does much the same, but adds some additional (free) space first; hence, this will *not* produce an infinite substrate.

Modelling hints – parametric modelling

Many CEM codes now permit one to "parametrize" the model. This means that instead of entering an actual length as the model is constructed, one instead defines this as a parameter which can then be changed subsequently. (We will see extensive use of this type of feature in Chapter 5.) MWS offers this capability, although we did not use it in these examples.

3.6 Further reading

The FDTD literature is truly massive, and a search on any of the electrical engineering databases will produce more hits than one will be able to process. One's first reference should be Taflove and Hagness [1], which provides encylopedic coverage of most aspects of the FDTD. The Schneider–Schlager FDTD database (see Appendix F) is also a very valuable resource.

We have only touched the surface of the modelling possibilities of the FDTD method. There are a whole number of issues which one can still address. Here follows just a selection of these.

- Our 2D example already indicated that the rectangular cells of the standard FDTD method may not approximate curved geometries very well. Methods of improving fine geometrical detail are generally known as "subcell" models, and usually rely on an equivalent formulation of the FDTD in terms of Faraday's and Ampère's laws, as briefly introduced in Section 3.5. See [1, Chapter 10] for more on this topic. Thin wires are

another type of structure which do not fit into the Yee grid very well. Bingle, the present author and Cloete describe a formulation incorporating finitely conducting wires in [18].

- When dealing with wideband pulses, one should appreciate that many materials cannot be represented accurately by a fixed value of ε_R. Again, elegant methods have been developed for dealing with materials with frequency-dependent material parameters; this is discussed in detail in [1, Chapter 9].
- For larger scatterers, it is extremely inefficient to try to position a field point in the far field. Formulations are available to compute the far field from a near field time domain computation, which permits one to use a much smaller mesh. See [1, Chapter 8] for details.
- Non-linear problems can *only* be addressed using time domain methods. A considerable amount of work has been done using the FDTD for such materials, including work at optical frequencies. FDTD codes have also been hybridized with circuit simulators to include non-linear devices (e.g. diodes). [1, Chapters 9 and 15] addresses these issues.
- We have discussed one-, two- and three-dimensional formulations of the FDTD. There is another interesting formulation, suitable for rotationally symmetric structures: the body of revolution FDTD. (This has been described as a two-and-a-half dimensional formulation; the full three-dimensional fields are computed, but using a two-dimensional grid for each Fourier mode present – for some problems, only one such mode is needed.) A discussion of this may be found in [1, Chapter 12]. The present author and Ziolkowski also used this formulation for modelling optical wave phenomena; in [19], we presented the formulation. Rather importantly, the correct numerical stability criterion (the Courant limit) for this case is also given in this paper.
- The FDTD can also be used for handling periodic structures. The present author, Smith and van Tonder used this for modelling frequency selective surfaces [20]. The treatment by Maloney and Kesler [1, Chapter 13] provides an up-to-date account of the formulations available in this context.
- Another type of boundary condition of interest is the complementary operator. Ramahi has worked extensively on this, and a summary may be found in [1, Chapter 6]. Work also continues on other types of ABCs for the FDTD; see, for instance, [21].
- A recently (re-)discovered algorithm, the alternating direction implicit (ADI) formulation of the FDTD method, permits one to exceed the Courant limit, but retain stability. The ADI-FDTD method does pose some other challenges [22].

3.7 Conclusions

Our treatment of the FDTD method, which started out in the previous chapter with a very simple 1D transmission line problem, solved essentially in the frequency domain, continued in this chapter with a quite sophisticated 2D simulation, incorporating wideband pulses, absorbing boundary conditions, and a physical analysis of scattering in the resonance regime[18] in both the time and frequency

[18] The region in which the dimension(s) of the scatterer are on the order of several wavelengths at most.

domains, and finished with some examples computed using the commercial package MWS.

We have also looked at computational issues, both run-time and memory, which impact on our ability to perform useful FDTD simulations. Berenger's PML has been introduced, and its extraordinary performance demonstrated. The 3D FDTD was briefly outlined; theoretically, there are no new issues to understand, but in practice writing a 3D code is challenging, since it needs to be very efficient in order to handle realistic problems (in 2D, far less optimal code can still be useful). Furthermore, for good results one should ideally use some of the more advanced FDTD approaches, in particular subcellular models and better modelling of curved boundaries. Unless one is fortunate enough to have access to an existing 3D FDTD code, such codes are generally best left to experts unless one has a very specific application in mind. The commercial code we discussed, MWS, provides a powerful implementation of the FDTD, offering (amongst other advanced modelling features) thin sheets, and a method called "perfect boundary approximation" which is essentially a type of subcell formulation improving geometrical modelling. It also features a user interface which at the time of writing was state-of-the-art. Other commercial FDTD codes are also available.

The FDTD has truly become the workhorse of CEM computation over the last decade – even when it is not necessarily the best technique to use! In the next chapter, we introduce the method of moments, which is a very powerful method for dealing with highly conducting structures, and often more efficient for these applications than the FDTD method.

References

[1] A. Taflove and S. Hagness, *Computational Electrodynamics: the Finite Difference Time Domain Method*. Boston, MA: Artech House, 2nd edn., 2000.

[2] W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design*. New York: Wiley, 2nd edn., 1998.

[3] J.-P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys.*, **114**, 185–200, October 1994.

[4] W. C. Chew and W. H. Weedon, "A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates," *Microwave Opt. Technol. Guided Wave Lett.*, **7**, 599–604, September 1994.

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

[6] Z. S. Sacks, D. M. Kingsland, R. Lee, and J. F. Lee, "A perfectly matched anisotropic absorber for use as an absorbing boundary condition," *IEEE Trans. Antennas Propag.*, **43**, 1460–1463, December 1995.

[7] R. W. Ziolkowski, "Time-derivative Lorentz materials and their utilization as electromagnetic absorbers," *Phys. Rev. E*, **55**, 7696–7703, 1997.

[8] A. Taflove, *Computational Electrodynamics: the Finite Difference Time Domain Method*. Boston, MA: Artech House, 1995.
[9] C. A. Balanis, *Advanced Engineering Electromagnetics*. New York: Wiley, 1989.
[10] D. S. Katz, E. T. Thiele, and A. Taflove, "Validation and extension to three dimensions of the Berenger (PML) absorbing boundary condition for FD-TD meshes," *IEEE Microwave Guided Wave Lett.*, **4**, 268–270, August 1994.
[11] J.-P. Berenger, "Three-dimensional perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys.*, **127**, 363–379, 1996.
[12] A. Taflove, *Advances in Computational Electrodynamics: the Finite Difference Time Domain Method*. Boston, MA: Artech House, 1998.
[13] D. C. Wittwer and R. W. Ziolkowski, "How to design the imperfect Berenger PML," *Electromagnetics*, **16**, 465–468, 1996.
[14] T. Weiland, "A discretization method for the solution of Maxwell's equations for six-component fields," *Electron. Commun. (AEÜ)*, **31**, 116–120, 1977.
[15] T. Weiland, "Time domain electromagnetic field computation with finite difference methods," *Int. J. Numerical Model.*, **9**, 295–319, 1996.
[16] P. Meyer and P. W. van der Walt, "Design of narrowband E-plane waveguide filters," in *Proceedings of the 1988 SAIEE 2nd Joint AP-MTT Symposium, Pretoria, South Africa*, 26.1–26.11. August 1988.
[17] B. Krietenstein, R. Schuhmann, P. Thoma, and T. Weiland, "The Perfect Boundary Approximation technique facing the big challenge of high precision field computation," in *Proceedings of the XIX International Linear Accelerator Conference (LINAC 98)*, pp. 860–862. Chicago, IL, 1988.
[18] M. Bingle, D. B. Davidson, and J. H. Cloete, "Scattering and absorption in rectangular waveguides: FDTD simulation," *IEEE Trans. Microwave Theory Tech.*, **50**, 1621–1627, June 2002.
[19] D. Davidson and R. W. Ziolkowski, "Body-of-revolution finite-difference time-domain modelling of space-time focusing by a three-dimensional lens," Special Issue on 3D Scattering, *J. Opt. Soc. Am. A*, **11**, 1471–1490, April 1994.
[20] D. B. Davidson, A. G. Smith, and J. J. van Tonder, "The analysis, measurement and design of frequency selective surfaces," in *Proceedings of the 10th International Conference on Antennas and Propagation*, vol. 1, 1.156–1.160. Edinburgh: IEE, April 1997.
[21] R. E. Díaz and I. Scherbatko, "A simple stackable re-radiating boundary condition (rRBC) for FDTD," *IEEE Antennas Propag. Mag.*, **46**, 124–130, February 2004.
[22] S. W. Staker, C. L. Holloway, A. U. Bhobe, and M. Piket-May, "Alternating-direction implicit (ADI) formulation of the finite-difference time-domain (FDTD) method: algorithm and material dispersion implementation," *IEEE Trans. Electromagn. Compat.*, **45**, 156–166, May 2003.

4

A one-dimensional introduction to the method of moments: thin-wire modelling

4.1 Introduction

The method of moments – MoM – was one of the first numerical methods to achieve widespread acceptance in electronic engineering for the analysis of antennas and scatterers. It is generally defined as a method for reducing an integro-differential equation to a set of linear equations. The origins of the method are old; as was already indicated in Chapter 1, some of the early work was done over a century ago. One of the widely used integral equation formulations still used for the analysis of thin wires (that due to Pocklington) was first presented in 1897 (although he used a series expansion method, rather than the modern segmentation approach). The first publications in the antenna and propagation professional literature were in the early 1960s, and some of the canonical papers (those of Harrington, Richmond, Mei and Andreasen) appeared at much the same time as Yee's paper. The specific name "method of moments" was introduced by Harrington in his early work, and the name caught on quickly; this was perhaps unfortunate, since the name has a slightly different meaning in contemporary applied mathematics. In that field, and also fields such as computational mechanics, the term "method of weighted residuals" is generally used for what has become known as the MoM in radio-frequency engineering. Another term widely used in other fields of engineering is "boundary element method"; for highly conducting structures, this term and the MoM as used in electromagnetics are synonymous.[1]

Primarily for two reasons, the MoM rapidly achieved widespread acceptance. Firstly, to a generation of engineers and scientists trained on analytical methods, along the lines of Harrington's classic text on time-harmonic fields [1] [2] – which in turn was based on methods of mathematical physics, as expounded by Stratton [2]

[1] We return to the topic of nomenclature in the penultimate section of this chapter.
[2] Originally published in 1961, but reprinted since.

and Morse and Feshbach [3] – the method was clearly based on sound electromagnetic theory and more generally, methods of mathematical physics, in particular variational calculus (which was then in widespread use). Secondly, because the method discretized *only* the metallic wires or surfaces of the antennas, it was far more efficient than methods such as the FDTD for analyzing the relatively small – typically resonance regime – antenna structures which were then the main topic of research. (As we have seen, the FDTD requires the discretization of all space surrounding the antenna or scatterer.) Furthermore, many problems then of current research interest could be solved used the MoM in a reasonable time – this was far less true of the FDTD, whose requirements for memory and computer time could generally not be accommodated on 1960s era computers.

In this chapter, we will present an introduction to the MoM, starting with an extremely simple electrostatic example. Again, as with the FDTD, the simple physics and geometry permit us to illustrate a number of core ideas without becoming overwhelmed by implementation details. Following this, we will extend the discussion to electrodynamics. Thin-wire modelling uses locally one-dimensional basis functions, but for general wire geometries, one must of course take the full three-dimensional geometry into account, and hence writing one's own MoM program for any reasonably interesting engineering problem is well beyond the scope of an introductory book of this nature. Fortunately, there are some excellent commercial implementations of the MoM, as well as one very useful public domain code; these are the topics of Chapter 5.

4.2 An electrostatic example

The problem we will address as an illustration of the MoM is the charge distribution $\rho(z)$ on a perfectly conducting straight thin wire, of radius $\rho = a$, charged to a potential V volts relative to ground. It is based on an example presented in [4, Chapter 12]. The wire could, for instance, be charged by induction. It is important to note that this is the *opposite* of typical work in introductory courses in electromagnetics, where $\rho(z)$ is given and one must then establish the potential (and hence field). Given $\rho(z)$, $V(\vec{r})$ (and hence $\vec{E} = -\nabla V$) is easily found:

$$V(\vec{r}) = \frac{1}{4\pi\epsilon_0} \int_V \frac{\rho(\vec{r'})}{R(\vec{r}, \vec{r'})} dV' \tag{4.1}$$

with

$$\begin{aligned} R(\vec{r}, \vec{r'}) &= |\vec{r} - \vec{r'}| \\ &= \sqrt{(x - x')^2 + (y - y')^2 + (z - z')^2} \end{aligned}$$

The primed coordinates ($\vec{r'}(x', y', z')$) are those of the source point. The field point coordinates are $\vec{r}(x, y, z)$.

However, our problem now is, given the voltage on the wire, to establish how the charge distributes itself. (In passing, we note that it cannot be a uniform distribution; the charges near the ends would clearly experience an unbalanced electrostatic force which would push them towards the ends of the wire.) This falls into the general class of *inversion* problems, and cannot generally be solved in *closed form*, i.e. analytically. A numerical approach is the only general solution method for such problems.

Before we proceed further, some terminology: Eq. (4.1) is known as an *integral equation*; the part inside the integral operator is frequently called the *kernel*. The function $V(\vec{r})$ is the *forcing function*. Two other concepts that are central to this theory is that the physical environment surrounding the radiator/scatterer (in this case, free space, i.e. an infinitely large and empty vacuum) *and* the boundary conditions are all included in the formulation. This is what permits the MoM to solve typical antenna problems (at least those involving perfect or highly conducting conductors) very efficiently. We will later encounter *Green functions*; it is these that effectively take the environment surrounding the structure into account, but they are only available for a very limited number of environments.

The critical idea is that Eq. (4.1) is valid *everywhere* – including on the wire itself, *where* $V(x, y, z)$ *is known*. This is the *boundary condition* (BC) for the problem. The idea that we will pursue to solve this problem is to approximate the charge by a number of simple functions, of unknown amplitude, which we will then find by assembling a matrix equation representing the geometry of the model and the BCs in discrete form.

4.2.1 Some simplifying approximations

Before we proceed further with the MoM solution of this problem, we will make a number of assumptions, which will considerably simplify the solution process.

- Equation (4.1) contains a volumetric integral. If we assume that the wire is a perfect electrical conductor (PEC), the charge is restricted to the surface and becomes a *surface charge* $\rho_s(z, \rho = a, \phi)$. (Note that we use cylindrical coordinates here, and that ρ refers both to the radius in this coordinate system and to the charge. The meaning will be clear from the context.)
- Secondly, we will simplify the geometry, by assuming a $\hat{z}$-directed wire.
- Thirdly, we will assume that the charge distribution is *uniform* in the circumferential direction, i.e. we can simply write $\rho_s(z, \rho = a, \phi) = \rho_s(z, \rho = a)$. This permits us to approximate further the *surface* charge $\rho_s(z, \rho = a)$ by an equivalent *line* charge, $\rho_l(z) = 2\pi a \rho_s(z, \rho = a)$, placed on the $\hat{z}$-axis.

Using these approximations, the integral equation Eq. (4.1) becomes:

$$V(z, \rho = a) = \frac{1}{4\pi\epsilon_0} \int_0^\ell \frac{\rho_\ell(z')}{R(z, z')} dz' \tag{4.2}$$

with (a is the wire radius):

$$\begin{aligned} R(z, z') &= \sqrt{[(x - x')^2 + (y - y')^2] + (z - z')^2} \\ &= \sqrt{a^2 + (z - z')^2} \end{aligned}$$

Note that we now write $V(z, \rho = a)$, rather than $V(\vec{r})$, since V is restricted to be on the wire surface (where the boundary is) and is rotationally invariant by assumption.

4.2.2 Approximating the charge

Up to this point, the approximations have been in the mathematical formulation (the integral equation). Now, we introduce the MoM as a method of approximately solving this equation. The wire, of length L, is broken up into N segments, using $N + 1$ nodes, defined as follows:

$$z_n = (n - 1)\Delta, \qquad n = 1, 2, \ldots, N + 1 \tag{4.3}$$

$$\Delta = \frac{L}{N} \tag{4.4}$$

In the following, "segment n" will mean the segment located between z_n and z_{n+1}.

The charge is approximated as

$$\rho(z') \approx \sum_{n=1}^{N} a_n h_n(z') \tag{4.5}$$

Here, a_n are unknown (but constant) coefficients, and $h_n(z')$ are *basis functions* – also often known as *expansion functions*. (Many texts use $f_n(z')$, but we want to reserve f and g for a specific purpose, discussed later in this chapter.) An example, with $N = 5$, is shown in Fig. 4.1. (Note that this is the solution obtained *after* the procedure to be discussed has been performed and the unknown coefficients obtained.) Equation (4.2) thus becomes:

$$V(z) = \frac{1}{4\pi\epsilon_0} \int_0^\ell \frac{1}{R(z, z)} \left[\sum_{n=1}^{N} a_n h_n(z') \right] dz' \tag{4.6}$$

Basis functions

The choice of the basis function is one of the most crucial parts of the MoM. A large variety of possible basis functions exists. Popular choices include functions

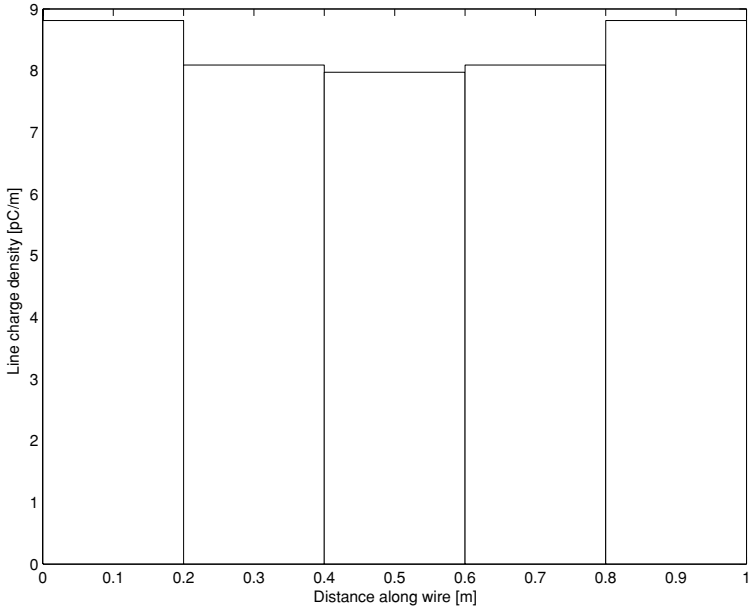


Figure 4.1 Equivalent line charge density for an $N = 5$ segment MoM solution using piecewise constant basis functions. $L = 1$ m, $V = 1$ V, $a = 0.001$ m.

with the following spatial variation: constant (also known as pulse or stair-step); linear; polynomial; piecewise sinusoidal; etc. Although deficient in some aspects, we will chose pulse basis functions for our introductory example. Each function is defined as:

$$h_n(z') = \begin{cases} 0 & \forall\ z' < (n-1)\Delta \\ 1 & \forall\ (n-1)\Delta \le z' \le n\Delta \\ 0 & \forall\ n\Delta < z' \end{cases} \tag{4.7}$$

In other words, the nth function is unity in one segment (segment n) and zero elsewhere.

Using these pulse basis functions in Eq. (4.5), and interchanging the order of integration and summation, one obtains:

$$4\pi\epsilon_0 V(z) = a_1 \int_0^{\Delta} \frac{h_1(z')}{R(z,z')} dz' + a_2 \int_{\Delta}^{2\Delta} \frac{h_2(z')}{R(z,z')} dz' + \cdots$$
$$+ a_N \int_{(N-1)\Delta}^{N\Delta} \frac{h_N(z')}{R(z,z')} dz' \tag{4.8}$$

This is one equation in N unknowns, viz. $\{a_1, a_2, \ldots, a_N\}$. To obtain a unique solution, one requires N equations, or *constraints*.[3]

[3] Strictly speaking, these must be *linearly independent* equations.

4.2.3 Collocation

To provide these N constraints, we enforce (match) the boundary condition at N points along the wire, z_m; this is also described as *testing* (sampling) $V(z, \rho = a)$. This method is called *collocation* or *point-matching*. It is convenient to locate these points in the middle of each segment, in between the nodes:

$$z_m = (n - 1/2)\Delta, \qquad m = 1, 2, \ldots, N \tag{4.9}$$

Note that *unlike* the FDTD method, this "half-point" offset has no adverse effect on the accuracy of the method, and is not essential to its implementation; sampling points at other locations within the segment would also work, this is merely convenient.

Sampling Eq. (4.6) at each of these N points, the following set of N equations is obtained:

$$\begin{aligned}
4\pi\epsilon_0 V(z_1) &= a_1 \int_0^{\Delta} \frac{h_1(z')}{R(z_1, z')} dz' + \cdots + a_N \int_{(N-1)\Delta}^{N\Delta} \frac{h_N(z')}{R(z_1, z')} dz' \\
&\vdots \\
&\vdots \\
4\pi\epsilon_0 V(z_N) &= a_1 \int_0^{\Delta} \frac{h_1(z')}{R(z_N, z')} dz' + \cdots + a_N \int_{(N-1)\Delta}^{N\Delta} \frac{h_N(z')}{R(z_N, z')} dz'
\end{aligned} \tag{4.10}$$

4.2.4 Solving the system of linear equations

The above set of equations is a *system* of linear equations. At this point, it is important to appreciate that the original *integral equation* inversion problem has now been reduced to a *matrix equation* inversion problem. It can be written as

$$\{V\} = [Z]\{I\} \tag{4.11}$$

sometimes known as generalized network parameters. Square braces indicate a matrix, curled braces a vector. The relevant entries are:

$$\begin{aligned}
V_m &= 4\pi\epsilon_0 V(z_m) \\
I_n &= a_n \\
Z_{mn} &= \int_{(n-1)\Delta}^{n\Delta} \frac{1}{[(z_m - z')^2 + a^2]^{1/2}} dz'
\end{aligned} \tag{4.12}$$

The n subscript refers to source points; m refers to testing (sampling) points. Symbolically, the solution is

$$\{I\} = [Z]^{-1}\{V\} \tag{4.13}$$

However, a linear system, usually written in the form $[A]\{x\} = \{b\}$, is *almost never* solved by inverting the matrix explicitly. Instead, the matrix $[A]$ is factored into the product of lower and upper triangular matrices:

$$[A] = [L][U] \tag{4.14}$$

Hence $[L][U]\{x\} = \{b\}$. An auxiliary vector $\{z\} = [U]\{x\}$ is introduced, and then $[L]\{z\} = \{b\}$ is solved by *forward substitution* to yield $\{z\}$; finally, $\{x\}$ is solved from $\{z\} = [U]\{x\}$ using *backward substitution*. (This process, an extension of Gaussian elimination, is generally covered in introductory undergraduate courses in numerical analysis.)

There are a number of reasons for pursuing this rather than direct inversion of the matrix; the most important is that solving a linear system using LU-factorization has a cost $\sim\mathcal{O}(N^3)$, whereas inverting a matrix costs at least twice this, since following the factorization N forward and backward substitutions are required, each of cost $\sim\mathcal{O}(N^2)$.

Before the matrix equation can be solved, however, there is still one issue to attend to. In Eq. (4.12), the term Z_{mn} is given as an integral over the nth segment. This usually has to be done numerically using *quadrature* (numerical integration). In this *specific* case, analytical results are available [4, p. 674]:

$$Z_{mn} = \begin{cases} 2\ln\left(\frac{\Delta/2+\sqrt{a^2+(\Delta/2)^2}}{a}\right) & \forall m = n \\ \ln\left(\frac{d_{mn}^{+}+\sqrt{(d_{mn}^{+})^2+a^2}}{d_{mn}^{-}+\sqrt{(d_{mn}^{-})^2+a^2}}\right) & \forall m \neq n \text{ but } |m-n| \leq 2 \\ \ln\frac{d_{mn}^{+}}{d_{mn}^{-}} & \forall |m-n| > 2 \end{cases} \tag{4.15}$$

with

$$\begin{aligned} d_{mn}^{+} &= l_m + \Delta/2 \\ d_{mn}^{-} &= l_m - \Delta/2 \\ l_m &= \sqrt{[(m-n)\Delta]^2 + a^2} \end{aligned} \tag{4.16}$$

The last parameter is the distance between the mth matching point and the center of the nth source segment.

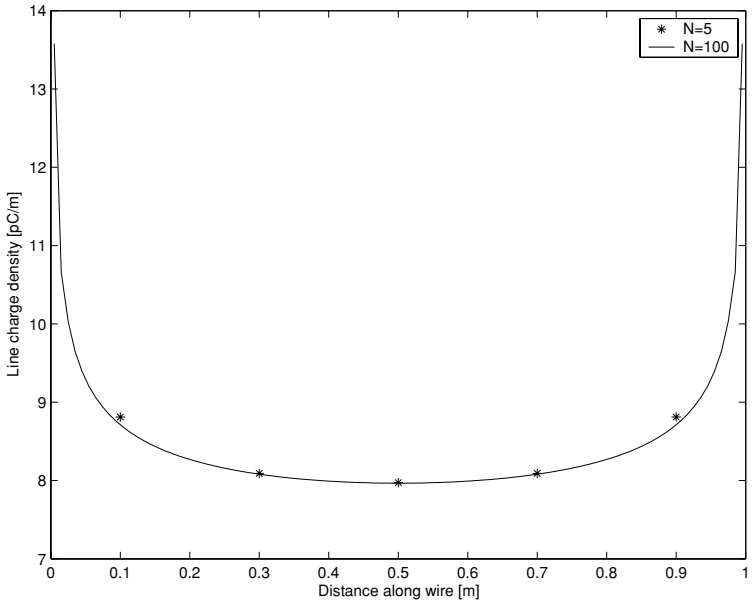


Figure 4.2 Comparison of 5 segment and 100 segment solutions.

4.2.5 Results and discussion

Results are shown in Figs. 4.1 and 4.2. In Fig. 4.1, the piecewise constant nature of the basis function has been explicitly shown (the `bar` command in MATLAB provides a simple way of doing this). In Fig. 4.2, one observes that the $N = 5$ solution is surprisingly accurate, although of course it does not correctly predict the behavior of the charge at the ends of the wire.

A number of approximations have been made in this development. These include the following, with the implications indicated.

- An equivalent line charge was assumed. This relied on a *rotationally symmetric* charge distribution. For a thin wire, this is generally a very good approximation.
- The ends of the wire were ignored; for instance, was the wire a hollow or solid tube? Again, for thin wires, this is a reasonable approximation.
- In the collocation process, the integrals (which represent the boundary conditions) were only exactly enforced at N discrete points. In between these points, the potential will depart from the specified value. Fortunately, using more (i.e. smaller) segments will reduce the impact of this.
- The specific basis function that was chosen – constant – is *discontinuous* at segment ends. Since we were approximating charge, which is continuous, this is non-physical. This is clearly evident in Fig. 4.1. (Again, the impact of this can be mitigated using smaller segments.)

- We assumed that the surface of the wire was perfectly conducting, so that the wire was an equipotential surface. For most good conductors, this is a very good approximation.

The reason that we are discussing these in detail is that all these comments also apply to electrodynamics.

4.3 Thin-wire electrodynamics and the MoM

With these basics behind us, electrodynamics (or *full-wave* behavior) can now be investigated. The ideas of incident and scattered field decomposition are important here. Other than this, and the more complex equations, we will find the overall process very similar indeed.

4.3.1 The electrically thin dipole

The problem that we now want to solve is the current distribution $I(z)$ on a straight thin wire. It is assumed here that the basics of the dipole radiator have already been studied. In such introductory courses on electrodynamics, some *assumption* is generally made regarding the distribution. For very short dipoles, a linear or even constant approximation of current can yield quite good results, and for the typical resonant dipole, the widely assumed sinusoidal distribution also produces useful results. However, the most obvious information which cannot be thus obtained is the *reactance* of the dipole.

Although the overall process is very similar to the electrostatic charge distribution problem just worked out, there are two important differences. Firstly, the boundary condition: for a perfect electric conductor, the boundary condition is:

$$\vec{E}_{\tan} = 0 \tag{4.17}$$

We will use the incident/scattered field decomposition method. This was already introduced with the FDTD. To revise this briefly: since the Maxwell equations are linear, the fields may be decomposed into an *incident* field $\vec{E}^{\text{inc}}$ and a *scattered* field $\vec{E}^{\text{scat}}$. The overall field, called the *total* field $\vec{E}^{\text{tot}}$, is then:

$$\vec{E}^{\text{tot}} = \vec{E}^{\text{inc}} + \vec{E}^{\text{scat}} \tag{4.18}$$

By definition, the incident field is the field which would exist if the scatterer were absent. As an example, if the incident field is a plane wave, propagating in the x-direction, in free space, with a z-polarized electric field, the expressions for the incident fields are:

$$\vec{E}^{\text{inc}} = e^{-jkx}\hat{z} \tag{4.19}$$

$$\vec{H}^{\text{inc}} = -\frac{1}{\eta_0}e^{-jkx}\hat{y} \tag{4.20}$$

As usual, $\eta_0 = \sqrt{\mu_0/\varepsilon_0}$ is the wave impedance of free space, and $k = 2\pi/\lambda_0$ is the wavenumber. It is of interest to compare these expressions to those used in Section 3.2.3. The main difference is of course that these expressions are *frequency domain* ones. (A rather more minor difference is that the electric field is polarized in the $\hat{z}$-direction rather than $\hat{y}$.) On the surface of a PEC wire, $\vec{E}_{\text{tot}} = 0$. The boundary condition on the surface of the wire thus becomes

$$\vec{E}^{\text{inc}} = -\vec{E}^{\text{scat}} \tag{4.21}$$

As indicated above, $\vec{E}^{\text{inc}}$ typically has a simple form. The scattered fields, $\vec{E}^{\text{scat}}$, must be computed from the surface current.

In general, the electric field can be computed from the magnetic vector potential $\vec{A}$ and electric scalar potential Φ as

$$\vec{E} = -j\omega\vec{A} - \nabla\Phi \tag{4.22}$$

It will be recalled that various *gauges* can be applied to these potentials.[4] The Lorenz[5] gauge is widely used in this context:

$$\nabla \cdot \vec{A} = -j\omega\mu_0\epsilon_0\Phi \tag{4.23}$$

Applied now to the $\hat{z}$-directed surface current source, and assuming that the wire is in free space, so that ϵ, μ and the wavenumber k have the usual values in vacuum[6] this becomes

$$\frac{\partial A_z}{\partial z} = -j\omega\mu_0\epsilon_0\Phi \tag{4.24}$$

Hence,

$$E_z^{\text{scat}}(r) = -j\frac{1}{\omega\mu_0\epsilon}\left(k^2 A_z + \frac{\partial^2 A_z}{\partial z^2}\right) \tag{4.25}$$

with

$$A_z = \frac{\mu_0}{4\pi}\int_{-l/2}^{l/2}\int_0^{2\pi} J_z(\phi', z')\frac{e^{-jkR}}{R} a\, d\phi'\, dz' \tag{4.26}$$

We have used the "free-space Green function" here ($\psi(z, z') = \frac{e^{-jkR}}{4\pi R}$), which gives the resulting magnetic vector potential for a current element.[7] R is the distance from source to field point coordinates. Substituting Eq. (4.26) in Eq. (4.25),

[4] The potentials are not unique, and contain elements of arbitrariness, which the gauging resolves.
[5] More properly attributed to L. Lorenz than H. Lorentz.
[6] This formulation is actually valid in any linear, isotropic and uniform medium, with μ and ϵ taking the appropriate values. For simplicity, we show only the free-space case.
[7] A more detailed discussion of Green functions is deferred to Chapter 7.

and integrating over the source region, one obtains

$$E_z^{\text{scat}}(r) = \frac{1}{j\omega\epsilon_0}\int_0^{2\pi}\int_{-l/2}^{l/2}\left[\frac{\partial^2\psi(z,z')}{\partial z^2} + k^2\psi(z,z')\right]J_z(\phi',z')a\,d\phi'\,dz' \tag{4.27}$$

Note that the differentiation in Eq. (4.25) has been taken inside the integral operator. This is valid since the differentiation is with respect to the field point coordinates, and the integration is over the source points.

At this stage, the unknown is still the $\hat{z}$-directed (by assumption) surface current $J_z(\phi', z')$. For sufficiently thin wires, this can be reduced to the *Pocklington* equation, first introduced in 1897:

$$\begin{aligned} E_z^{\text{scat}}(r) &= \frac{1}{j\omega\epsilon_0}\int_{-l/2}^{l/2}\left[\frac{\partial^2\psi(z,z')}{\partial z^2} + k^2\psi(z,z')\right]I_z(z')\,dz' \\ &= -E_z^i(r) \end{aligned} \tag{4.28}$$

This equation is obtained by assuming that (as for the electrostatic case), we locate the filament on the axis and enforce the boundary condition on the surface (the reciprocal case is sometimes more convenient in deriving this). Although it looks fairly straightforward, the presence of the second derivative of z inside the integral kernel, acting on the Green function, makes this non-trivial to implement. A useful further simplification can be made if the wire is assumed *very* thin ($a \ll \lambda$):

$$\int_{-l/2}^{l/2} I_z(z')\frac{e^{-jkR}}{4\pi R^5}\left[(1+jkR)(2R^2-3a^2)+(kaR)^2\right]dz' = -j\omega\epsilon_0 E_z^i(\rho=a) \tag{4.29}$$

with a the wire radius and $R = \sqrt{a^2+(z-z')^2}$. This is now a convenient form to program. It appears in numerous texts (for example, [4, p. 720]) and appears to have been first introduced by Richmond [5] (reprinted in [6]).

Further discussion on these and other integral equations (such as Hallén's) may be found in [4, 7].

Before solving this numerically, recall that we are assuming the following.

- Circumferential currents are negligible.
- The axial current $I(z')$ does not vary circumferentially. (This is *not* the same as the first assumption!)
- As for the electrostatic case, we locate the filament on the axis and enforce the boundary condition on the surface, or the reciprocal case.

The reason that we offset the source filament and testing surface (or vice versa) is, as in the electrostatic case, to avoid the singularity present at $z = z'$. Although

approximate, this method works well for thin wires. As for the static case, the kernel is not singular, but for small a can become more nearly so than in the electrostatic case – the R^5 term in the denominator of Eq. (4.29) is largely responsible – and more sophisticated treatments are frequently used. The problem usually occurs with the "self" term (the element of $[Z]$ with $m = n$). The usual remedy is to subtract a term with the same order of singularity but which can be integrated analytically, and then to integrate numerically the difference between the singular term and the remainder, since this is usually quite well behaved. Examples of this type of treatment of singular integrals will be discussed in Chapter 7 (although in a slightly different context).

Approximating the current

The same idea is used for approximation of the current as we used for charge, namely some sort of discrete approximation using a set of functions of known shape but unknown amplitude. The most widely used basis functions are pulse (piecewise constant, as used for the electrostatic problem), triangular (piecewise linear) and piecewise sinusoidal. An especially convenient form arises when piecewise sinusoidal basis functions are chosen. In this case, for a wire ℓ in length, lying on the z-axis from $-\ell/2$ to $\ell/2$, the nodes are defined as

$$z_n = -\ell/2 + (n-1)\,\Delta, \qquad n = 1, 2, \ldots, N+1 \tag{4.30}$$

$$\Delta = \frac{\ell}{N} \tag{4.31}$$

The basis function on the nth segment is:

$$h_n(z) = \begin{cases} \frac{I_n \sin k(z_{n+1}-z) + I_{n+1} \sin k(z-z_n)}{\sin k\Delta z_n} & \forall |z - z_n| \le \Delta \\ 0 & \text{otherwise} \end{cases} \tag{4.32}$$

It actually consists of two parts, with two associated (and unknown) coefficients I_n and I_{n+1}. It is often convenient to reinterpret the function as spanning two segments, with one associated coefficient I_n. With this interpretation, it may be shown [8] that the $\hat{z}$-directed scattered field from the nth basis function is given by:

$$\vec{E}_z^{\text{scat}} = -j30 \left[\frac{e^{-jkR_{n-1}}}{R_{n-1} \sin k(z_n - z_{n-1})} - \frac{e^{-jkR_n} \sin k(z_{n+1} - z_{n-1})}{R_n \sin k(z_n - z_{n-1}) \sin k(z_{n+1} - z_n)} + \frac{e^{-jkR_{n+1}}}{R_{n+1} \sin k(z_{n+1} - z_n)} \right] \tag{4.33}$$

[8] A detailed derivation of this was given in the first edition of Stutzman and Thiele's antenna text [8, p. 330], but was removed from the second edition [7].

The lengths R_{n-1}, R_n, and R_{n+1} are respectively the distances from nodes $n-1$, n and $n+1$ to the field point.

With this particular choice of basis function, the integrals can be carried out *analytically*, and this has been quite widely used in MoM codes. Note that at the ends of the wire, the terms I_0 and I_{N+1} are ignored, essentially forcing them to zero (which is the expected behavior of the current).

As for the electrostatic case, a linear system is assembled using the results for the field scattered by each segment. The simplest "testing" scheme is again collocation: this is most conveniently done at the *nodes* in the case of sinusoidal basis functions, which are in the center of the basis functions as defined above.

The incident field

It is important to realize that an MoM problem requires some form of excitation (in the same way as an FDTD model, for instance); commercial codes are no exception. A key difference between the electrostatic and electrodynamic cases is the concept of the incident field, as already outlined, which provides this excitation. For an incident plane wave, peak value E_0 V/m normally incident on the z-directed dipole (along the x-axis in this case), the expression is:

$$E_z^{\text{inc}} = E_0\, e^{-jkx} \tag{4.34}$$

as already discussed.

For an antenna problem, a very simple form of feed is the "delta-gap"; in this case

$$E_z^{\text{inc}} = \pm V/\delta \tag{4.35}$$

for an impressed voltage of V at the terminals of the antenna and gap length δ (quite often, the length of the segment). This source is also sometimes placed at the node between segments. The sign depends on the convention adopted regarding voltage. For the basis functions discussed, the direction of positive current flow is from node n to $n+1$.

More realistic models are available, such as the "frill" source. This models a coaxial line, whose center conductor becomes a monopole and whose outer conductor opens into an infinite ground plane. In this case, the electric field on the axis of the $\hat{z}$-directed monopole is (again, similar comments pertain regarding the sign):

$$E_z^{\text{inc}} = \pm\frac{V}{2\ln(b/a)}\left(\frac{e^{-jkR_1}}{R_1} - \frac{e^{-jkR_2}}{R_2}\right) \tag{4.36}$$

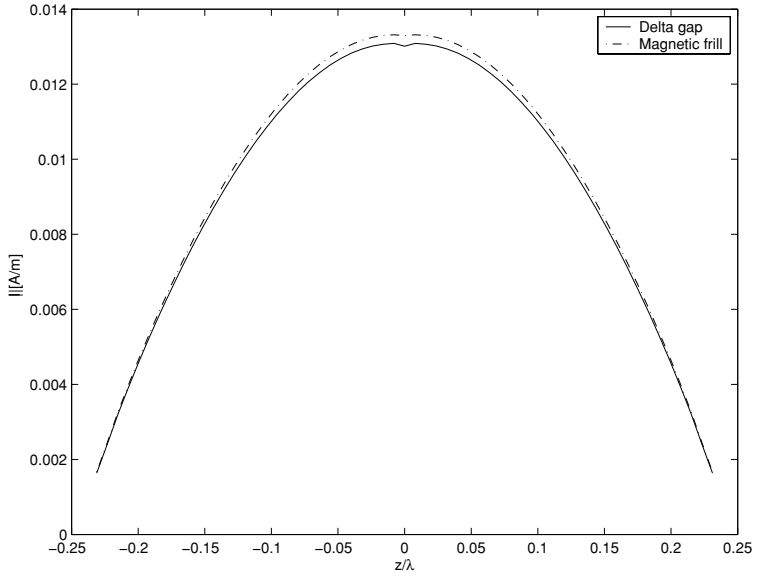


Figure 4.3 Current on a resonant dipole computed with the MoM using piecewise sinsuoidal basis functions and collocation. $L = 0.47\lambda$, $a = 0.005\lambda$, with $N = 60$ segments.

with

$$R_1 = \sqrt{z^2 + a^2}$$
$$R_2 = \sqrt{z^2 + b^2} \tag{4.37}$$

where a and b are the inner and outer radii of the coaxial feedline. V is the terminal voltage. Usually, this is used an an equivalent model, in which case a is the radius of the wire and b is then chosen as some reasonable value – often the equivalent characteristic feedline impedance $Z_0 = 60\ln(b/a)$ is chosen as $50\,\Omega$, i.e. $b \approx 2.3a$. It is worth commenting that the current (and hence antenna terminal impedance) is very little affected by this value.

Some computed results

An example for the current distribution on a thin resonant dipole ($L = 0.47\lambda$, $a = 0.005\lambda$) computed using the MoM is shown in Fig. 4.3. This MoM code, implementing the theory in this chapter in MATLAB, uses piecewise sinsuoidal basis functions and collocation. Results are shown for both the delta-gap and magnetic frill sources, using $N = 60$ segments. The impedance computed with the former was $Z_L = 76.7 + j4.7\,\Omega$ and for the latter, $Z_L = 74.8 + j8.2\,\Omega$. Considering the relative simplicity of the approximation, this agreement is excellent. An even better comparison is to look at the magnitude of the reflection coefficient Γ; a $75\,\Omega$

system is appropriate here (and was also used for the equivalent coaxial radius in the frill model); the results are −29.7 dB and −25.3 dB respectively. Anyone who has ever tried to measure the reflection coefficients of antennas will be aware that such agreement is more than satisfactory.

However, this computed result appears better than it actually is! What is *not* shown on Fig. 4.3 is that the magnetic frill source converges very slowly; 60 segments corresponds to a sampling density of around 120 per wavelength, approximately an order of magnitude times the usual rule-of-thumb for full-wave MoM codes. Using $N = 6$, the delta-gap model produces $Z_L = 62.1 - j67.8\ \Omega$; the real part is moderately accurate although the reactive part is not; however, the magnetic frill prediction, $Z_L = 13.8 - j15.1\ \Omega$, is unconverged and entirely misleading. In Chapter 5, we discuss checking convergence of computed data in some detail. Commercial codes use somewhat more sophisticated treatments than those discussed here to obtain more rapid convergence.

4.3.2 A caveat regarding thin-wire formulations

An important point to note with thin-wire formulations is that they admit no exact solution, and exhibit a phenomenon known as relative convergence: as the number of unknowns in an MoM solution is increased, the solution converges initially to a value close to the exact solution (what has been called the region of rapid initial convergence), then enters a stable region, and finally diverges in a region of instability. For wires which are too thick for effective use of the thin-wire approximation, there is *no* stable region at all. This was considered in detail by Collin [9] (reprinted in [6]) and is also discussed in his textbook [10].

4.4 More on basis functions

Suitable basis functions were the topic of research for many years, and in this section, some details are provided of two other solutions which have been widely adopted. Firstly, it is appropriate to provide some background on a public domain code, NEC-2, which for many years was the workhorse of MoM computation. The basis function used by NEC-2 had some particularly elegant features. Following this, some more details are provided on piecewise linear basis functions, which are also very popular.

4.4.1 The numerical electromagnetic code (NEC) – method of moments

It would be inappropriate in a book of this nature not to include some discussion of NEC, or NEC-2 in the case of the public domain version. This code has a long

lineage, with its genesis in a code called BRACT (released in 1970), which was developed by contractors MBA Associates, primarily for US Air Force applications. The code that eventually became NEC started as the AMP (Antenna Modelling Program), first released in 1974, again with US military funding. A discussion of the theoretical background and a number of applications for what is clearly this code (although unnamed in the article) may be found in [11], available in the collection [6].

NEC-1 was released in 1977, and NEC-2 in 1981. NEC-2 became, and still is, the most widely used public domain MoM code.[9] NEC-3 was an intermediate version, and saw only limited distribution; NEC-4 was released in 1992 and was the last major release of the code, which is no longer being actively developed further. Until very recently, NEC-4 was still US Military Restricted technology,[10] although all NEC-4 functionality is now available in commercial codes (most prominently FEKO). The various NEC codes were developed at the Lawrence Livermore National Laboratory, one of the major US government research laboratories. Here, we will focus on NEC-2, owing to its ready availability; despite its even more venerable age, it is still a useful tool and quite widely used as a benchmark.

NEC-2 incorporates the Pocklington integral equation formulation for thin wires, as well as a treatment for closed conducting surfaces (the magnetic field integral equation, which will be discussed in Chapter 6). It includes support for a number of features very useful in modelling wire antennas, including: non-radiating networks (e.g. transmission lines); lumped element loading; perfectly or highly conducting wires; incident plane-wave or voltage sources; and treatments of perfect or imperfect grounds. The last included the Sommerfeld formulation for half-spaces; this will be discussed in Chapter 7. It can compute induced currents and charges; near- and far-fields (electric or magnetic); radar cross-section; antenna impedance (and admittance); gain and directivity; and antenna to antenna coupling. It can exploit symmetry of rotation or reflection.

NEC-2 was primarily developed for wire antenna modelling, and many of the problems which have been reported with NEC-2 arose because users tried to use it for modelling surfaces via meshes of wires. Although one can obtain useful answers with careful work with this approach, it is not the purpose for which the code was primarily designed. Provided that NEC-2 is used within its limits, it is still a very useful code.

[9] Whether it was indeed the intention of the US government to make the code public domain is still not entirely clear, but this became the de facto situation by the 1990s.

[10] Since 2003, NEC-4 has been available for a very modest license fee for users in most countries.

4.4.2 NEC basis functions

Much of the success of NEC was due to the basis function used. (In the following discussion, NEC and NEC-2 will be used interchangeably; this theory is also applicable to NEC-4.) A highly desirable requirement of a good basis function is that it should satisfy physical requirements of current and charge continuity. This implies that both the current and its first derivative should be continuous. NEC makes the usual thin-wire approximations, viz. transverse currents are negligible; the circumferential variation of current is negligible; current can be represented by filament on the wire axis; and the boundary conditions on the tangential electric field are only enforced axially, so the basis function is one-dimensional, as in our preceding discussion in this chapter. In developing the basis function, the following interpolation function is first introduced for segment j:

$$I_j = A_j + B_j \sin k(s - s_j) + C_j \cos k(s - s_j), \qquad \forall\, |s - s_j| < \Delta_j/2 \tag{4.38}$$

The parameter s is a local coordinate along the length of the wire, with s_j the value of s at the center of segment j. Δ_j is the length of segment j. This is based on a function originally proposed by Yeh and Mei [12] (reprinted in [6]). Although this is quite often described loosely as the basis function, this is *not* entirely correct. The full basis function is rather more complex. Each NEC basis function spans at least three segments: central, left (minus) and right (plus), supporting interpolation functions of the form of Eq. (4.38) on *each* segment, f_i^0, $f_{i_j}^-$ and $f_{i_j}^+$ respectively. The double subscript is used to identify the jth segment connected to segment i. Figure 4.4 shows the situation for a wire segment with two wire segments connecting to the left and two to the right of the central segment. (In this case, the basis function "centered" on segment i spans all five segments.) For a straight wire, with only one segment on the left and one on the right, one can drop the double subscript and the basis function comprises interpolation functions f_i^0, f_i^- and f_i^+ associated with it (nine unknowns in total) – *each* interpolating as Eq. (4.38). For a

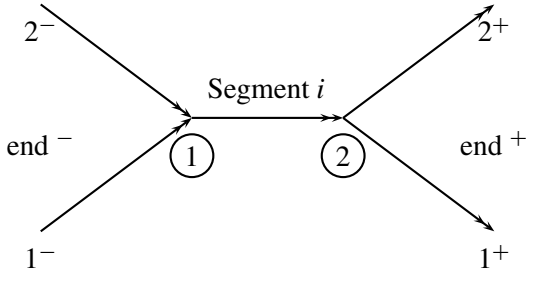


Figure 4.4 Segments covered by the ith basis function.

wire junction as in Fig. 4.4, there are contributions from five segments (and hence 15 unknowns).

The unknowns are now reduced to one per segment by the following constraints:

(1) The current must go to zero at outer edges of connected segments.
(2) The derivative of the current must go to zero at outer edges of connected segments.
(3) The current must be continuous at a segment junction.
(4) At a segment junction, the charge must satisfy a condition known as the *Wu–King* condition; it is continuous for a straight, uniform wire.

These conditions are then enforced on *each individual* basis function – these are sufficient (but not necessary) conditions to ensure current and charge continuity, since the final approximation of current is a linear sum of these basis functions. This was a crucial insight.

For example, these constraints for a segment in a straight wire are as follows.

(1) One from end 1^- and one from end 1^+.
(2) Again, one from end 1^- and one from end 1^+.
(3) Two (one at each end of the central segment).
(4) Four (one each from the segments connected to the $-$ and $+$ ends, two from the central segment itself).

This amounts to ten constraints. From Eq. (4.38), there are three unknowns per interpolation function, and three such functions, making nine unknowns. A charge-related parameter at the segment junctions provides two additional ("invisible") unknowns, producing eleven unknowns per wire segment (more details on this are given below). The ten constraints are then applied to yield one unknown per segment, which is arbitrarily chosen as $-A_i^0$, i.e. the coefficient associated with the constant part of the interpolation function centered on segment i. The details of this process are quite lengthy, and are available in [13].

The advantage of this formulation is that it can be generalized to handle multi-wire connections. Although it appears complex (and indeed the implementation is non-trivial), it is handled entirely within the code and the user is unconcerned with the details.

NEC-2 can also handle junctions involving wires of different radii. The so-called Wu–King condition (an attempt to enforce the continuity of scalar potential, which is the correct quasi-static continuity condition) is applied at each junction:

$$\left.\frac{\partial I(s)}{\partial s}\right|_{\text{at junction}} = \frac{Q}{\ln(2/ka) - \gamma} \tag{4.39}$$

In this expression, $\gamma = 0.5772$, Euler's constant. Q is an unknown related to

charge: it is constant for all wires at a junction and is the "invisible" unknown in the previous discussion.

4.4.3 Piecewise linear basis functions

The NEC-2 basis function is very useful for modelling wire antennas, but is difficult to apply when the structure to be modelled comprises large amounts of conducting surfaces. We will discuss effective methods for modelling surfaces in Chapter 6; at present, all we need to know is that the usual basis function for this is piecewise linear. Hence, such basis functions are very convenient for models including both wires and surfaces. The formulation is very similar to that of Eq. (4.32):

$$h_n(z) = \begin{cases} \frac{I_n(z_{n+1}-z)+I_{n+1}(z-z_n)}{\Delta} & \forall |z - z_n| \leq \Delta \\ 0 & \text{otherwise} \end{cases} \tag{4.40}$$

As with the piecewise sinusoid, the basis function consists of two parts, with two associated (and unknown) coefficients I_n and I_{n+1} , and again, it is often convenient to reinterpret the function as spanning two segments, with one associated coefficient I_n. This idea is very useful at wire junctions.

4.4.4 Junction treatments with piecewise linear basis functions

The NEC junction treatment is sophisticated, but a simpler approach first introduced by Chao and Strait in 1970 is worth mentioning, since it is still quite widely used. The only place the proof appears to have been published is a report for a government research laboratory [14, pp. 22–25] and given that these are frequently rather difficult to obtain, even when unlimited distribution was approved as was the case here, it is worth briefly deriving their approach. A description of the method (without proof) appears in [15, Chapter 4]. Chao and Strait used a slightly more complex variant of the piecewise linear function, with an interior node in each segment to permit better approximation of curved wires; here, we use straight segments.

Consider a three-wire junction at node n, as shown in Fig. 4.5. (The method works for any number of wires, but this keeps things simple. The general case is outlined at the end of the discussion.) Firstly, we introduce a "half-triangle" function of the form

$$h_n(s) = \begin{cases} \frac{I_n(s-s_{n-1})}{\Delta} & \forall s - s_{n-1} \leq \Delta \\ 0 & \text{otherwise} \end{cases} \tag{4.41}$$

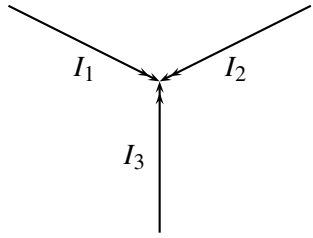


Figure 4.5 A three-wire junction.

This is simply half the basis function defined in Eq. (4.40), but with z replaced by s, a local distance parameter along each wire. In this discussion, it is convenient if $s = 0$ corresponds to the end of each wire away from the junction, with s increasing as one approaches the junction. There are three currents to consider: the current on wire 1, just before the junction; and the same for wires 2 and 3. Note that at these points, the only basis functions contributing to the current are these half-triangle functions. We will call the corresponding coefficients I_1, I_2 and I_3. (We will not include the node n in the notation since it is unnecessary here.) With only two wires, it is sufficient to set $I_1 = -I_2$ and Kirchoff's current law is automatically satisfied. (The negative sign is due to the convention on s adopted above.) With three wires, one possibility is to allow the MoM procedure to include I_1, I_2 and I_3, and then impose the additional constraint

$$I_1 + I_2 + I_3 = 0 \tag{4.42}$$

However, this often results in a constraint equation with very different magnitudes to the usual impedance matrix elements.

The approach suggested by Chao and Strait is to consider each half-triangle coefficient as the sum of two components, hence:

$$\begin{aligned} I_1 &= I_1' + I_1'' \\ I_2 &= I_2' + I_2'' \\ I_3 &= I_3' + I_3'' \end{aligned} \tag{4.43}$$

Further, they propose that

$$\begin{aligned} I_1'' &= -I_3' \\ I_2'' &= -I_1' \\ I_3'' &= -I_2' \end{aligned} \tag{4.44}$$

What this implies is that these basis functions are simply the usual piecewise linear basis functions, spanning both the last segment of the relevant wire and

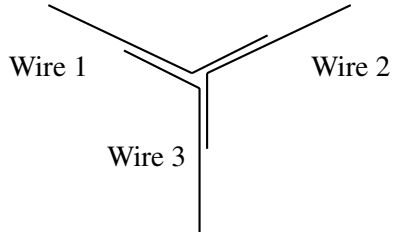


Figure 4.6 The three-wire junction, with the wires overlapped.

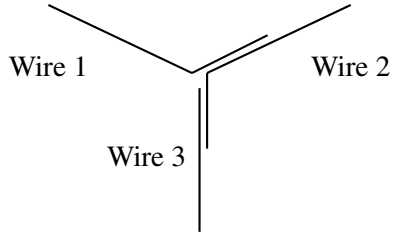


Figure 4.7 The final junction treatment, with two overlapped wires, and one not.

overlapping by one segment onto the next wire: that is wire 1 overlaps onto wire 2, wire 2 onto wire 3, and wire 3 onto wire 1. This is shown in Fig. 4.6. (Our previous comment regarding sign convention applies here too.) Substituting Eq. (4.44) into Eq. (4.43),

$$\begin{aligned} I_1 &= I_1' - I_3' \\ I_2 &= I_2' - I_1' \\ I_3 &= I_3' - I_2' \end{aligned} \tag{4.45}$$

one notes that this choice identically satisfies Eq. (4.42) for any values of I_1', I_2' and I_3'. A unique solution is obtained by arbitrarily choosing one of the degrees of freedom; it is convenient to set $I_3' = 0$. This yields

$$\begin{aligned} I_1 &= I_1' \\ I_2 &= I_2' - I_1' \\ I_3 &= -I_2' \end{aligned} \tag{4.46}$$

A little thought shows that this implies that we overlap wire 1 onto wire 2, wire 2 onto wire 3, but do *not* overlap wire 3 onto wire 1, as in Fig. 4.7. For a general N wire junction, the procedure is to overlap wire n onto wire $n + 1$, but not wire N onto wire 1. Each of these overlapped wires is then treated with the usual

MoM procedure as an open wire, with zero current at the end, as is the one non-overlapped wire.

This is a somewhat cruder approximation than in NEC-2, since it satisfies *only* Kirchoff's current law, and not the continuous scalar potential. However, for junctions involving wires of the same or similar radius it works satisfactorily. It also incorporates an element of arbitrariness, since which wire is not to be overlapped can be chosen at will. Finally, note that this procedure also works with piecewise sinusoidal basis functions.

4.5 The method of weighted residuals

Even at an introductory level, one cannot leave the subject of the method of moments without introducing a very important extension. It was commented that the point-matched procedure which was used only enforced the boundary condition at the sample points. A method generally known in the applied mathematics literature as the *method of weighted residuals* provides a systematic method for improving this. Before we do this, some notation needs to be introduced first. We return to Eq. (4.2), repeated here for convenience:

$$V(z, \rho = a) = \frac{1}{4\pi\epsilon_0} \int_0^\ell \frac{\rho(z')}{R(z, z')} dz' \tag{4.47}$$

and introduce *linear operator* notation

$$\mathcal{L} f = g \tag{4.48}$$

where $\mathcal{L}$ is the operator which maps function f to function g. In the case of Eq. (4.47), for instance, the function f is the charge ρ; the function g is the voltage on the wire; and the linear operator $\mathcal{L}$ is

$$\mathcal{L} = \frac{1}{4\pi\epsilon_0} \int_0^\ell \frac{1}{R(z, z')} (\cdot)\, dz' \tag{4.49}$$

The bracketed dot is used as a place-holder for the function on which this operator acts. Using this notation, the previous development then produces

$$\mathcal{L} \sum_{n=1}^{N} a_n h_n = g \tag{4.50}$$

where, as before, f has been approximated using the basis functions, viz.

$$f \approx \sum_{n=1}^{N} a_n h_n$$

Using point-matching, the $N \times N$ linear system can be obtained by testing the above at N test points. But now, instead of doing this, we form the *residual* as:

$$\mathcal{R} = \mathcal{L} \sum_{n=1}^{N} a_n h_n - g \tag{4.51}$$

This residual is the difference between the approximate solution and the actual solution. (At the risk of belaboring the obvious, if this was one of the very rare problems which can be solved exactly using the MoM procedure, then the residual would be zero.) The point-matching procedure forces this residual to zero at N discrete points. A better approach would be to try to obtain some type of average value of the residual over the domain of the problem (the length of the wire in this case), and set this to zero. One can do this in a quite general fashion by introducing the idea of a *weighting function*, which is multiplied by the residual (and hence the name, method of weighted residuals) and integrated over the domain. The weighting function (also often known as a testing function) is also usually expressed as some type of finite series:

$$w = \sum_{m=1}^{M} w_m \tag{4.52}$$

In this case, the equality is appropriate, since we are not approximating this function. Note also that there are no unknown coefficients. Symbolically, the weighted residual method becomes

$$\int_L \mathcal{R} \sum_{m=1}^{M} w_m dz = \int_L \sum_{m=1}^{M} w_m \mathcal{L} \sum_{m=1}^{N} a_n h_n - \int_L \sum_{n=1}^{M} w_n g = 0 \tag{4.53}$$

Usually, the number of basis functions (N) and the number of weighting functions (M) are equal. Because this integration process frequently defines an *inner product*, an equivalent notation frequently encountered is

$$\langle w_m, \mathcal{L} a_n h_n \rangle = \langle w_m, g \rangle \tag{4.54}$$

This is of course the bracket notation widely used in quantum mechanics, for the matrix algebra formulation of Heisenberg. We will not pursue this further, other than to note that the reason for this analogy is that both classical electromagnetics and quantum mechanics are at heart field theories.

It is easy to show that the method of weighted residuals produces a matrix equation, of the same form as Eq. (4.11), repeated here:

$$\{V\} = [Z]\{I\} \tag{4.55}$$

except that the matrix entries are now

$$\begin{aligned} Z_{mn} &= \langle w_m, \mathcal{L}h_n \rangle \\ V_m &= \langle w_m, g \rangle \\ I_n &= a_n \end{aligned} \tag{4.56}$$

In addition to the question of which type of basis functions to adopt, one now can also choose a variety of weighting functions. This matter has been quite extensively researched. In practice, however, there are two very popular choices. The *Galerkin* procedure uses the same basis and weighting functions. The collocation method, which we have already studied, uses Dirac delta functions, which of course reduce to just testing the operator at the sample points.

Before concluding this section, one or two points which can (and have) caused confusion in the past should be highlighted. Firstly, the inner product implied above for two functions f and g defined on domain $\mathcal{D}$ is:

$$\langle f, g \rangle = \int_{\mathcal{D}} fg \, dV \tag{4.57}$$

For real valued functions, the operation thus defined satisfies the mathematical requirements of an inner product. However, for complex valued functions, it defines a *symmetric* rather than *inner* product, and in this case, the Galerkin procedure requires weighting functions which are the complex conjugate of the basis functions. (The symmetric product defines a quantity known as *reaction* in electromagnetic theory [7, Section 10.7; 4, Section 7.6].) A valid inner product for complex-valued functions is:

$$\langle f, g \rangle = \int_{\mathcal{D}} fg^* \, dV \tag{4.58}$$

where g^* is the complex conjugate of g. In this case, the basis and weighting functions are identical in the Galerkin procedure. Heated debates have arisen over this in the literature; mathematically, it is important, because functions and operators defined within the framework of a proper inner product (and also with some additional properties) are known to be elements of Hilbert and/or Sobolev *spaces*, which confer various properties, important with regard to error analysis and convergence studies, on the problem. In practical engineering applications, the difference is usually unimportant.

On a different topic, the use of Dirac delta functions to derive the collocation approach from the method of weighted residuals has been criticized by some writers [16]. The core of this criticism is the observation that functions such as these are only properly defined in a distributional sense (i.e. under an integral sign). Again, whilst valid from a theoretical viewpoint, in practice the collocation method stands

on its own merits, does not need to be derived thus, and is often a very effective formulation.

One final point we can now explain – the origin of the name "method of moments." Again, consider a one-dimensional problem, such as the electrostatic one we started the chapter with. If we use a method of weighted residuals approach, but select as weighting functions the set $\{z, z^2, z^3, \ldots\}$ we form the *moments* of the residual. In applied mathematics, the method of moments is this specific form of the method of weighted residuals. Harrington chose it as the generic name for method of weighted residuals approaches in electromagnetics, and the name stuck. (In [17], he explained that when first working with the method, he tried to avoid introducing new jargon, and that the name method of moments had previously been used by the Russian mathematicians Kantorovich and Akilov.) Arguably, it may not have been the best choice of name, but four decades of usage in computational electromagnetics have established it so firmly as to be beyond debate. One will also sometimes find the term *boundary element method* used instead of MoM; usually, these terms are identical, although we caution that volumetric MoM formulations are available which are *not* boundary, but rather volume, element methods. (We briefly discuss volume elements in Chapter 6.)

4.6 Further reading

Although elegant theoretically, the MoM is probably the most difficult formulation of those presented in this book to implement accurately and efficiently. In the next chapter, we will turn our attention to the use of commercial codes, and not attempt to develop the simple codes presented in this chapter further. For those intending to develop codes themselves, the MoM is surprisingly badly served by textbooks for applications significantly more advanced than the introductory level treatment presented here, and the following notes may be of use.

Firstly, one still needs to refer to some of the original papers on the topic – there is no MoM equivalent of the books by Silvester and Ferrari or Jin on the FEM [16, 18] or Taflove on the FDTD [19]. In this context, the original paper by Pocklington [20] is both still available in specialized libraries, and still interesting reading, although it will be of little help in developing an MoM code.

An historical aside – H. C. Pocklington

Reading scientific papers from this age can be a little humbling for modern researchers. At the same meeting of the Cambridge Philosophical Society where Pocklington presented his work (25 October 1897), a paper by C. T. R. Wilson on his cloud chamber was presented. At other meetings of that year, numerous

papers appear by J. J. Thomson. 1897 was of course the year that Thomson announced the discovery of the electron at the 30 April, 1897 meeting of the Royal Institution – although he called it a corpuscle at that time. Pocklington was a fellow of St. John's College, Cambridge, and during a sabbatical visit to Cambridge the present author tried to obtain more details about his life. Sadly, no photograph or any other information about him was available, unlike Thomson, who went on to become Master of Trinity College, Cambridge, one of the most prestigious positions at that University, as well as of course winning the Nobel prize. Both Trinity and St. John's have a proud tradition of scientific accomplishment, Trinity numbering Newton and Maxwell amongst its fellows in addition to Thomson, and St. John's Dirac.

The collection of reprints edited by Miller *et al*. is very useful in this context, over a decade after publication [6]. It contains a number of seminal papers, many of which have been referenced in this chapter, as well as a translation from the original German of an important basic theoretical paper by Maue [21], dating back to 1949, which derived what have become known as the electric/magnetic field integral equations, discussed in Chapter 6. The original text by Harrington [22], although reprinted on several occasions and still very widely referenced, is not particularly useful when implementing complex RF simulation codes since its focus is more on basic concepts. However, several important chapters in the now hard to find [23], such as [24], are of considerable interest when implementing complex wire codes, and this still appears to be the only comprehensive derivation available of the magnetic field integral equation as generally used; this work generalized some aspects of Maue's original derivation. Another hard to find reference with useful information on MoM procedures for arbitrarily oriented wire antennas is [25]. In this context, Moore and Pizer's monograph [15] was useful in its time, but unfortunately has never been revised and may be difficult to locate. Finally, another useful source on this topic, which should be far easier to obtain, is the theory manual for NEC-2 [13]. Good introductory treatments of the MoM for antenna applications are available in [4, 7, 26], which provide a somewhat more extended coverage of the subject than in this chapter; however, these are by no means fully comprehensive treatments. The only extended text on the MoM is Wang's [27], and the book has some material which has dated quickly, specifically in the context of a controversy then raging in the literature about iterative methods. Peterson *et al*.'s book [28] has a good theoretical treatment of canonical problems, but as with the introductory MoM treatment in the antenna textbooks mentioned above (and also Wang's volume), it does not deal with the complexities of arbitrarily oriented wire antennas, providing only a brief overview of the topic.

Finally, the question of the convergence of the MoM has proven far from trivial; a brief discussion may be found in Appendix C.

4.7 Conclusions

Although highly simplified, the theory discussed in this first chapter on the MoM is at the core of very complex and powerful MoM programs such as NEC-2 and FEKO. The former uses collocation, with a variant of the sinusoidal basis function as discussed; the latter uses a Galerkin formulation with piecewise linear basis functions, also as discussed. Extensions to arbitrarily oriented wire antennas rapidly become complex, due to the presence of different components of the electric field (set up by the arbitrarily oriented currents) which need to be taken into account. Highly (as opposed to perfectly) conducting metallic structures can also be addressed with very similar theory. NEC-2 was one of the first codes to incorporate a large number of such facilities; modern commercial codes such as FEKO incorporate all these, as well as many other powerful analysis capabilities.

In the next chapter, we will look specifically at the use of FEKO and NEC-2 for wire antenna modelling. Following this, we return to more theoretical topics, considering modelling highly conducting surfaces in Chapter 6, as well as hybrid formulations to reduce the computational cost of this, and we conclude our study of the MoM in Chapters 7 and 8 with a discussion of Green functions, stratified media formulations, and the Sommerfeld potentials. In Chapter 10, we will introduce a very powerful hybrid of the MoM with the finite element method, which permits a very efficient solution of certain classes of problems.

References

[1] R. F. Harrington, *Time-Harmonic Electromagnetic Fields*. New York: McGraw-Hill, 1961.
[2] J. A. Stratton, *Electromagnetic Theory*. New York: McGraw-Hill, 1941.
[3] P. M. Morse and H. Feshbach, *Methods of Theoretical Physics*. New York: McGraw-Hill, 1953.
[4] C. A. Balanis, *Advanced Engineering Electromagnetics*. New York: Wiley, 1989.
[5] J. H. Richmond, "A wire-grid model for scattering by conducting bodies," *IEEE Trans. Antennas Propag.*, **14**, 782–786, November 1966.
[6] E. K. Miller, L. Medgyesi-Mitschang, and E. H. Newman, eds., *Computational Electromagnetics: Frequency Domain Method of Moments*. New York: IEEE Press, 1992.
[7] W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design*. New York: Wiley, 2nd edn., 1998.
[8] W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design*. New York: Wiley, 1st edn., 1981.
[9] R. E. Collin, "Equivalent line current for cylindrical dipole antennas and its asymptotic behavior," *IEEE Trans. Antennas Propag.*, **32**, 200–204, February 1984.

[10] R. E. Collin, *Antennas and Radiowave Propagation*. New York: McGraw-Hill, 1985.
[11] S. Gee, E. K. Miller, A. J. Poggio, E. S. Selden, and G. J. Burke, "Computer techniques for electromagnetic scattering and radiation analyses," in *IEEE Int. Electromagn. Compat. Symp. Rec.*, pp. 122–131, 1971. Reprinted in *Computational Electromagnetics: Frequency-Domain Method of Moments* (E. K. Miller, L. Medgyesi-Mitschang, and E. H. Newman, eds.), pp. 142–152. New York: IEEE Press, 1992.
[12] Y. S. Yeh and K. K. Mei, "Theory of conical equiangular-spiral antennas. Part 1: numerical techniques," *IEEE Trans. Antennas Propag.*, **15**, 634–639, September 1967.
[13] G. J. Burke and A. J. Poggio, "Numerical Electromagnetics Code (NEC) Method of Moments. Part I: Program Description – Theory." Lawrence Livermore National Laboratory, CA, UCID 18834, January 1981.
[14] H. H. Chao and B. J. Strait, "Computer programs for radiation and scattering by arbitrary configurations of bent wires." Syracuse University, Report number AFCRL-70-034, September 1970.
[15] J. Moore and R. Pizer, eds., *Moment Methods in Electromagnetics Techniques and Applications*. Letchworth, Hertfordshire: Research Studies Press, 1986.
[16] P. P. Silvester and R. L. Ferrari, *Finite Elements for Electrical Engineers*. Cambridge: Cambridge University Press, 3rd edn., 1996.
[17] R. F. Harrington, "Origin and development of the method of moments for field computation," in *Computational Electromagnetics: Frequency-Domain Method of Moments* (E. K. Miller, L. Medgyesi-Mitschang, and E. H. Newman, eds.), pp. 43–47. New York: IEEE Press, 1992.
[18] J. Jin, *The Finite Element Method in Electromagnetics*. New York: Wiley, 2nd edn., 2002.
[19] A. Taflove, *Advances in Computational Electrodynamics: the Finite Difference Time Domain Method*. Boston, MA: Artech House, 1998.
[20] H. C. Pocklington, "Electrical oscillations in wires," *Cambridge Philos. Soc. Proc.*, **9**, 324–332, 1897.
[21] A. E. Maue, "Toward formulation of a general diffraction problem via an integral equation," in *Computational Electromagnetics: Frequency-Domain Method of Moments* (E. K. Miller, L. Medgyesi-Mitschang, and E. H. Newman, eds.), pp. 7–14. New York: IEEE Press, 1992.
[22] R. F. Harrington, *Field Computation by Moment Methods*. Malabar, FL: Robert E. Krieger, 1982. Reprint of 1968 edition.
[23] R. Mittra, ed., *Computer Techniques for Electromagnetics*. Oxford: Pergamon, 1973.
[24] A. J. Poggio and E. K. Miller, "Integral equation solutions of three dimensional scattering problems," in *Computer Techniques for Electromagnetics* (R. Mittra, ed.). Oxford: Pergamon, 1973.
[25] W. A. Imbriale, "Applications of the Method of Moments to thin-wire elements and arrays," in *Numerical and Asymptotic Techniques in Electromagnetics* (R. Mittra, ed.). Berlin: Springer-Verlag, 1975.
[26] C. A. Balanis, *Antenna Theory: Analysis and Design*. New York: Wiley, 2nd edn., 1997.
[27] J. J. H. Wang, *Generalized Moment Methods in Electromagnetics*. New York: Wiley, 1991.
[28] A. F. Peterson, S. L. Ray, and R. Mittra, *Computational Methods for Electromagnetics*. Oxford and New York: Oxford University Press and IEEE Press, 1998.

5

The application of the FEKO and NEC-2 codes to thin-wire antenna modelling

5.1 Introductory comments

With the theoretical background now established, one is in a position to start using commercial and public domain MoM programs intelligently. In this chapter, we will discuss primarily the application of the commercial code FEKO for antenna modelling, but will also discuss the use of the public domain code NEC-2[1] in this regard. Other than FEKO, few commercial programs (other than some proprietary NEC-2 extensions) provide good support for modelling thin-wire antennas, the topic of this chapter; such antennas are still very widely used indeed. For commercial programs, material is usually available to assist novice users to get started with the codes.[2] Hence we will not describe the basic concepts of entering the geometry of the problem, including the source, and specifying parameters such as operating frequency and radiation patterns, since these vary from program to program, indeed quite often from release to release, and are usually quite well documented by the suppliers. However, in the case of NEC-2, some comments are in order.

NEC-2 is a "card driven" program, dating back to the days of "decks" of punched cards. A NEC model is described by a geometry file, usually with a `.nec` extension. An example is given in Fig. 5.1. If using NEC in this form, one *must* obtain a copy[3] of the usual manual [1]. Each line in this file describes either a geometrical element or an analysis operation; the first two lines are simply comments; the third line `GW` is a straight wire, with a *tag* of 1 in this case (a tag is a number referring to the particular wire, and is used to simplify later references), divided into 41 segments, with (x, y, z) coordinates of the first end $(0, 0, -0.25)$, of the

[1] Again, as in the previous chapter, we will use NEC-2 and NEC interchangeably in this chapter. All the comments made are equally applicable to NEC-4.
[2] In the case of FEKO, a "Getting Started" manual is provided.
[3] This has been made available on the Internet. See Appendix F for a list of websites which can assist in this regard.

```
CM Dipole Example
CE Start of geometry
GW  1   41  0.000000  0.000000 -0.250000  0.000000  0.000000  0.250000   0.00500
GE  0    0
FR  0   51    0    0 250.00000 2.0000000
EX  0    1   21   00   1.00000   0.00000
XQ  0
EN
```

Figure 5.1 A sample NEC input file.

second end (0, 0, 0.25) and radius 0.005. All dimensions are in meters by default. The fourth line `GE` indicates that the geometry section has ended. The fifth line `FR` specifies the frequency; the sixth, `EX`, specifies a voltage-source excitation on the 21st segment of the wire with tag 1; and the penultimate line, `XQ`, executes the program, computing input impedances and (possibly) radiation patterns. The final line `EN` ends the "deck."

Code tip – using NEC-2

NEC-2 is *only* the computational engine, originally written in one of the earlier versions of FORTRAN, which performs the MoM computations as specified in the input file, and writes data to an output file. *No* graphical support is provided at all. An entire industry grew up providing such support; some packages are fully featured commercial products with major additional computational features, such as SuperNEC; others, such as Wiregrid for Windows, are freeware, providing only graphical user interface (GUI) support. Wiregrid does have one feature worth highlighting; it is able to generate wire-mesh approximations of *surfaces*, using an algorithm published in [2]; no other NEC-2 GUI appears to support this at the time of writing. (Generating such a mesh by hand is an incredibly tedious operation.)

Although not clear from Fig. 5.1, the column spacing can be *crucial* – i.e. the x coordinate of end 1 *must* be entered between columns 11 and 20 for some versions of NEC-2. There are many slightly different versions of the code, compiled by different authors, and the earlier versions had limited parsing ability on data files. Later versions relaxed this, and also permitted the use of commas to demarcate data fields. One is well advised to get one of the many GUI interfaces mentioned above, since otherwise preparing a NEC-2 data file can be very frustrating indeed.

An advantage of the NEC-2 open-source mode of operation is that it lends itself to use in a variety of applications – optimization, for instance – since it is relatively easy to generate NEC-2 input files automatically, and using tools such as `grep`, the output file can be parsed for the required output parameters.

However, this is *not* an operation recommended for beginners. In some cases the code has even been partially or entirely rewritten in other languages – part of the present author's doctoral dissertation was an implementation in a language called Occam, to permit efficient parallelization of the code [3].

FEKO was also influenced by NEC; at the time of writing FEKO still referred to "cards" in the input file. The actual input file used by FEKO has a `.fek` extension, and consists of lines of data, usually preceded by a two-letter label. (It is either in ASCII or binary format; the former is advantageous when generating geometry files on a PC for running on a more powerful computer such as a workstation or even supercomputer.) However, this is a very difficult format for users to comprehend. At the time of writing, FEKO was usually run from a PREFEKO file (with extension `.pre`). PREFEKO is a type of *scripting* language which generates the `.fek` file from elementary geometrical and other primitives.[4]

The code FEKO

This code had its genesis in the doctoral work of Jakobus at the University of Stuttgart in Germany during the early 1990s. It is an acronym of the German name: "**FE**ldberechnung bei **K**örpern beliebiger **O**berfläche", which translates as *field computations involving bodies of arbitrary shape*. It incorporates a powerful MoM treatment using piecewise linear triangular functions for metallic structures – both wires and surfaces. It also supports the MoM treatment of dielectric structures, using either *surface* or *volumetric* treatments. A unique feature of FEKO is the approximate hybrid treatment available using *physical optics*. We will discuss many of these topics in Chapter 6. FEKO is available across a wide range of platforms, including supercomputers. The code ships with a very usable GUI (although this was being redesigned at the time of writing). Recent additions have included the Sommerfeld treatment for stratified media (the topic of Chapters 7 and 8) and the fast multipole method (see Chapter 6). The code is very popular in Europe, and is starting to penetrate the USA and Asian markets at the time of writing. A version with restricted capabilities (sometimes called FEKO Lite) is available at no cost. See Appendix F for contact details.

[4] Many CEM codes have an entirely graphical geometry-enter process; although attractive for first-time users this does not offer the same fine control as the current FEKO approach, but this may well change in future releases of the code. FEKO does of course provide visual feedback of the geometry which has been created as soon as PREFEKO is run. Some, such as MWS, discussed in Chapter 3, offer both.

Historical note – other thin-wire codes

MININEC is another program which one quite frequently sees mentioned. The name is slightly misleading, since it implies that it is a stripped-down version of NEC-2 – this was indeed the original intent of authors Rockway and Logan when the project was first mooted in 1980. However, it evolved into an entirely separate implementation, using a different formulation, and different basis functions (in the current version, triangular ones). See Appendix F for contact details.

Wire (also known as Thin Wire) was a program originally developed by Richmond at Ohio State University; it still has a loyal following there and versions have been made publicly available. See [4, Appendix F] for more details of the code in its 1989 incarnation, WIRE89, with a FORTRAN listing.

5.2 An introductory example: the dipole

No matter what numerical technique has been used – MoM, FDTD, FEM – one of the first things to check is that the solution is indeed *converged*. What we mean by this is that, after a certain point, refining the mesh (making segment size smaller, for a simple MoM problem) does not change the solution. (In Chapter 3, Section 3.2.7, we saw how making Δ smaller improved the quality of the solution by comparison to the analytical result.) To investigate this we will study the half-wavelength dipole. A note is in order here: this term can cause confusion for newcomers in antenna engineering, since what is usually meant is the wavelength at which the dipole exhibits its first resonance – i.e. has no reactive part of the impedance. This is usually equivalent to the wavelength at which the reflection coefficient is minimized in a typical 50 Ω or 75 Ω system, since the real part of the input impedance is generally on the order of 50–70 Ω and changes far less rapidly than the reactance at resonance. It generally occurs at somewhere between $0.46\lambda \sim 0.49\lambda$, depending on the dipole thickness.

Modelling hints – convergence studies using FEKO

In FEKO, there is unfortunately no simple way to undertake a convergence study by creating multiple structures in one file, and one needs to change the discretization manually in the PREFEKO file, run PREFEKO again, and also of course re-run FEKO.[a] There are various ways of proceeding from here, but probably the easiest is to save the output file (`out`) after each run with a distinctive name, and then use the `Import – Select File` option in the FEKO graphical post-processor to read the data in from each file.

[a] It is possible to do this with OptFEKO, the FEKO optimizer, but this is beyond the scope of the present discussion.

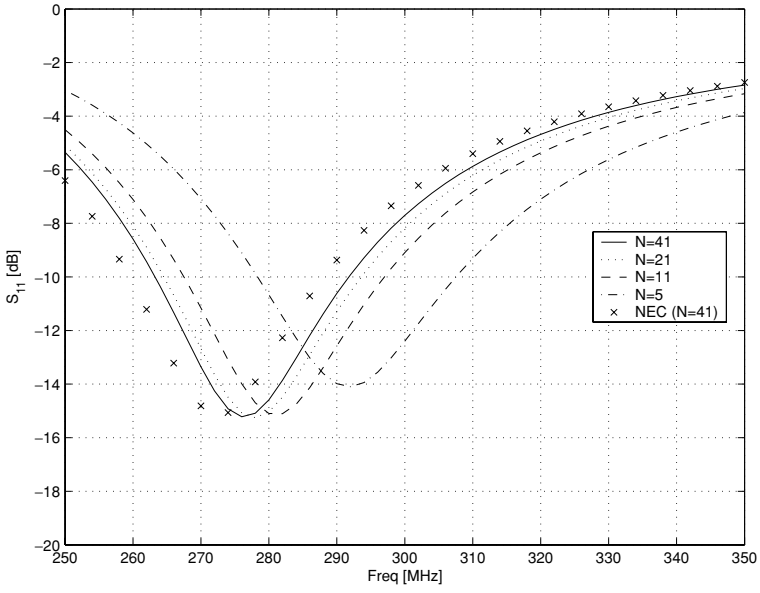


Figure 5.2 Results of convergence study for a dipole of length 0.5 m, radius 0.005 m. `A1` feed model.

The result of such a convergence study is shown in Fig. 5.2. The default reference impedance of 50 Ω was used to create these plots.[5] All produce a minimum reflection coefficient of around −15 dB except for the coarsest mesh (−14 dB); interpolating a little, the frequency of this varies from 292 MHz (5 segments) through 281 MHz (11 segments) and 278 MHz (21 segments) to 276 MHz (41 segments). The five segment model has a segment size of just under $\lambda/10$, which is about the largest segment length which should be used in thin-wire modelling, certainly near a source. FEKO will issue a warning or an error if the segmentation is grossly inadequate. NEC, however, does *not* – many of the preprocessors now available provide this functionality, another reason that it is strongly recommended to use one!

The obvious course is now to proceed with further refining of the mesh (81 segments, etc.) but for subtle theoretical reasons, this is not wise. The problem is that the FEKO solution is based on the *thin-wire* approximation, discussed in Chapter 4. With a large number of segments, each segment becomes very short, and although the wire overall may indeed be thin, this is no longer true for a particular segment. FEKO issues a warning if the ratio of segment length to radius is less than around 3.3, and an error if this is less than 1. (The developers of NEC suggest an even more conservative ratio of around 8 as a preferred lower bound [1, p. 4].) Indeed, our 41 segment model actually violated this, with a ratio of 2.5. If one opens

[5] FEKO offers the ability to *load* sources – this is not the same as setting the reference impedance Z_0.

the output file and views the warnings, one will observe that a warning was indeed issued with the 41 segment model. (FEKO computes the ratio as radius to segment length, so the values reported in the file are the inverse of those in this discussion.)

The difference in resonant frequencies between the 21 and 41 segmentation runs is under 1%. It is important to note that resonant frequencies predicted numerically are often in error, typically by some few percent; indeed, this is perhaps the *least* accurate physical parameter computed by the MoM (and other numerical methods). This is especially true of thin-wire structures, but is generally true of resonant devices. To illustrate this further, we also show a result computed using NEC-2 in Fig. 5.2. NEC-2 predicts a center frequency of around 273 MHz using 41 segments, as opposed to the 276 MHz of the corresponding FEKO computation, an error also on the order of around 1%. NEC-2 uses different basis functions and a collocation approach, whereas FEKO uses piecewise linear basis functions and the Galerkin formulation, so one cannot expect the NEC-2 and FEKO results to be identical. To improve this further, one will need a more sophisticated source model for both codes [6] and one should be aware that this is about the level of accuracy for this parameter which can be expected from standard thin-wire codes.

FEKO offers other methods for driving dipoles, and it is worth looking at them briefly. The A1 model essentially replaces a segment with a region of impressed electric field. It is important to note that this is done *within the code*!

Modelling hints – feed points for wire antennas

Many new users of MoM codes – FEKO, NEC-2 etc. – try to create a dipole from two wires, with a gap in the middle for the feed. *This is incorrect!* The correct approach is to specify a feed on an existing segment. In the region of the feed, the current is of course *displacement* current, rather than *conduction* current; it is effectively the former which the MoM is approximating in the feed region, but it still needs a segment (even though it is fictitious) and its associated expansion function in order to do this.

The other feed models for thin-wire structures offered by FEKO are the A2 and A3 models. The former uses a very thin gap between two nodes. The latter models a coaxial feed; it is derived by considering the TEM fields in a coaxial cable feeding a monopole against a very large ground plane, as discussed in Section 4.3. In Fig. 5.3, the results obtained by applying these three different feed models to

[6] One such approach uses a quasi-static MoM model first to establish the incident field, which is usually *assumed* in such MoM models, and then uses this in the full-wave solution. One also needs to treat end-caps carefully. The best source on this is [5], whose results were also supported by careful measurements.

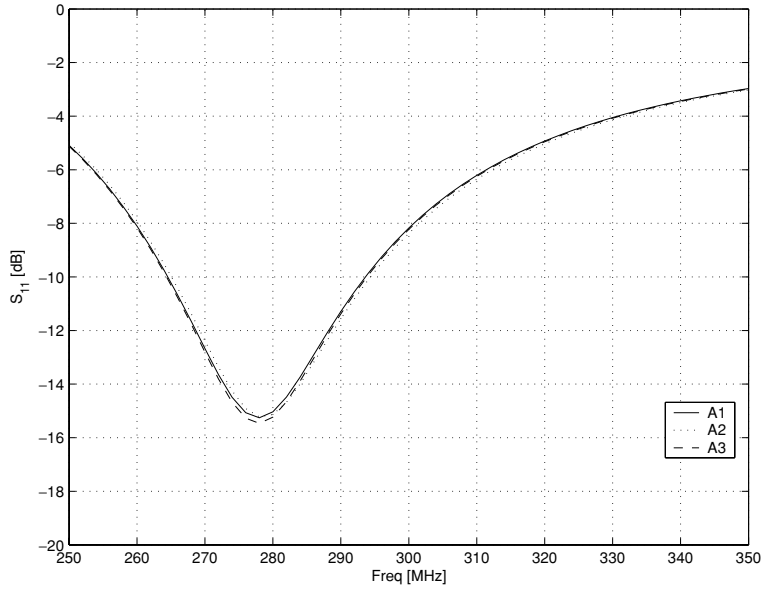


Figure 5.3 Comparison of different sources using 20 or 21 segments: voltage gap on segment (`A1`); voltage gap at node between segments (`A2`); magnetic frill feed (`A3`).

this dipole are shown. Twenty-one segments were used for the `A1` and `A3` sources, and 20 for the `A2` source. (Because the `A2` source models a feed *at a node* rather than on a segment, the model requires an even number of segments for this case in order to place the feed at the dipole center.) For the `A3` source, an equivalent inner and outer radius must be specified; usually, the former is chosen as the wire radius, and hence the latter is 2.3 times this for a 50 Ω system. This was used to produce the results shown in Fig. 5.3. For this example, excellent agreement between the various feed models is observed, which is very gratifying. However, for other problems, one or other model may be far easier to use, hence the provision of different models.

A final comment on convergence testing. For complex models, in particular ones using geometrical data imported from other programs, checking convergence may be very difficult. This example gives some guidelines for the type of errors one should expect. Our coarsest mesh ($\Delta \approx \lambda/10$) produced an error of around 5.5% (with respect to the finest mesh). Refining the mesh to $\Delta \approx \lambda/20$ more than halved the error to around 2%. Refining the mesh again to $\Delta \approx \lambda/40$ once again halved the error. However, the actual values of the errors will vary from problem to problem, and we caution that if it is not possible to use a quite fine mesh (i.e. small segment size of $\Delta \approx \lambda/20$) one needs to be very careful indeed in accepting results generated using *any* MoM program. In the examples to follow, we will

generally use quite fine meshes satisifying at least this criterion, and will not explicitly remark again on convergence, but it should always be kept in mind.

Code tips – structural versus control cards in NEC

NEC differentiates between two different types of cards, namely structural and control cards. The former define actual metallic segments and patches, either via the direct creation of a wire or surface, or via operations on structural elements such as copying or reflection. The latter control parameters such as the location of the excitation, operating frequencies, grounds, near- and far-fields requested etc.

Note that a NEC file requires at least one card which triggers execution, such as a field computation. The `XQ` card is a convenient way of forcing execution otherwise.

FEKO also distinguishes cards in a similar fashion, using the terms geometry and control cards respectively.

5.3 A wire antenna array: the Yagi–Uda antenna

In the preceding section, we discussed how to specify feed models, as well as the importance of checking that the analysis has converged. However, the thin-wire half-wavelength dipole is not a very stimulating engineering design on its own. A much more interesting example is an *array* of dipoles. Two well-known examples here are the Yagi–Uda antenna[7] and the log-periodic antenna, invented at the University of Illinois Urbana-Champaign during the 1950s. Design tables are available for both antennas, and some are reproduced in [6, 7]. The main difference is that the former is a narrowband, moderately high-gain structure, but with only one element (the driven element) fed; the latter is a wideband structure, somewhat lower in gain, with all the elements fed in parallel via a transmission line network. Both are very widely used for VHF and UHF communication, as well as TV reception from terrestrial broadcasts. (Satellite transmissions are in the microwave band and a high-gain dish is generally used.) As an example, we will analyze a simple Yagi–Uda array, with one *reflector*, one *driven element* and four *director* elements. This is illustrated in Fig. 5.4.

We use the design data of Viezbicke, available in [6, Section 5.4] or in [7, Section 10.3.3]. Viezbicke's design process usually consists of two stages: firstly, establish the director and reflector lengths for the prototype Yagi [6, Table 5.4];

[7] S. Uda is credited with the original design in 1926; the first English language publication was by his professor, H. Yagi, in 1927 [6, p. 188].

Table 5.1 *Design data for a six-element Yagi array, wire radius* $a = 0.00425\lambda$*, using Viezbicke's results*

Element	Length (in wavelengths)	Spacing (in wavelengths)
Reflector	0.482	−0.2
Driven	0.475	N/A
D_1	0.428	0.25
D_2	0.420	0.25
D_3	0.420	0.25
D_4	0.428	0.25

Spacing is relative to the previous element.

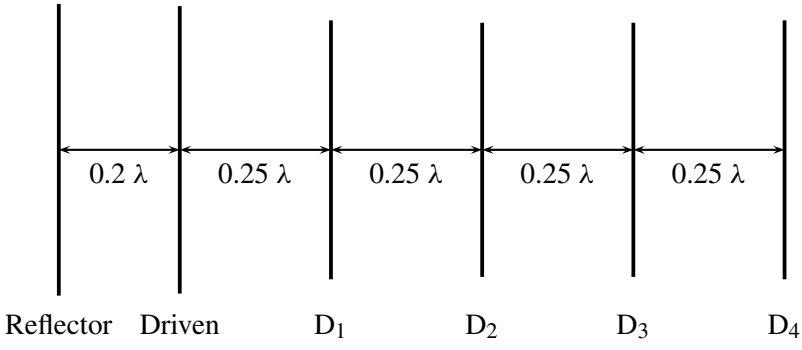


Figure 5.4 The six-element Yagi array described in the text.

secondly, compensate for the actual wire radius using [6, Fig. 5–37]. By using the wire diameter $d = 2a = 0.0085\lambda$ of the prototype given in [6, Table 5.4], no compensation is required. These tables do not give the length of the driven element; this is usually the resonant dipole length in free space [6, p. 190]. (This can be established from standard results, for instance [6, Table 5.2]: for $L \approx 0.5\lambda$, $L/2a \approx 59$, the required shortening is about 5%, i.e. 0.475 λ.) Hence our design is as summarized in Table 5.1. Director 1 is closest to the driven element. Extracts from the FEKO `.pre` file are given in Fig. 5.5; a NEC-2 data file is shown in Fig. 5.6.

Results for the reflection coefficient and the H-plane pattern at 291 MHz (the actual resonant frequency) are given in Figs. 5.7 and 5.8 respectively. The simulation indicates around a 5% −10 dB impedance bandwidth (the range of frequencies for which $|S_{11}|$ is less than −10 dB, corresponding to VSWR ≤ 2), which is as expected for a thin-wire structure. (These results were obtained for a segment length of around $\lambda_0/40$ at the center frequency.) The resonant frequency is 291 MHz, some 3% lower than the design frequency. Since quite fine segmentation has been

```
.
.
#freq_o = 300.0e6           ** centre frequency in Hertz
#lam_o = #c0/#freq_o   ** wavelength in metre,  #c0 = speed of light in vacuum

#rf_len = 0.482*#lam_o   ** Reflector
#dr_len = 0.475*#lam_o  ** driven element
#d1_len = 0.428*#lam_o
#d2_len = 0.420*#lam_o
#d3_len = 0.420*#lam_o
#d4_len = 0.428*#lam_o
#S_R = 0.2*#lam_o
#S_D = 0.25*#lam_o
#diam = 0.0085*#lam_o
#num_seg=21
#delta=#dr_len/#num_seg


** Parameters for segmentation
IP                            #diam/2             #delta
** Geometry of radiating structure
DP   rf_n                     -#S_R     0         -#rf_len/2
DP   rf_p                     -#S_R     0         #rf_len/ 2
BL   rf_n rf_p
DP   dr_n                     0         0         -#dr_len/2
DP   dr_p                     0         0         #dr_len/ 2
BL   dr_n dr_p
DP   d1_n                     1*#S_D    0         -#d1_len/2
DP   d1_p                     1*#S_D    0         #d1_len/2
BL   d1_n d1_p
DP   d2_n                     2*#S_D    0         -#d2_len/2
DP   d2_p                     2*#S_D    0         #d2_len/2
BL   d2_n d2_p
DP   d3_n                     3*#S_D    0         -#d3_len/2
DP   d3_p                     3*#S_D    0         #d3_len/2
BL   d3_n d3_p
DP   d4_n                     4*#S_D    0         -#d4_len/2
DP   d4_p                     4*#S_D    0         #d4_len/2
BL   d4_n d4_p
** End of geometric input
EG   1    0    0    0    0
.
.
.
```

Figure 5.5 Part of a PREFEKO file for the six-element Yagi array illustrating the use of user-defined variables and scaling.

```
CM 6 element Yagi
CE Start of geometry
GW1,21,-0.200000,0.000000,-0.241000,-0.200000,0.000000,0.241000,0.00425
GW2,21,0.000000,0.000000,-0.237500,0.000000,0.000000,0.237500,0.00425
GW3,19,0.250000,0.000000,-0.214000,0.250000,0.000000,0.214000,0.00425
GW4,19,0.500000,0.000000,-0.210000,0.500000,0.000000,0.210000,0.00425
GW5,19,0.750000,0.000000,-0.210000,0.750000,0.000000,0.210000,0.00425
GW6,19,1.000000,0.000000,-0.214000,1.000000,0.000000,0.214000,0.00425
GE  0    0
FR  0   51    0    0 275.00000 1.0000000
EX  0    2   11   00   1.00000   0.00000
XQ  0
EN
```

Figure 5.6 A NEC-2 file for the six-element Yagi array. This file uses the comma-delimited format.

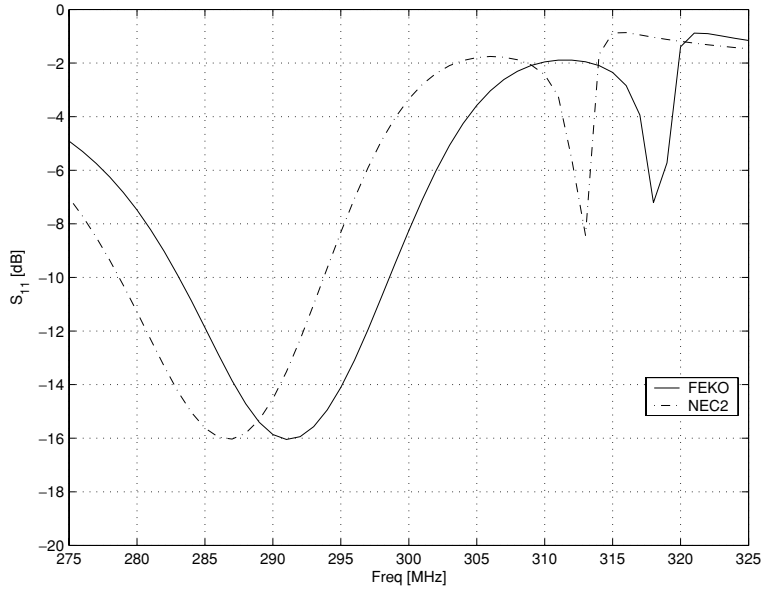


Figure 5.7 Reflection coefficient of the six-element Yagi array.

used, this is probably a real effect, and were one to build this antenna, all the dimensions should be scaled by a factor of 0.97 to obtain a resonant frequency of 300 MHz. The peak directivity is just over 11 dBi (i.e. referred to an isotropic radiator). Viezbicke's tables indicated a gain of 10.2 dBd (referred to a half-wave

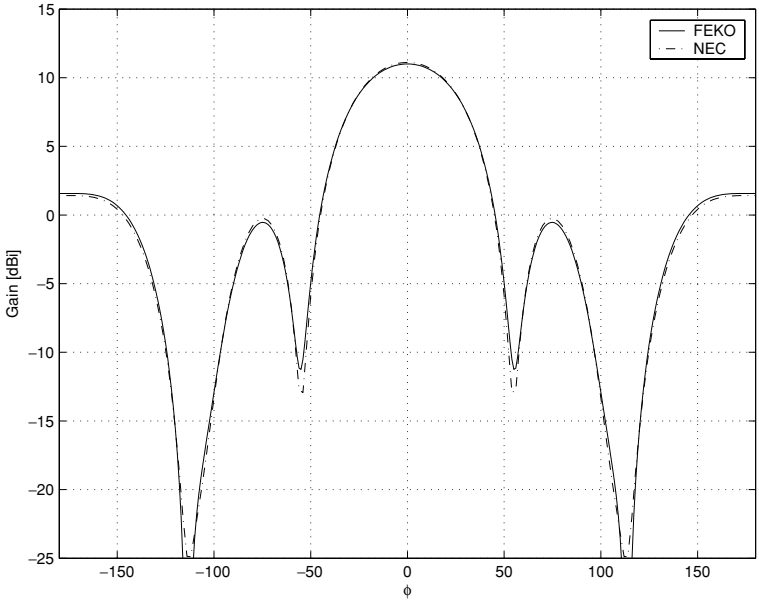


Figure 5.8 H-plane pattern of the six-element Yagi array at its resonant frequency.

dipole), which is equivalent to 12.35 dBi. The reason for the difference is that the directivity quoted here has been computed at the resonant frequency, whereas the peak gain is achieved at around 305 MHz, and is indeed about 12.3 dBi. From Fig. 5.8, the front-to-back ratio (the difference between the radiation in the forward and rear directions) is around 10 dB; Viezbicke's tables indicated around 19 dB, but again, the comparison is at a different frequency. Note that gain and directivity are not synonymous in antenna engineering, but since our antenna is lossless, we can use the terms interchangably here.

Also shown on Fig. 5.7 are the results of a NEC-2 simulation, run with a similar segmentation. The NEC-2 data file is shown in Fig. 5.6. The NEC-2 results show a yet lower resonant frequency of about 287 MHz, some 1.4% lower than the FEKO results. As we commented in the previous example, this is about as accurate a result as one can expect with two different MoM codes using relatively basic feed models. Interestingly, both simulations show another very narrow quasi-resonance just above the design frequency.

Code tip – using Wiregrid for Windows

This very useful NEC-2 preprocessor and postprocessor is available free, and is very largely self-explanatory, with on-line documentation, but here are a few useful tips which can otherwise cause frustration.

- The program has a function which permits one to see the actual wire radius visually. However, this deactivates most of the editing functions and needs to be switched off before proceeding further.
- There is an extremely useful function which forces an odd number of segments on all wires.
- Finally, after graphing etc., the NEC output file must be `Released` before one can edit the model again.

Figure 5.8 also shows the NEC radiation pattern predictions (the NEC results are computed at 287 MHz, the resonant frequency computed by NEC); we use these to illustrate an important point, namely the far-field radiation patterns are not as sensitive a parameter as the input impedance, and hence excellent agreement with other codes can usually be expected. (Agreement with measurements tends to be less satisfactory; frequently, the problem lies with the experimental setup, for instance problems with the feed cables interfering with the patterns.)

We did not explicitly perform a convergence check, since we are using a fine discretization with around 40 segments per wavelength, but of course the comments in our introductory dipole section apply. Due to the relatively thick dipoles in use, one cannot refine the mesh further without starting to violate the thin-wire assumptions.

Aside from the lower center frequency – which as we commented above, is easily fixed in practice (or indeed in simulation) by scaling – our six-element Yagi array works moderately satisfactorily. Now, we are in a position to evaluate quickly the effect of having to use a different wire radius etc., as is quite probable in an actual design. This however might degrade the performance of the antenna. We might also not be satisfied with the front-to-back ratio, for instance, and wish to improve this. This leads into the field of optimization, which FEKO supports, although we will not pursue this further here.

Modelling hints – using user-defined variables and scaling

When developing a general-purpose model, it is often useful to specify dimensions in terms of λ_0, which makes it very easy to change the operating frequency. Also, all the dimensions are given in terms of user-defined variables, so that if we want to change the design of the antenna (perhaps by optimization), we have already done a lot of the work. An example of this is shown in Fig. 5.5, which shows part of the PREFEKO file exploiting user-defined variables. Some other commercial codes, such as MWS, have similar abilities. Connected to this is scaling: a popular use of this is to permit microwave structures to be entered in millimeters. Whilst NEC-2 does support scaling, it does not support user-defined variables.

Modelling hints – wire radius versus diameter

Here is an important point to note, which even experienced users forget from time to time: wire thicknesses in FEKO and NEC-2 are specified in terms of *radius*, whereas especially older texts in antenna design often use *diameter*. Accidentally confusing these is a common source of error; to make things worse, the simulation will often still appear to work, but the results produced are usually subtly incorrect.

5.4 A log-periodic antenna

The Yagi–Uda example highlighted a number of points, but in a sense was simply an extension of the dipole problem, since the additional wires – the reflector and the directors – were passive, and it was just a case of adding these into the `.pre` file. The problem we will now investigate, however, brings some new points, with regard to both FEKO modelling and antenna engineering. It also serves as an introduction to some ideas in wideband antennas.

The log-periodic (log-p) antenna consists of a number of wire dipoles, but unlike the Yagi–Uda antenna, they are *all* fed (by means of a transmission line, which provides a parallel feed). Also, each element is smaller than and more closely spaced to its predecessor; the *ratio* is constant, and τ is the design parameter which specifies this. With dipole lengths L_n and spacing d_n, this is defined as:

$$\tau = \frac{L_{n+1}}{L_n} = \frac{d_{n+1}}{d_n} \tag{5.1}$$

The other parameter which defines a log-periodic array is the *spacing factor* σ, defined as

$$\sigma = \frac{d_n}{2L_n} \tag{5.2}$$

One can also compute α, the angle of the wedge bounding the dipole arms of the log-p, from these parameters:

$$\alpha = 2 \arctan\left(\frac{1-\tau}{4\sigma}\right) \tag{5.3}$$

A value of τ close to 1 indicates a log-p with a very slow expansion, i.e. long overall length, but also higher gain. The design of a log-p is typically a trade-off between length, gain and impedance match. Most design data are based on tables originally published by Carrel in 1961; subsequent research has improved these tables and a typical set are presented in [6, Section 6.7]. We will base our FEKO simulation on [6, Example 6.2].

Table 5.2 *Design data for a nine-element log-periodic array*

Element	Length (in meters)	Spacing to next element (in meters)
1	2.78	0.828
2	2.29	0.682
3	1.88	0.560
4	1.54	0.459
5	1.27	0.378
6	1.04	0.310
7	0.858	0.256
8	0.705	0.210
9	0.579	—

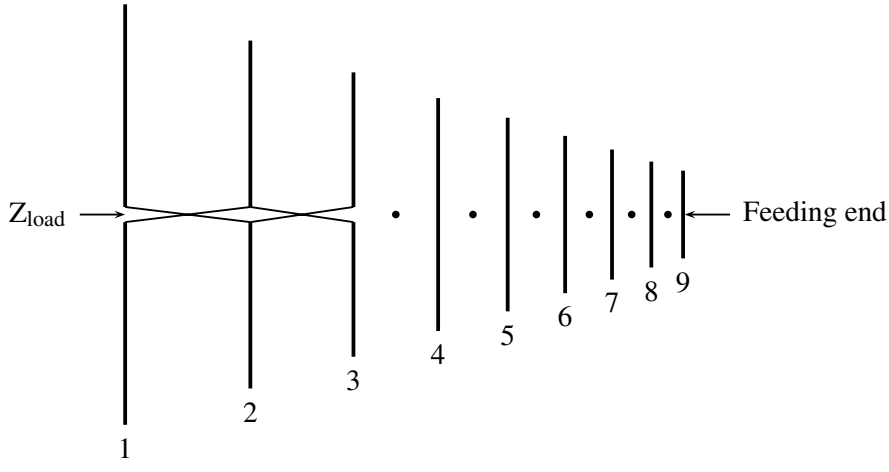


Figure 5.9 The nine-element log-periodic array described in the text. The details of the crossed feed are only shown for the largest three elements, but repeat to the end of the array. Also shown is the feeding end, as well as the position for a possible terminating load, as discussed in the text.

To summarize this briefly for readers without ready access to this reference, the design specification is for a 6.5 dB gain antenna over the VHF-TV and FM broadcast bands, which span the frequency range 54–216 MHz (a 4:1 bandwidth). From the design tables, $\tau = 0.822$ and $\sigma = 0.149$ are selected to satisfy the gain requirement. The lowest frequency determines the length of the longest element, usually chosen as $\lambda_{max}/2$, or 2.78 m in this case. Elements are then placed until an element shorter than $\lambda_{min}/2$ is produced. In this case, nine elements are required. The tabulated data are for a dipole radius 1/250 of the dipole length, clearly varying from element to element. The characteristic impedance of the transmission line is 100 Ω. The design is summarized in Table 5.2 and illustrated in Fig. 5.9.

To implement this in FEKO, there are several approaches that can be taken. The first is simply to create nine wires. A better approach is to use the `!!FOR ... !!NEXT` loop structure, as illustrated in Fig. 5.10. It will also be noted that we construct the elements from four points: two at each end, but also two very close to the center. We do the latter for two reasons. Firstly, there is then always a segment at the center of the element to feed, no matter what the segment length. Secondly, we use the label (`LA`) card (the equivalent of a tag in NEC) to attach a unique label to these central segments; this makes connecting these fed segments (which represent the terminals of the elements) via a transmission line much easier. This is the next step to consider.

Modelling hints – using iteration loops and conditional execution structures in PREFEKO

Many antennas consist of repeated components, and PREFEKO has a very useful feature to implement this, namely the `!!FOR ... !!NEXT` loop (iteration) structure. This is illustrated in Fig. 5.10. We have used another useful feature as well, namely the `!!IF ... !!THEN ... !!ELSE` conditional. Note that d_n is computed from the *current* length, and is computed before we update (reduce) the length for the next execution of the loop.

NEC-2 has no such functionality – the closest NEC-2 gets is the coordinate transformation `GM` card, which allows one to copy, translate or rotate parts of the geometry.

We also have to consider how to interconnect the radiating elements. The obvious way is to connect wires to the elements to form a transmission line explicitly. However, this is not a very efficient way of handling the problem. Transmission lines are non-radiating structures, and can be succinctly described using two-port circuit theory. FEKO incorporates this feature, implemented using the `TL` card. (This functionality is also available within NEC-2, with the same name.) We need eight of these transmission lines; a subtle design point is that the transmission lines are *crossed*, i.e. reverse phase, from element to element; this is done to compress the overall length of the antenna. (In NEC, such crossed lines are specified by using a negative characteristic impedance.) These are also implemented using a loop. Finally, the transmission lines of log-periodic antennas are often terminated with a resistive load (usually equal to the transmission line characteristic impedance, 100 Ω in this case) to improve the impedance match. This is done here via the special handling of the last transmission line, which adds a shunt (parallel) admittance of $1/100$ S to the feed segment of the last antenna.

```
** Analysis of a 9-element logarithmic periodic antenna.

** Some definitions for the geometry

#sigma = 0.149      ** scaling factor for spacing [eqn.6.83,S&T]

#tau = 0.822        ** scaling factor for elements [eqn.6.85,S&T]

#len = 2.78         ** length of element (initially L_1)

#rad = #len/250.0   ** radius of first element: L/2a = 125

#Zline = 100        ** transmission line impedance

#Zload = 100        ** load impedance at the last element (set to very large value
                       if not present)

#num =9             ** number of elements


** Frequency specification and segmentation

#freq_min = 50.0e6             ** start frequency

#freq_max = 250.0e6            ** stop frequency

#lambda_min = #c0/#freq_max     ** minimum


#seglen = #lambda_min / 20

IP                                                  #seglen

** Initial values for the loop

!!FOR #i = 1 to #num

!!IF (#i = 1) THEN

**   This is the first element to be created, at origin

#x = 0

!!ELSE

**   Other elements spaced logarithmically

#x = #x+#d

!!ENDIF

**   Create the wire with the correct radius, use a unique

**   label #i for the centre segment

#z = #seglen   ** ensure that just one segment at the centre

DP   P1                       #x        0         -#len/2.0

DP   P2                       #x        0         -#z/2.0

DP   P3                       #x        0         #z/2.0

DP   P4                       #x        0         #len/2.0

LA   0
```

Figure 5.10 PREFEKO file for the nine-element log-periodic array.

```
BL   P1   P2                  #rad

LA   #i

BL   P2   P3                  #rad

LA   0

BL   P3   P4                  #rad

** Compute inter-element spacing to next element. Note that d_n is the spacing
   between elements

** L_n and L_n+1 and must be computed using current length.

#d = 2.0*#sigma*#len

**   Now apply scaling for next element (shorter)

#len = #len*#tau

#rad = #rad*#tau

!!NEXT

** End of the geometry

EG   1    0    0    0    0


** Create all the transmission lines (again a loop is very useful)

!!FOR #i = 1 to #num-1

**   Extra shunt admittance at the first element

!!IF #i=1 THEN

#YS = 1 / #Zload

!!ELSE

#YS = 0

!!ENDIF

**   Define the transmission line from label #i to label #i+1 (crossed)

TL   1    #i   #i+1 1         -1        #Zline              #YS

!!NEXT

** Excitation by a voltage source at the last (shortest) element

FR        2                  #freq_min           #freq_max

A1   0    #num               1         0

** Vertical radiation pattern - gain

FF   1    1    1    1         90        0

** Vertical radiation pattern - directivity

FF   1    1    1    0         90        0

EN
```

Figure 5.10 (*Continued*)

```
CM 9 element log-p
CE Start of geometry
GW1,47,0.000000,0.000000,-1.390000,0.000000,0.000000,1.390000,0.01110
GW2,39,0.828400,0.000000,-1.142600,0.828400,0.000000,1.142600,0.00910
GW3,33,1.509400,0.000000,-0.939200,1.509400,0.000000,0.939200,0.00750
GW4,27,2.069200,0.000000,-0.772000,2.069200,0.000000,0.772000,0.00620
GW5,23,2.529300,0.000000,-0.634600,2.529300,0.000000,0.634600,0.00510
GW6,19,2.907500,0.000000,-0.521600,2.907500,0.000000,0.521600,0.00420
GW7,15,3.218400,0.000000,-0.428800,3.218400,0.000000,0.428800,0.00340
GW8,13,3.474000,0.000000,-0.352500,3.474000,0.000000,0.352500,0.00280
GW9,11,3.684100,0.000000,-0.289700,3.684100,0.000000,0.289700,0.00230
GE  0    0
PT -1
PL  3    1    0    1
TL  1   24    2   20 -100.0000   0.00000   0.01000   0.00000   0.00000   0.00000
TL  2   20    3   17 -100.0000   0.00000   0.00000   0.00000   0.00000   0.00000
TL  3   17    4   14 -100.0000   0.00000   0.00000   0.00000   0.00000   0.00000
TL  4   14    5   12 -100.0000   0.00000   0.00000   0.00000   0.00000   0.00000
TL  5   12    6   10 -100.0000   0.00000   0.00000   0.00000   0.00000   0.00000
TL  6   10    7    8 -100.0000   0.00000   0.00000   0.00000   0.00000   0.00000
TL  7    8    8    7 -100.0000   0.00000   0.00000   0.00000   0.00000   0.00000
TL  8    7    9    6 -100.0000   0.00000   0.00000   0.00000   0.00000   0.00000
EX  0    9    6   00   1.00000   0.00000
FR  0  101    0    0  50.00000 2.0000000
RP  0    1    1 1000  90.00000   0.00000   0.00000   0.00000   0.00000         0
EN
```

Figure 5.11 NEC file for the nine-element log-periodic array.

In NEC, the absence of user-defined variables, loops etc. means that we have no option other than to compute the values explicitly and enter them by hand, either into a NEC file directly, or using a preprocessor. An example of a NEC file for this log-periodic array is given in Fig. 5.11.

Code tip – some useful NEC functions

In Fig. 5.11, two cards `PT` and `PL` are used which offer useful functionality. The former is used for selectively or entirely suppressing outputting of the currents,

which is, perhaps unfortunately, the NEC default, since this otherwise inflates output files with data which are rarely used. The latter produces an extra data file (the specific name varies from implementation to implementation) with radiation patterns or currents suitable for later plotting.

Results for the reflection coefficient and the gain of the log-periodic array are given in Figs. 5.12 and 5.13 respectively. Results computed with NEC-2 using a similar segment length are given for some of the parameters, and excellent agreement is noted. Also indicated on Fig. 5.12 is the reflection coefficient level corresponding to a VSWR of 2, widely used as a specification for antenna impedance. (A VSWR ≤ 2 actually corresponds to $|S_{11} \leq -9.54|$ dB, as indicated, but $|S_{11} \leq -10|$ dB is often used instead for convenience.) It will be noted how the use of the terminating resistance improves the impedance match; the antenna has $|S_{11}| < -10$ dB over almost the entire band in this case. Without the terminating resistance, the reflection coefficient varies far more over the frequency band, sometimes lower, but also sometimes unacceptably high. Another point to note is that the log-p array must be fed from the *shorter* end; if fed from the longer end, the long dipoles are excited (but not very effectively) so that there is too little power at the higher frequencies to radiate properly from the shorter dipoles.

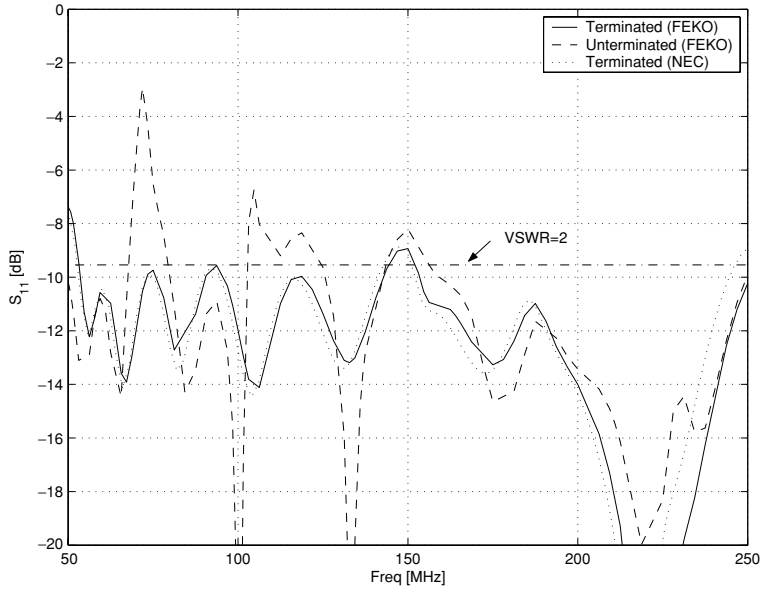


Figure 5.12 Reflection coefficient of the nine-element log-periodic antenna in the text, for both resistively terminated and unterminated cases.

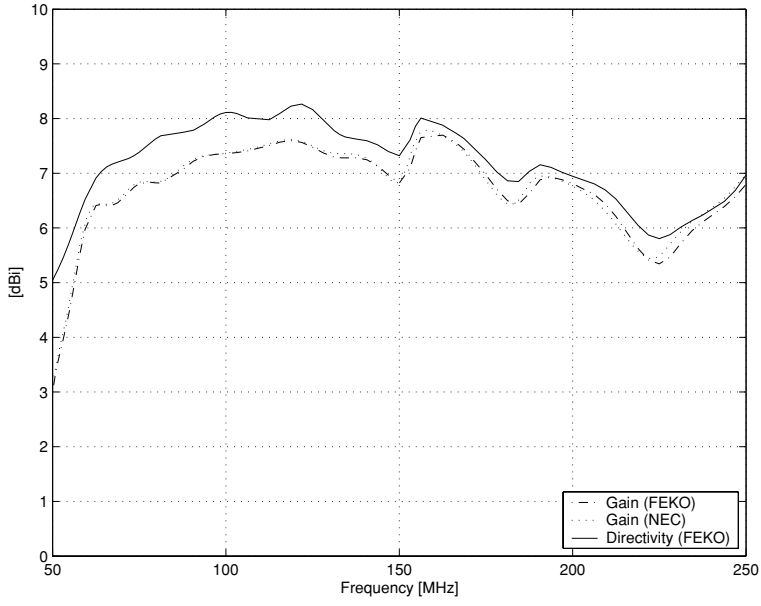


Figure 5.13 A comparison of gain and directivity for the nine-element resistively terminated log-periodic antenna. The gains computed by FEKO and NEC lie essentially on top of one another.

On Fig. 5.13, both *gain* and *directivity* (also sometimes known as directive gain) are given. To revise these terms briefly, the former indicates how well the antenna focusses power spatially, relative to the power *delivered to* it; the latter indicates how well the antenna focusses power spatially, relative to the power *radiated by* it. Clearly, if the antenna has any loss, the two will not be identical, and the difference on Fig. 5.13 is due to the losses in the termination. We have traded off a better impedance match for a slightly poorer gain. (At the very top of the band, we are slightly under the 6.5 dB gain design specification. To improve this, we would have to repeat the design using a longer array, i.e. with more elements, but we will leave this as an exercise.) A final point: because the transmission line has a characteristic impedance of 100 Ω, it is tempting to use this as the impedance level when computing S_{11} etc. However, one should recall that this line is in parallel with the radiating dipole(s), with an impedance of typically $50 \sim 70$ Ω. The net result is that this antenna is quite well matched to a 50 Ω system, which is the FEKO default. Note also that we only compute the gain at one angle, in the direction along the axis of the antenna. A log-p is an end-fire antenna, and radiates in the direction from longest to shortest element.

This example also introduces another feature which FEKO supports, namely the use of *adaptive frequency sampling* (this is not supported by NEC). This example

is sufficiently complex (227 wire segments as discretized) that FEKO takes a noticeable amount of time, typically a second or two, to compute each frequency point. However, the data change rapidly over frequency, requiring a *lot* of points; to obtain good results with uniformly spaced frequency points over the frequency band of interest, one would need at least 100 points, preferably more. FEKO has the ability to determine where to place frequency points in a *non-uniform* fashion, as well as then intelligently interpolating the data by using what is termed a *model based parameter* representation. We use the defaults for this option in the frequency card in the PREFEKO file in Fig. 5.10.

Modelling hints – gains in dB or actual values

Be very careful when plotting gains for these relatively low-gain structures; the gains in dB or in actual value are quite similar numerically, and it is easy to plot the wrong dataset, especially when exporting data!

5.5 An axial mode helix antenna

Helical antennas are another interesting type of antenna. The axial mode helix was invented by Kraus at Ohio State University in 1946, and his textbook on antennas is a mine of information on the subject [8]. Their bandwidth ratio is given theoretically as approximately 1.78 [6, Section 6.2.2]. (A wideband antenna is conventionally defined as one where this ratio exceeds 2, so the helix is close to being "wideband.") Details are also available in [7, Section 10.3.1]. It is also a wire structure, but *unlike* all the previous antennas we have analyzed, which all relied on a *standing wave* on some part of the structure, this is a *traveling wave* antenna, at least in its axial mode of operation, which is the most common mode of employment. The circumference of the antenna is chosen such that currents on opposite sides of the antenna (which would radiate fields out of phase due to the winding of the helix, if the currents were in phase) are delayed by a half-wavelength, so that the resulting radiation is now in phase again along the axis of the helix (and hence the name, axial mode). The radiation is circularly polarized, with the sense of the winding, i.e. a right(left)-hand wound helix generates right(left)-hand polarization. Compared to other candidates, the axial mode helix is quite compact – the helical structure permits a lot of wire to be contained in a moderately small volume – and the design is very popular in the UHF band, especially for satellite communication. (A closely related structure, namely the normal mode helix, is very popular for mobile telephones. It radiates almost isotropically.)

FEKO provides the `HE` card, which greatly simplifies creation of a helix. Indeed, all that is required other than this card is to add a short segment below the helix to

feed it with, and to add a ground plane of some type underneath it. A ground plane of around 0.75λ on each side is usually adequate [6, Section 6.2.2].

To create a ground plane, one can use a mesh of wires – and indeed this was a very widely used method with NEC-2. However, FEKO supports the creation and meshing of surfaces. A simple method of defining a surface is using the parallelogram card (`BP`). This surface is then meshed using triangles.

Modelling hints – connecting wires and plates

Here is something to be careful of. The obvious approach when grounding the helical wire is to generate one surface in the plane $z = 0$, where the feed segment terminates. However, this usually will not work properly! The reason is that FEKO, and indeed any MoM code, needs the *nodes* defining the segments on the wire and the triangular segments on the surface to coincide. Many new users overlook this and it is a frequently encountered fault. In the PREFEKO file, we have generated only a quarter of the ground; this of course includes a point at the origin, where the feed segment connects. We then use geometrical symmetry in two planes ($x = 0$ and $y = 0$) first to create half the ground plane, and then to create the entire ground plane. (The PREFEKO file supplied does this in first the $x = 0$ plane, then the $y = 0$ plane, but the order is actually irrelevant in this example.) A recent addition to FEKO permits users to specify *internal nodes* in polygonal plates, using the `PM` card, which makes it easier to make sure that wires correctly connect to nodes on surfaces.

One final point regarding creating the geometry. FEKO also offers a ground plane (`BO`) card, and this would appear to be very useful. However, one needs to read the "fine print" in this case. This card uses a reflection coefficient *approximation*; i.e. the *fields* radiated by the structure are imaged in the ground plane, but the ground plane is *not* taken into account when the currents are computed by the MoM. As such, it is very useful for antennas some distance above a ground plane, where the currents are indeed hardly changed by the presence of the ground, but entirely *inappropriate* for an antenna fed right against a ground, as the helix is. A careful reading of the user manual cautions that segments should not connect to the ground, but does not describe in detail why this ground plane would be incorrect in this application.

A detailed design example is given in [6, Example 6.2]. The antenna is to operate in the microwave band, with center frequency 8 GHz. The circumference of the antenna, C, is specified as $0.92\lambda = 34.5$ mm. (It will be noted that the scaling card is also used in the PreFEKO file to permit all dimensions to be entered in

millimeters, which is far more convenient than meters in this frequency range.) The pitch angle α is chosen as 13° (a value based on prior design experience). The spacing between turns, S, works out at 7.96 mm, and the antenna has $N = 10$ turns. With the 1.78 bandwidth ratio and center frequency of 8 GHz, the lower and upper frequencies are 5.75 GHz and 10.25 GHz respectively. The PREFEKO file is given in Fig. 5.14, and Fig. 5.15 shows a FEKO model of the antenna.

Radiation patterns at the lower, center and upper frequencies are shown in Fig 5.16 with a ground plane 1.5λ on a side, somewhat larger than the minimum recommended. The gain at 8 GHz is exactly 13 dBi, somewhat higher than the 10.5 dBi gain predicted by the approximate formula [6, Eq. (6-34)]

$$G \approx 6.2 \left(\frac{C}{\lambda}\right)^2 N \frac{S}{\lambda} \tag{5.4}$$

Commensurate with this increased gain, the half-power (HP) beamwidth of 40° is somewhat smaller than that predicted by the approximate formula [6, Eq. (6–33)]

$$HP \approx \frac{65^\circ}{\frac{C}{\lambda}\sqrt{N\frac{S}{\lambda}}} \tag{5.5}$$

of 48°. It must be emphasized that these are approximate empirical formulas, so some differences are to be expected. Kraus provides another formula [8, Eq. (7), p. 235] for directivity, which he describes as more realistic:

$$D \approx 12 \left(\frac{C}{\lambda}\right)^2 N \frac{S}{\lambda} \tag{5.6}$$

Using this formula yields a gain of around 13.3 dBi, almost exactly as simulated. (Since the antenna is essentially lossless, we are again using gain and directivity interchangeably.)

From Fig. 5.16, the gain at the lower frequency is almost 3 dB less than at the center frequency, and the pattern is starting to show some "squint"; the main beam has moved slightly to the left. At the upper frequency band, the gain has increased and the main beam has narrowed (which may or may not be acceptable, depending on the design requirements).

Impedance results are shown in Fig. 5.17. (These data were generated using adaptive frequency sampling.) It will be noted that the antenna is largely resistive across most of the frequency band. However, towards the lower end of the band, the otherwise smooth impedance curves break down. This type of behavior is *not* predicted by the simple description of operation as a traveling wave antenna [6]. Measured data by Baker [8, Fig. 8–73], who worked on helix arrays with Kraus, indicate almost exactly the same impedance behavior at around 0.7 of the center

```
** A 10-turn helical antenna
**
** Variables
**   Optional scaling factor (set to 0.001 for geometrical data
**   coordinates etc. defined in cm instead of metres)
#scaling = 0.001
**   Frequency and wavelength
#freq = 8e9           ** frequency in Hertz
#freq_min=5.75e9
#freq_max=10.25e9
#lam = #c0/#freq      ** wavelength in metre,  #c0 = speed of light in vacuum
#lam_mm = #lam/#scaling
#circum = 34.5        ** helix cicumference
#h_rad = #circum/(2*#pi)
#h_len = 79.6         ** helix length
#gnd = 1.5*#lam_mm
** Parameters for segmentation
#seg_rad = #lam_mm/100 ** radius of the wire segments
#seg_len = #lam_mm/20  ** maximum length of wire segments
#tri_len = #lam_mm/10  ** maximum size of triangles
IP                          #seg_rad  #tri_len  #seg_len

** Quarter of ground plane
DP   G1                     0.0       0.0       0.0
DP   G2                     #gnd/2    0.0       0.0
DP   G3                     #gnd/2    #gnd/2    0.0
DP   G4                     0.0       #gnd/2    0.0
BP   G1   G2   G3   G4
** Generate rest of ground - imaged first in x=0, then y=0 planes.
SY   1    1    0    0
SY   1    0    1    0
** Helix
DP   ZERO                   0.0       0.0       0.0
DP   A1                     0.0       0.0       2*#seg_len
DP   B1                     0         0.0       #h_len
DP   C1                     #h_rad    0.0       2*#seg_len
HE   A1   B1   C1   0       10
** Wire
LA   1
BL   A1   ZERO
```

Figure 5.14 PREFEKO file for the 10 turn helix.

```
** Apply the scaling factor
SF   1                        #scaling
** End of geometric input
EG   1    0    0    0    0
** Voltage gap excitation at segment just above ground
A1   0    1                   1.0

** Note: using adaptive frequency sampling permits only
** ONE of the following analysis options:
** ** Set the frequency card for adaptive frequency sampling.
** FR        2                   #freq_min           #freq_max
**
** ** Trigger execution, no patterns.
** FF   0
**
** Set discrete frequency for radiation patterns.
FR   3    0                   #freq_min           #freq_max
**  Radiation pattern
FF   1    181  1    0         -90       0         1.0
FF   1    181  1    0         -90       90        1.0

** End
EN
```

Figure 5.14 (*Continued*)

frequency, with the reflection coefficient suddenly increasing dramatically from less than -20 dB to -2 dB or worse over a very small frequency change. (Baker's helix was not precisely the same as the one simulated here, hence the frequency at which this effect occurs is slightly different.) The reason is that the axial mode ceases effective operation quite abruptly; [8, Fig. 8–34] provides more information on this, in particular via the phase velocity.

In the region near the design frequency, the resistance and reactance values are well behaved, as shown in Fig. 5.18. An approximate formula for the input resistance of the axial mode helix is

$$R \approx 140C/\lambda\,\Omega \tag{5.7}$$

At 8 GHz, this gives a value of $\approx$129 Ω, whereas FEKO indicates a value closer to 170 Ω. It should be noted that the above formula is to be regarded only as an approximation, so the FEKO result is very credible. Also giving confidence in the FEKO results is the approximately linear increase in resistance, at least in the central part of the frequency band. In practice, such an antenna would probably be

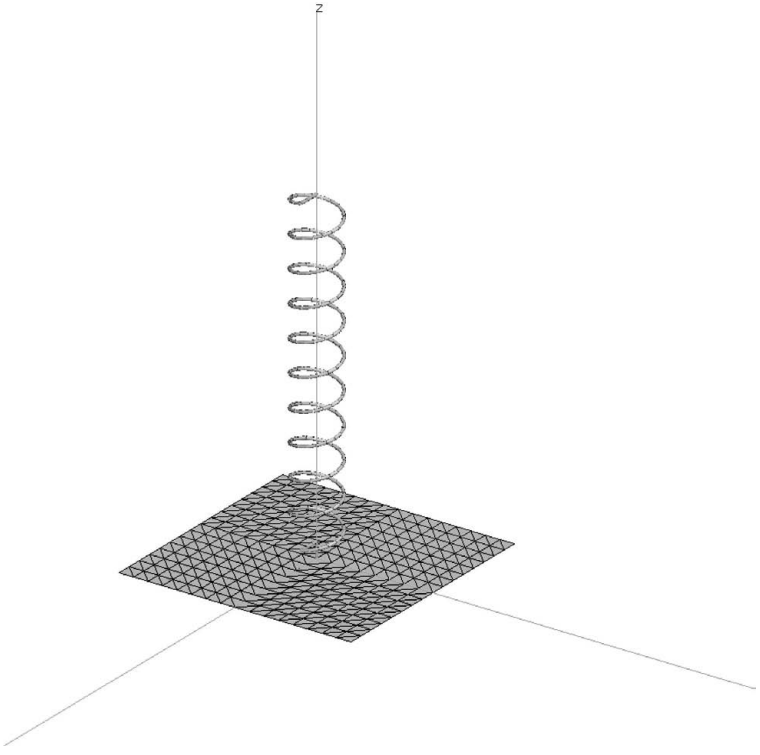


Figure 5.15 The FEKO model of the axial mode helix antenna discussed in the text.

fed via an impedance matching transformer, probably with a 3:1 ratio. As such, a reference impedance of $Z_0 = 150\,\Omega$ is appropriate when plotting the reflection coefficient, which is shown in Fig. 5.19.

To evaluate this antenna fully for an actual design exercise, one should also check the axial ratio of the polarization, since this is an important parameter when designing circularly polarized antennas. This information is also available in the `.out` file, but may require some manipulation to present graphically. More details are available in [8].

In summary, the helix performs well from around 6.2 GHz to at least 10.75 GHz, in terms of impedance match (S_{11} less than -10 dB, assuming a 3:1 impedance transformer for a 50 Ω system, as above) and offering reasonable pattern behavior. This is a bandwidth ratio of 1.73. The gain at the center frequency agrees very well with Kraus's improved formula, and at the lower end of operation, the reflection coefficient shows the same behavior as measured data for a similar (but not identical) helix. The empirical design formulas give reasonable guidelines for gain and half-power beamwidth, but the numerical simulation provides much more

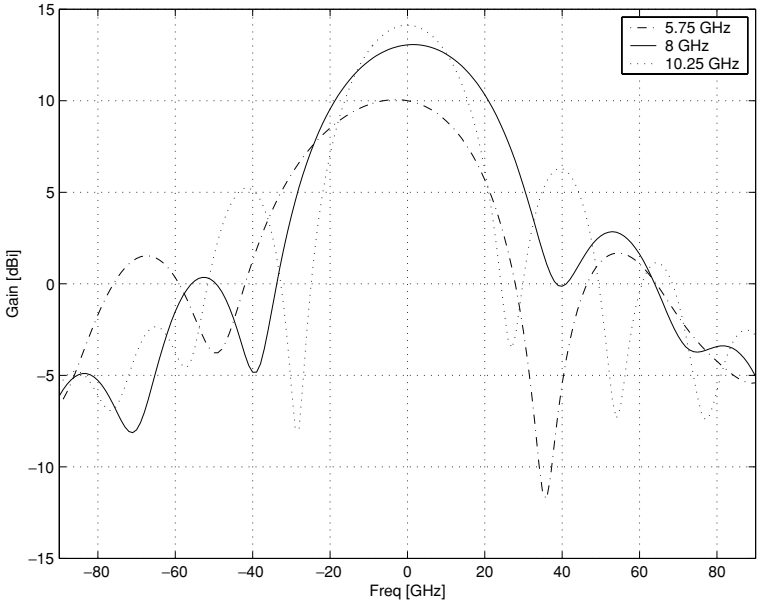


Figure 5.16 The gain of the axial mode helix antenna at the lower, center and upper ends of its operating band.

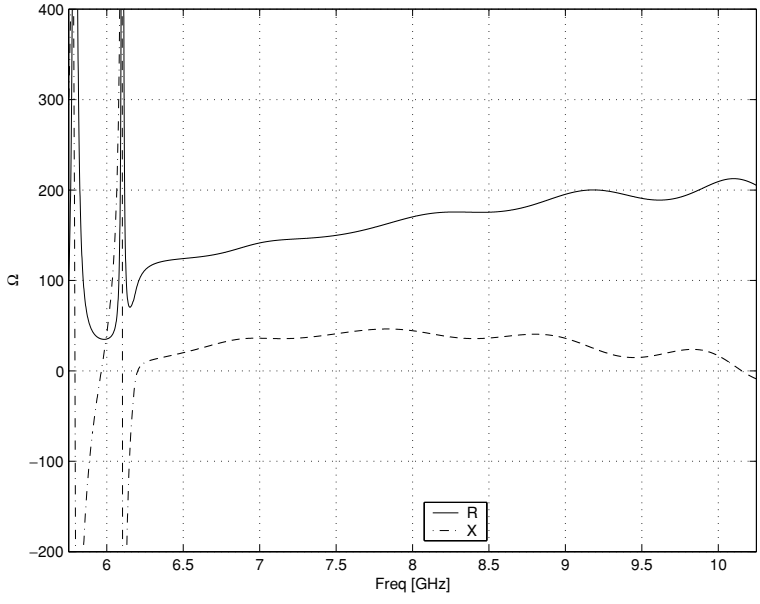


Figure 5.17 Resistance (R) and reactance (X) of the helix antenna across the entire operating band.

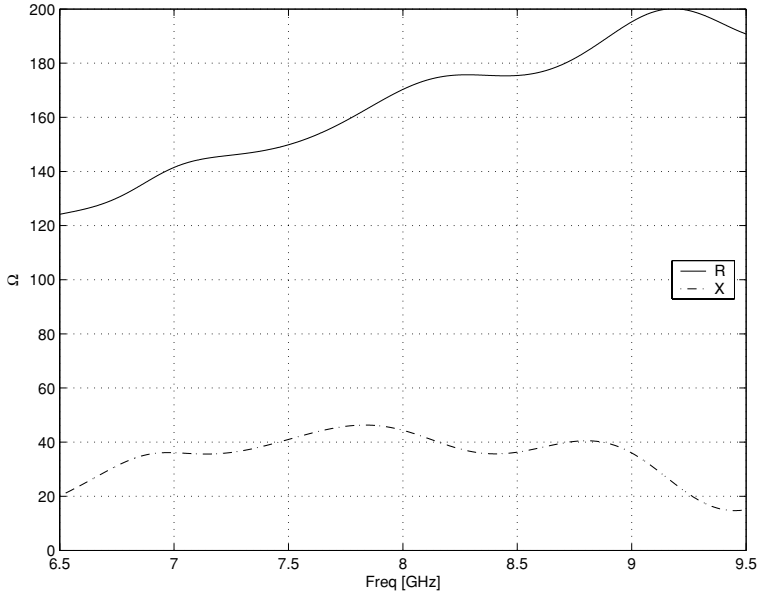


Figure 5.18 Resistance (R) and reactance (X) of the helix antenna near the design frequency.

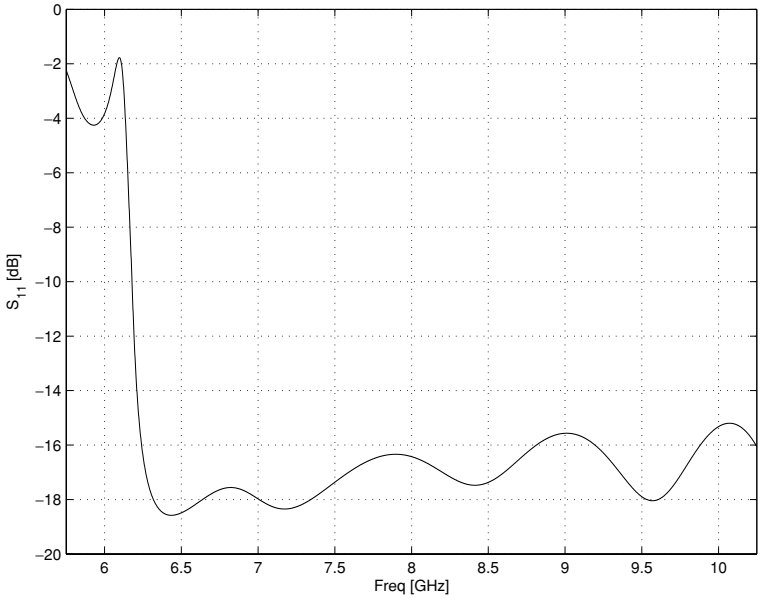


Figure 5.19 Reflection coefficient of the helix.

accurate data. In an actual design, the helix as simulated may be acceptable for the application; if not, one at least is aware that a redesign is likely to be advisable, without even the need first to build a prototype.

Code tips – modelling this structure in NEC

Later versions of NEC-2 included a `GH` card, which permits one to specify a helix or spiral with the same ease as the FEKO model discussed in this section. However, modelling grounds in NEC is more problematic. One is tempted to use the `SP` card, which generates a surface patch model. However, this uses the magnetic field integral equation, which as we will see in Chapter 6 is not suitable for modelling an open structure. Instead, a ground plane will have to be built from a wire mesh, either by hand (there is no automatic means to do this in NEC) or using the Wiregrid for Windows package, which supports this functionality.[a] Wiregrid approximations of surfaces were studied in detail by Ludwig [9], who confirmed using a careful analysis that the long-used "equal area rule" produced a good approximation. This rule requires that the surface area of the wires parallel to one linear polarization when "rolled flat" should equal the surface area of the solid surface. (For an arbitrary polarization, the wire surface area should be doubled.) One quickly see that this implies that segment length $\Delta \approx 2\pi a$, with a the wire radius, which is pushing the limits of the thin-wire approximation. Also, we repeat our earlier warning: one must be very careful to ensure that the helix wire and wires representing the ground plane actually connect.

[a] FEKO includes a `WG` card to do this, although due to its surface meshing capabilities, one will probably not use this too often.

5.6 A Wu–King loaded dipole

Thus far, all the antennas discussed in this chapter were assumed to consist of perfectly conducting wires. (The log-periodic antenna included a terminating resistance, which was introduced to improve the impedance match, although the elements were still assumed to be perfectly conducting.) In practice, the conductance of the metals traditionally used for constructing wire antennas (aluminum, steel etc.) is sufficiently high that this is an excellent assumption. In this example, however, we are going to study an antenna deliberately loaded with resistance – the *Wu–King* resistively loaded dipole. This antenna, first described in [10, 11], has a continuous resistive loading. In practice, this can be made either using thin tubular sections of varying radius and material [10], or by approximating the continuous loading by discrete resistors [12].

Wu and King showed that if the loading on a dipole, half-length h, had the following form:[8]

$$Z(z) = \frac{\eta_0 \Psi}{2\pi h(1 - |z|/h)} \, [\Omega/\mathrm{m}] \tag{5.8}$$

then the current had the following approximate form:

$$I(z) \sim h(1 - |z|/h)\, e^{-jk_0|z|} \tag{5.9}$$

This is clearly a *traveling wave*. By comparison, on the usual half-wave resonant PEC dipole, the current has the *standing wave* form $\sin[k(h - |z|)]$.

The dimensionless parameter Ψ is complex valued, and a function of the electrical dimensions of the antenna. It is usually approximated by its DC value, Ψ_0. It must be computed numerically; typical values are from just under 10 for moderately thick dipoles to around 20 for very thin ones.

We will study the loaded dipole described by Maloney and Smith [12]; for their antenna, the ratio of half-height to radius h/a was 65.8, and $\Psi_0 = 7.79$. For convenience, we will work with $h = 0.25$ m, so that the unloaded PEC dipole resonates close to 300 MHz.

In FEKO, loading can be accomplished using several different cards: `LD`, `LS` and `LP`. The first implements *distributed* loading, in Ω/m, which is what we need here. (The other two cards implement lumped loads in series and parallel respectively.) FEKO loads segments via their label number, and hence one needs to label each segment on the dipole separately. (A FEKO label is the equivalent of a NEC tag.) One way of doing this is shown in Figs. 5.20 and 5.21, where the dipole is first built from individual segments, and then loading is applied to each of these.

The reflection coefficient of the Wu–King dipole is compared to a PEC (unloaded) one of the same dimensions in Fig. 5.22. In these results, two values of loading are shown: the "high" value is as in Eq. (5.8), the "low" value is as given in their original paper, with an 8 instead of 2 in the denominator. The Wu–King dipole has a rather high input impedance (given approximately by $60\Psi_0$), so $Z_0 = 300\,\Omega$ was used when computing the reflection coefficient (for the PEC dipole, $Z_0 = 75\,\Omega$ was used). The loaded dipole clearly has a *much* larger impedance bandwidth, and is indeed a wideband antenna. The rather poor result for the higher loading is due to a large, but slowly varying, reactive component, as shown in Fig. 5.23; this could be removed by adding a tuning component in an actual application, but this has not been done here. Figure 5.24 shows the current distributions along both loaded and unloaded dipoles at 280 MHz, the resonant

[8] Note the major corrections in [11]; the corresponding expression [12, Eq. (1)] is correct.

```
** A resistively loaded (Wu-King profile) dipole.
** As in Maloney and Smith, IEEE T-AP, May 1993 p.668-676.

** Variables
**   Frequency and wavelength
#lam = 1.00        ** wavelength in metre
#freq = #c0/#lam   ** frequency in Hertz
#h = #lam/4        ** half-height of antenna [m]
#seg_rad = #h/65.8 ** radius of wire
#f_l = 200e6
#f_u = 600e6
#psi_0 = 7.79      ** Wu-King parameter
#eta_0 = sqrt(#mu0/#eps0)

** Parameters for segmentation
#seg_ln = #lam/40    ** nominal length of wire segments
IP                            #seg_rad            #seg_ln
#num_sg2 = ceil(#h/#seg_ln)     ** segments on each dipole half (excl. source)
#num_sg = 2*#num_sg2+1          ** to ensure odd number of overall segments
#delta = 2*#h/#num_sg           ** actual length of wire segments
** Geometry of radiating structure
** Has to be constructed with two loops and a special source segment, since a
   separate label
** is required for each segment

** Construct centre (source) segment
#lab = #num_sg
LA   #lab
DP   A                        0.0       0.0       -#delta/2
DP   B                        0.0       0.0       #delta/2
BL   A    B
** Construct upper half
#ell1 = #delta/2
#ell2 = #delta/2+#delta
!!for #ii = 1 to #num_sg2
#lab = #ii
LA   #lab
DP   A                        0.0       0.0       #ell1
DP   B                        0.0       0.0       #ell2
BL   A    B
#ell1 = #ell1+#delta
#ell2 = #ell2+#delta
!!next
.
.
.
```

Figure 5.20 PREFEKO file for the Wu–King loaded dipole, geometry.

```
** Construct lower half
#ell1 = -#delta/2
#ell2 = -#delta/2-#delta
!!for #ii = 1 to #num_sg2
#lab = #ii+#num_sg2
LA   #lab
DP   A                        0.0       0.0       #ell1
DP   B                        0.0       0.0       #ell2
BL   A    B
#ell1 = #ell1-#delta
#ell2 = #ell2-#delta
!!next
** End of geometric input
EG   1    0    0    0    0
```

Figure 5.20 (*Continued*)

```
.
.
.
** Load the structure - again, a loop structure is used.
** Load source segment
#load = #psi_0*#eta_0/(8*#pi*#h)
#lab = #num_sg
LD   #lab                     #load  ** Loss
** Upper half and lower half at same time now:
!!for #ii = 1 to #num_sg2
#z = #ii*#delta
#load = #psi_0*#eta_0/(8*#pi*#h*(1-#z/#h))
#lab = #ii
LD   #lab                     #load  ** Loss
#lab = #ii+#num_sg2
LD   #lab                     #load  ** Loss
!!next

** Set the frequency
FR   41   0                   #f_l               #f_u
** Voltage gap excitation at a segment
#lab = #num_sg
A1   0    #lab                1.0
** Calculate surface currents for current display
OS   1         1
EN
```

Figure 5.21 PREFEKO file for the Wu–King loaded dipole, loading.

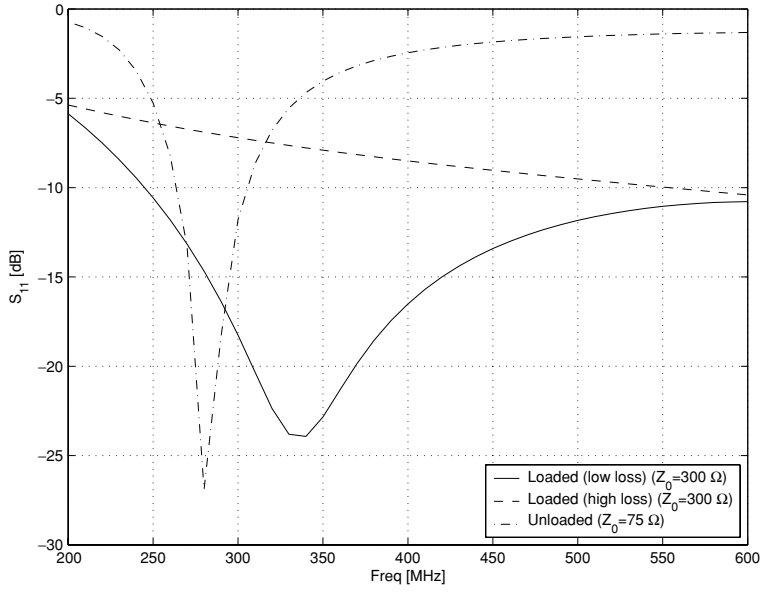


Figure 5.22 The reflection coefficient of the Wu–King dipole compared to a PEC dipole.

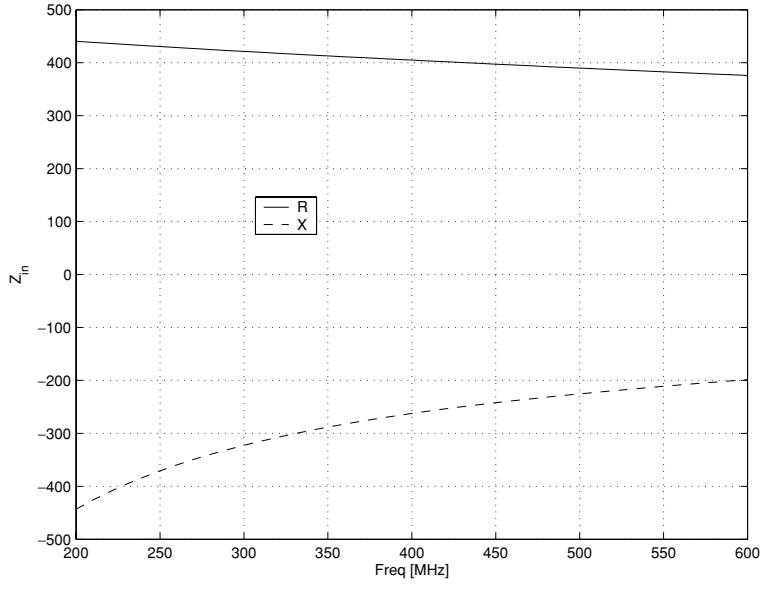


Figure 5.23 The impedance of the Wu–King dipole, "high" loading.

frequency of the unloaded dipole. (The magnitudes have been normalized; the higher impedance of the loaded dipole results of course in smaller values of current.) The loaded dipole with the higher loading clearly supports a traveling wave, with a phase difference along the dipole arm of a little more than the 90° predicted

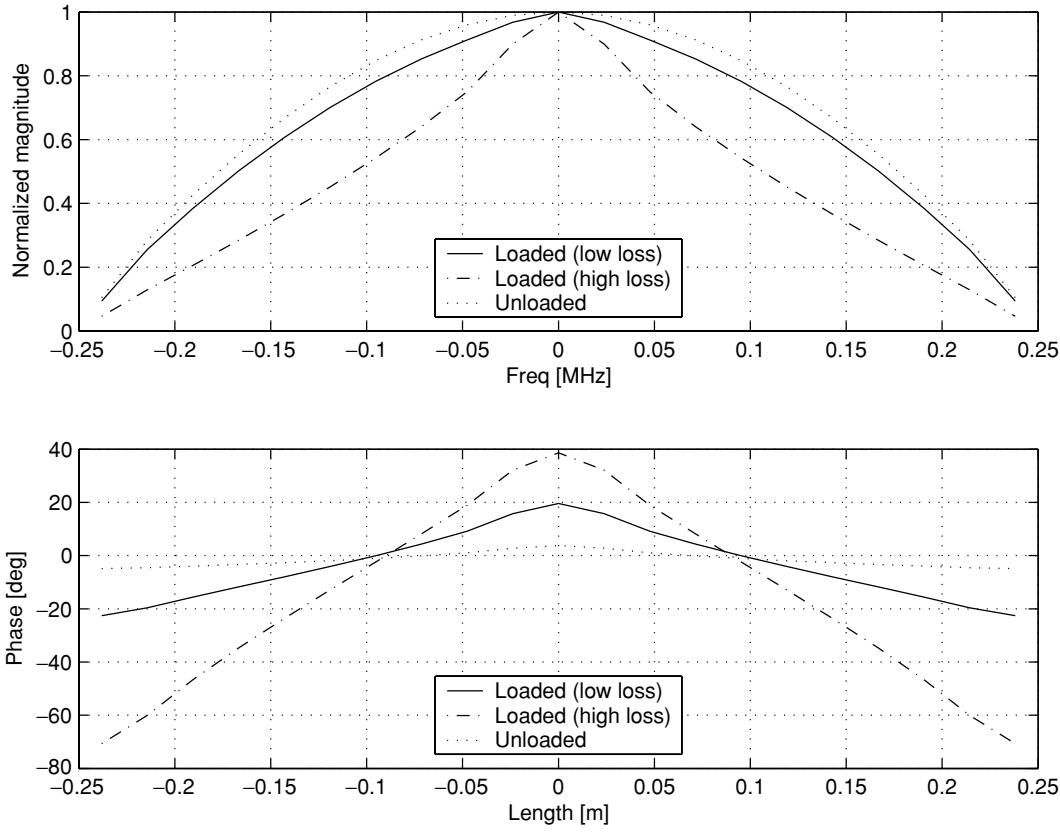


Figure 5.24 The current (normalized magnitude and phase) on the Wu–King and PEC dipoles.

by Eq. (5.9) for $h \approx \lambda_0/4$, and with an almost linear current distribution, also as predicted. The phase for the unloaded dipole is almost constant, as one would expect from a standing wave distribution. The results for the lower loading are somewhere in between the pure standing wave of the unloaded dipole and the pure traveling wave of the dipole with higher loading.

The wide bandwidth is, however, bought at a price: efficiency. Wu and King originally predicted a theoretical efficiency of 50% for $h = \lambda_0/4$, but FEKO shows a *much* lower efficiency of around 7% at 300 MHz (Fig. 5.25). In a subsequent correction [11], Wu and King drastically revised their calculation, predicting a very similar value to the FEKO computation. The result for the lower loading case is around 23%, rather better. Interestingly and serendipitously, the original (incorrect) result by Wu and King provides generally better antenna performance,

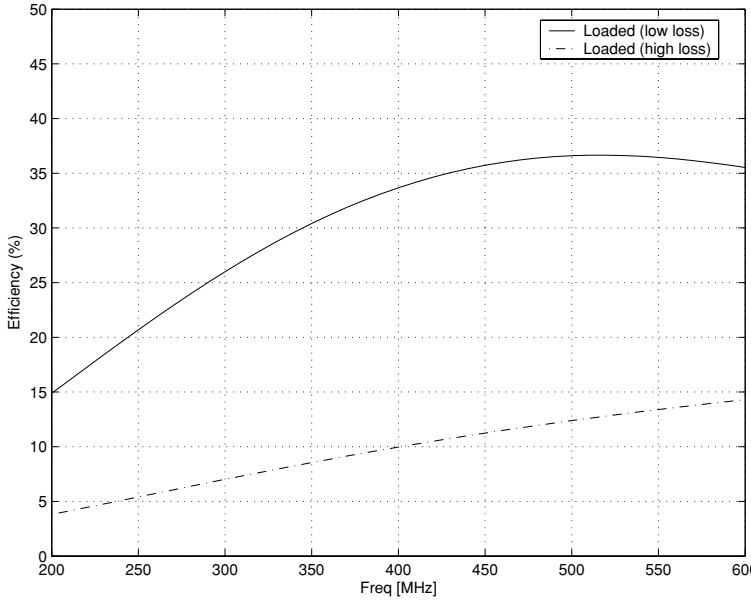


Figure 5.25 The efficiency of the Wu–King dipole.

certainly in terms of reflection coefficient and efficiency, even though the current is not a pure traveling wave.

Unfortunately, the Wu–King loaded dipole is one structure for which measured data are very scarce, and hence we have had to evaluate this model in terms of expected theoretical behavior. Useful measured data were published in [12], but in the *time domain*. (Although FEKO has a time domain option, it is only available for scattering problems.)

Before leaving this structure, a fundamental point should be noted about wideband antennas. The definition of this is inherently a frequency domain concept, and one should be careful to differentiate between a wideband antenna on the one hand, and a *non-dispersive* antenna on the other. The former type of antenna works well over a wide range of operating frequencies; the latter can radiate actual time domain pulses without distortion (obviously, it will also be wideband). A little thought about this from the viewpoint of the Fourier transform shows that this translates to requirements on not just constant magnitude response, but also phase linearity. Many wideband antennas (such as spirals and log-periodics) are dispersive because different frequencies radiate from different parts of the structure. We will not pursue this further, but will mention in closing that the loaded dipole exhibits limited dispersion, and because of this is widely used in time domain antenna systems despite its low efficiency.

Code tips – modelling this structure in NEC

The `LD` card provides the same functionality as in FEKO, but the absence of user-defined variables in NEC means that one will have to compute the loading manually at each segment, so this will be a tedious structure to model in NEC.

5.7 Conclusions

In this chapter, we have discussed modelling thin-wire antennas using FEKO and NEC-2. Starting with a very simple dipole example, we progressed to more complex antennas, including Yagi–Uda and log-periodic dipole antennas, an axial mode helix and a loaded dipole. The helix example also introduced the use of *surface* modelling. We highlighted a number of points which one must be careful with; perhaps the most crucial is to check that the solution is converged (but also not over-converged, due to the limits of the thin-wire approximation). We also emphasized the importance of validation, that is, checking computed results in some way. Historically, comparison to measured data or an analytical solution has been the most convincing method of validation. Nowadays, comparisons with data computed using other codes and/or formulations are increasingly widely used and accepted, and we have directly compared FEKO and NEC-2 results on several occasions, noting that one cannot expect exact agreement. (It has also been commented in this context that measured data must also be used with discretion.) A number of features supported by FEKO (but not NEC-2) which simplify antenna modelling were introduced, including iteration and conditional execution. Several other FEKO and NEC-2 features were also discussed, including the use of labels/tags, transmission lines, and various types of grounds. We also took the opportunity afforded by numerical simulation to improve an antenna design, by adding a terminating resistance to a log-periodic antenna and evaluating the change in antenna performance.

Properly used, within its region of validity, we have seen that the thin-wire formulation is both accurate and very efficient computationally. Having completed this chapter, the reader should feel far more confident in modelling a wide range of wire antennas using tools such as FEKO and NEC-2.

During this chapter, we very briefly touched on the modelling of surfaces in Section 5.5. This is an important part of many antenna designs – and also for scattering problems – and in the next chapter we will comprehensively discuss the modelling of surfaces and volumes using the MoM, as well as the attendant problems of high computational cost.

References

[1] G. J. Burke and A. J. Poggio, "Numerical Electromagnetics Code (NEC) Method of Moments. Part III: User's Guide." Lawrence Livermore National Laboratory, CA, UCID 18834, January 1981.
[2] C. du Toit and D. B. Davidson, "Wiregrid: a NEC2 pre-processor," *Appl. Comput. Electromagn. Soc. J.*, **10**, 31–39, March 1995.
[3] D. B. Davidson, "Parallel processing revisited: a second tutorial," *IEEE Antennas Propag. Mag.*, **34**, 9–21, October 1992.
[4] J. J. H. Wang, *Generalized Moment Methods in Electromagnetics*. New York: Wiley, 1991.
[5] D. J. Janse van Rensburg and D. A. McNamara, "On quasi-static source models for wire dipole antennas," *Microwave Opt. Technol. Lett.*, **3**, 396–398, November 1990.
[6] W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design*. New York: Wiley, 2nd edn., 1998.
[7] C. A. Balanis, *Antenna Theory: Analysis and Design*. New York: Wiley, 2nd edn., 1997.
[8] J. D. Kraus and R. J. Marhefka, *Antennas for All Applications*. Boston, MA: McGraw-Hill, 3rd edn., 2002.
[9] A. C. Ludwig, "Wire grid modelling of surfaces," *IEEE Trans. Antennas Propag.*, **35**, 1045–1048, September 1987.
[10] T. T. Wu and R. W. P. King, "The cylindrical antenna with nonreflecting resistive loading," *IEEE Trans. Antennas Propag.*, **13**, 369–373, May 1965.
[11] T. T. Wu and R. W. P. King, "Correction: The cylindrical antenna with nonreflecting resistive loading," *IEEE Trans. Antennas Propag.*, **13**, 998, November 1965.
[12] J. G. Maloney and G. S. Smith, "A study of transient radiation from the Wu-King resistive monopole – FDTD analysis and experimental measurements," *IEEE Trans. Antennas Propag.*, **41**, 668–676, May 1993.

6

The method of moments for surface modelling

The helix antenna discussed in the previous chapter used a new type of element to model *surfaces*. The theory underlying this is described in this chapter. Not only is the basic theory quite complex, but implementations are especially challenging, so we focus largely on an introductory discussion, followed by some examples of using available codes, rather than going into the frequently lengthy details of full 3D implementations. We will see that not only can perfectly (or highly) conducting structures be efficiently modelled using surface currents, but also homogeneous dielectric and/or magnetic regions, using fictitious equivalent currents. (We will even briefly describe how *in*homogeneous bodies can be modelled using volumetric currents, but note at the outset that this is not one of the strong points of the MoM.) Modelling surfaces is far more computationally expensive than modelling wires, and some methods for reducing the computational cost will also be discussed. These include a hybrid of the MoM and physical optics, and the general class of fast methods, including both those based on the FFT and the fast multipole method. We will also briefly touch on the use of parallel processing.

6.1 Electric and magnetic field integral equations

Following the same lines as the Pocklington equation (Chapter 4), integral equations in either the magnetic or electric fields can be derived for problems with currents flowing on surfaces. The derivation is quite complex, and only the results will be presented here. One integral equation couples the incident electric field to the induced surface current, and is known as the electric field integral equation (EFIE):

$$\hat{n} \times \vec{E}^{\text{inc}}(\vec{r}) = \hat{n} \times \int_S \Big[jk\eta \vec{J}_S(\vec{r}\,') G(\vec{r}, \vec{r}\,') + \frac{\eta}{jk} \{\nabla'_s \cdot \vec{J}_S(\vec{r}\,')\} \nabla' G(\vec{r}, \vec{r}\,') \Big] dS', \qquad \forall\, \vec{r}, \vec{r}\,' \in S \quad (6.1)$$

The ∇' operator implies differentiation in the *source* coordinates. $\hat{n}$ is the unit vector on the surface S. $G(\vec{r}, \vec{r}\,')$ is the scalar free-space Green function given by

$$G(\vec{r}, \vec{r}\,') = \frac{e^{-jkR}}{4\pi R} \tag{6.2}$$

$$R = |\vec{r} - \vec{r}\,'| \tag{6.3}$$

Equation (6.1) is valid for both closed and open surfaces. In the latter case, $\vec{J}_S$ is the sum of surface currents on both sides of the sheet.

The other integral equation couples the incident magnetic field to the induced surface current, and is known as the magnetic field integral equation (MFIE):

$$\frac{1}{2}\vec{J}_S(\vec{r}) = \hat{n} \times \vec{H}^{\text{inc}}(\vec{r}) + \hat{n} \times \oint_S \vec{J}_S(\vec{r}\,') \times \nabla' G(\vec{r}, \vec{r}\,') dS', \qquad \forall\, \vec{r}, \vec{r}\,' \in S \tag{6.4}$$

This is valid *only* for closed surfaces. (The reason is not by any means straightforward, and emerges during the derivation thereof.) It is interesting to note that if we neglect the surface integral, what remains is the physical optics approximation, $\vec{J}_S(\vec{r}) = 2\hat{n} \times H^{\text{inc}}(\vec{r})$, of which more later.

The integrals in the above should be interpreted as the principal value of the integral. (The principal value of an integral with a singularity at $\vec{r}_0$ is essentially the value of the integral with a δ neighborhood around $\vec{r}_0$ removed; then the limit as $\delta \to 0$ is found.) In both these equations, the presence of singularities raises delicate issues and requires careful treatment. The simple expedient of slightly offsetting field and source points as was done with the one-dimensional wire problem (in that case, by treating the source as a filament on the wire axis, but still imposing the boundary condition on on the surface of the wire) can still be done, although in this case one offsets the quadrature points corresponding to source and field points rather than concentrating the source elsewhere.

Mathematically, the EFIE is a *Fredholm integral equation of the first kind* – the unknown is present only in the kernel. The MFIE is a Fredholm integral equation of the *second kind* – the unknown is present both inside *and* outside the kernel. The reason for the difference is due to the boundary condition. The EFIE and MFIE are both derived from the Statton–Chu formula, which states that for points *on* the surface, the following relations hold [1, p. 172]:

$$\vec{E}^{\text{inc}}(\vec{r}) + \text{PV} \int_S \vec{E}_s \, dS' = \frac{1}{2}\vec{E}(\vec{r})$$

$$\vec{H}^{\text{inc}}(\vec{r}) + \text{PV} \int_S \vec{H}_s \, dS' = \frac{1}{2}\vec{H}(\vec{r}) \tag{6.5}$$

The PV here reminds us that these are the principal values of the relevant integrals. In the equations above, E_s and H_s are not directly the scattered fields, but rather the kernels which are integrated to obtain these (the full expressions may be found in [1, p. 173]). For a PEC, the boundary condition on $\hat{n} \times \vec{E}$, i.e. tangential E, is of course zero, whereas $\hat{n} \times \vec{H}$ is the surface current $\vec{J}_s$; hence the different nature of the two integral equations. For more details, see [1, Section 12.3].

Mathematically, it is well known that Fredholm type two integral equations are generally more *well posed* – this motivated much work using the MFIE. (Put simply, a well-posed problem is one whose solution is not strongly dependent on the physics and geometry of the problem.) However, the requirement for a *closed* surface S is frequently a problem in applied CEM work, with the result that the EFIE is usually preferred in practical codes. Finally, linear combinations of the EFIE and MFIE have also been used; not surprisingly, this method is known as the combined field integral equation (CFIE). The CFIE will not be discussed here.

Because the EFIE and MFIE are both quite complex, it is convenient to introduce a simplifying notation. As an example, for the EFIE, the right-hand side of Eq. (6.1), which represents the scattered field, is often written in the following shorthand:

$$\mathbf{E}_s = \mathcal{L}_J^E \mathbf{J}^{MM}$$

$\mathcal{L}$, which represents all the mathematical operations to be performed on the current $\vec{J}$, is known as an *operator* – it is an extension of the concept of a function.

A mathematical aside – functions, functionals and operators

A function, of course, maps a number (integer, real or complex) to another number; a functional maps a function to a number; and an operator maps a function to another function. (We will encounter functionals in Chapter 9.) The Fourier transform is a commonly encountered example of an operator: it maps a function of time to a function of frequency (or more generally, from one domain to the corresponding spectral domain). Operator notation will be used subsequently in this chapter.

6.2 The Rao–Wilton–Glisson (RWG) element

When dealing with surfaces using the MoM, two matters need attention. The first is that we need to split the geometry up into small elements. The simplest approach, and the first one explored historically in codes such as NEC-2, was to use square (or rectangular) patches. However, for general two-dimensional geometries,

triangular elements are better for approximating the geometry, and this is the approach which most modern codes (including FEKO) use. The second matter is that the physical parameter being approximated, $\vec{J}_S$, is now two dimensional. The basis function must also incorporate this.

In this context, a very widely used basis function for the triangular patch was introduced by Rao, Wilton and Glisson in 1982 [2]. The basis function is often known simply as the RWG element. Subsequent work led to the realization that this basis function is very closely related to the edge-based elements widely used in contemporary finite element analysis. We will return to this later in this book when we address finite elements.

The basis function includes some new features which have not yet been encountered in this book. Most importantly, the basis function is *vector* in nature, which means that the individual scalar components (eg J_x, J_y, and J_z), can only be recovered with some manipulation. The essential idea is to enforce current continuity over an *edge* of a patch. The interpolation function used to achieve this is the following:

$$\vec{f}_n(\vec{r}) = \begin{cases} \frac{l_n}{2A_n^+}\,\vec{\rho}_n^+ & \forall \vec{r} \text{ in } T_n^+ \\ \frac{l_n}{2A_n^-}\,\vec{\rho}_n^- & \forall \vec{r} \text{ in } T_n^- \\ 0 & \text{otherwise} \end{cases} \tag{6.6}$$

Figure 6.1 defines the vectors $\vec{\rho}_n^+$ and $\vec{\rho}_n^-$. Note that the basis function is defined over two adjoining triangles T_n^+ and T_n^- which share a common edge. A_n^+ is the area of triangle T_n^+ (and similarly A_n^-). l_n is the length of the shared edge. The vector $\vec{\rho}_n^+$ is the vector position within triangle T_n^+, with the left-hand node of T_n^+ as origin; similarly, $\vec{\rho}_n^-$ is the (negative of the) vector position, with the right-hand node of T_n^- as origin. There exists a coordinate system known as *simplex coordinates* which makes the study of interpolation functions on triangles much simpler,

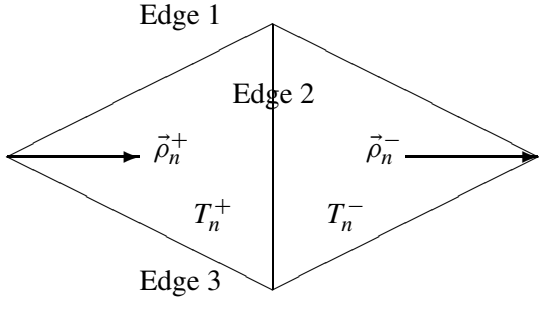


Figure 6.1 The two connected triangles T_n^+ and T_n^-, sharing a common edge, which support a RWG basis function.

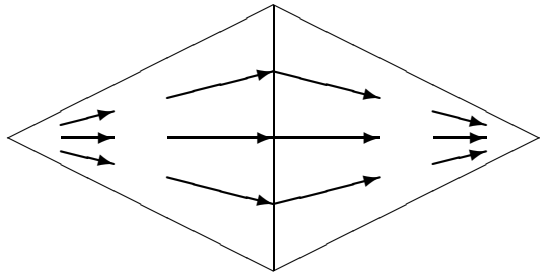


Figure 6.2 A vector plot of the RWG basis functions.

and is widely used in finite element analysis; these vectors can be written rather simply in that coordinate system. The terms $l_n/2A_n^-$ and $l_n/2A_n^+$ are normalizing constants.

The resulting current interpolation is shown in Fig. 6.2. The following points may not be immediately apparent. Firstly, it should be noted that this basis function has *no* component normal to the upper or lower sides of either of the triangles, but only to the central (shared) edge. Without more detailed theoretical analysis, the following is not obvious, and is stated without proof here:[1] the current crossing this shared edge is linearly interpolated in the tangential direction (i.e. along the edge) and interpolated as a constant normal to (i.e. across) the edge. This latter value is usually the "degree of freedom" (the unknown value of current) which is associated with this basis function; the current associated with this edge is thus approximated as $\vec{J}_n(\vec{r}) \approx I_n \vec{f}_n(\vec{r})$. Note that all these terms are expressed in terms of the local coordinates on the triangle; again, the conversion to Cartesian coordinates is readily performed using simplex coordinates.

What of the current flowing across the two *other* edges? To approximate these, one defines *additional* basis functions on each of the other two connected triangles; thus on any one triangle, there are *three* such basis functions, with three associated unknowns, which are the normal components of current on each edge. Within the element, it should be appreciated that the *total* current is thus approximated by the sum of these three basis functions. With the edges numbered as on Fig. 6.1, the total current on triangle T_n^+ is given by:

$$\vec{J}(\vec{r}) \approx I_1 \vec{f}_1(\vec{r}) + I_2 \vec{f}_2(\vec{r}) + I_3 \vec{f}_3(\vec{r}), \qquad \forall \vec{r} \text{ in } T_n^+ \tag{6.7}$$

At the risk of repetition, note that the basis functions carry the vector information; the unknowns for which the code solves (I_1, I_2 etc.) are just scalars.

[1] Again, because this RWG basis function is so intimately related to the edge-based Whitney function of finite element analysis, we postpone detailed mathematical analysis of this class of element until Chapter 9.

Some MoM formulations use what is called the "mixed potential integral equation" – MPIE. In this case, charge is also present as an unknown. From the continuity equation,[2] this implies that charge will be thus be approximated as piecewise constant. We will see an example of the MPIE in Chapter 7.

With the integral equations posed, and suitable basis functions defined, one is now in a position to solve problems involving surfaces using the MoM. We will not attempt to implement such a code, since this is a complex task, but rather move directly to the study of a problem using a commercial package. The problem we choose is one of the classic canonical problems of electromagnetic analysis, namely scattering from a PEC sphere in the resonance regime.

6.3 Some examples of surface modelling

6.3.1 Scattering from a sphere

One of the classical problems of analytical electromagnetics was that of scattering from a sphere. Early work on this was done in the nineteenth century by Lord Rayleigh (John William Strutt, 1842–1919), who has lent his name to the general field of scattering from electrically small objects. For electrically small spheres, Lord Rayleigh showed that scattering was proportional to the fourth power of frequency; this permitted him to explain the color of the sky. For electrically large spheres, the scattering cross-section is simply the cross-sectional area of the sphere. In between these extremes, the *resonant regime* is encountered, where energy creeping around the surface of the sphere results in constructive and destructive interference. The process of electromagnetic scattering will be recalled from Chapter 3.

A brief historical aside – why *is* the sky (usually) blue?

The color of the sky is due to the presence of the earth's atmosphere. On the moon or in space, the sky appears black. For our present purposes, we can view the atmosphere as consisting of a large number of small particles and molecules in suspension. These are considerably smaller than the wavelength of visible light (approximately 400 to 700 nm), so that the scattering from each particle is proportional to $1/\lambda^4$, as in the text. Hence, the scattering from the violet (short-wavelength) end of the spectrum is almost an order of magnitude larger than that from the red (long-wavelength) end. The spectral irradiance of sunlight – see for example [3, Fig. 7.49] – which peaks near the wavelength of blue light,

[2] The time rate of change of charge is the negative of the divergence of the current.

470 nm, and varies by about 30% over the visible spectrum, makes the overall calculation slightly more complex. It is this scattered radiation which colors the sky blue. (It is worth noting in passing that the scattered light is also polarized, although we will not pursue this here.) At sunset, however, the radiation has to pass through much more of the atmosphere, and the blue scatters out completely, leaving the red sunset. When there is dust in the air, this exacerbates the effect, leading to spectacular sunsets. More details of this may be found in many texts, such as [4, Chapter 12] and [3, Chapter 7]. The latter has a particularly insightful discussion, and also provides extensive historical background on this topic.

The echo width of a three-dimensional target is also known as its radar cross-section (RCS). It is usually abbreviated σ. The RCS is defined as follows:

$$\sigma(\theta, \phi, f) = \lim_{R \to \infty} 4\pi R^2 \frac{|E^{\text{scat}}|^2}{|E^{\text{inc}}|^2} \tag{6.8}$$

R is the distance to the target. The dimensions of the RCS are square meters, since it is in essence an equivalent area. Frequently, results are given in dB form, and quite often normalized to 1 m^2, in which case the symbol dBsm is often used. The RCS of a target is in general a function of orientation and frequency, and this has been explicitly indicated above. Note that this definition is entirely equivalent to

$$\sigma(\theta, \phi, f) = \lim_{R \to \infty} 4\pi R^2 \frac{P^{\text{scat}}}{P^{\text{inc}}} \tag{6.9}$$

The RCS is a far-field parameter; once the surface currents are have been found using the MoM, the radiated fields may be computed in a straightforward fashion using standard antenna theory.

As a simple example of a scattering problem, we will now study the RCS of a sphere. A highly conducting sphere with a radius of 5 cm will be chosen; this is the typical size of anti-personnel landmines (although of course these are generally buried, and also unfortunately usually made largely of non-metallic materials to make detection even more difficult). We expect the interesting resonance interactions to occur when the circumference of the sphere is of the order of a wavelength, hence $\lambda \approx 2\pi a$. This corresponds to a frequency of around 1 GHz. Running the simulation from 300 MHz to 6 GHz should produce some interesting results.

Note that we are only going to investigate the *back-scatter* from the sphere; hence, only one RCS angle is required (the same one the field is incident from). The results of the analysis are shown in Fig. 6.3. The RCS has been normalized by the high-frequency limit πa^2 to illustrate more clearly the different scattering

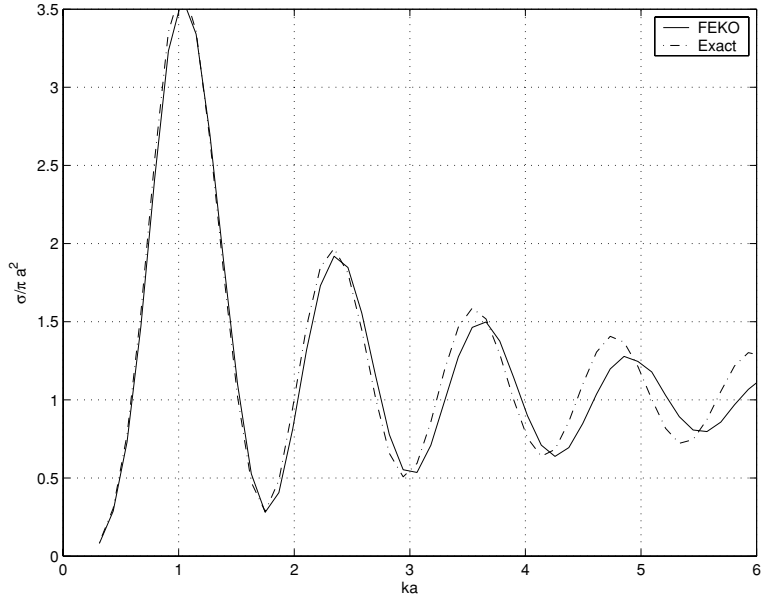


Figure 6.3 Normalized RCS of a PEC sphere plotted against circumference in wavelengths ka.

regimes; note how the RCS initially climbs steeply (this is the Rayleigh scattering regime), then oscillates sharply through the resonance regime, before finally converging to the high-frequency limit. The horizontal axis has also been normalized, by plotting $ka = \frac{2\pi}{\lambda}a$ (the sphere circumference in wavelengths). Note the peak as expected at $ka = 1$. Also shown on this plot is the exact analytical solution, computed as a sum of spherical Hankel functions [5, Eq. (11–247), p. 657] – more on this shortly. When compared with the exact solution, we note that the accuracy with which the resonances are computed decreases as the frequency increases.

If we were to analyze this problem over a rather larger frequency band, we would find that eventually, the result should converge to the high-frequency limit. We cannot do this with the present file, because our discretization will not be sufficiently fine for frequencies much beyond 6 GHz. Refining the discretization will result in far longer execution times. However, some thought about the problem shows that we can use symmetry to generate a more efficient solution. The incident electric field is $\hat{x}$ polarized, traveling in the $-\hat{z}$ direction. As such, there is a plane of electric symmetry in the plane $x = 0$. Similarly, there is a a plane of magnetic symmetry in the plane $y = 0$. Finally, there is a plane of geometrical symmetry in the plane $z = 0$. (In this last plane, the geometry is symmetrical, but not the excitation.) Results for a wider frequency range computed using symmetry are shown in Fig. 6.4. Note the improvement in the resonances when compared to

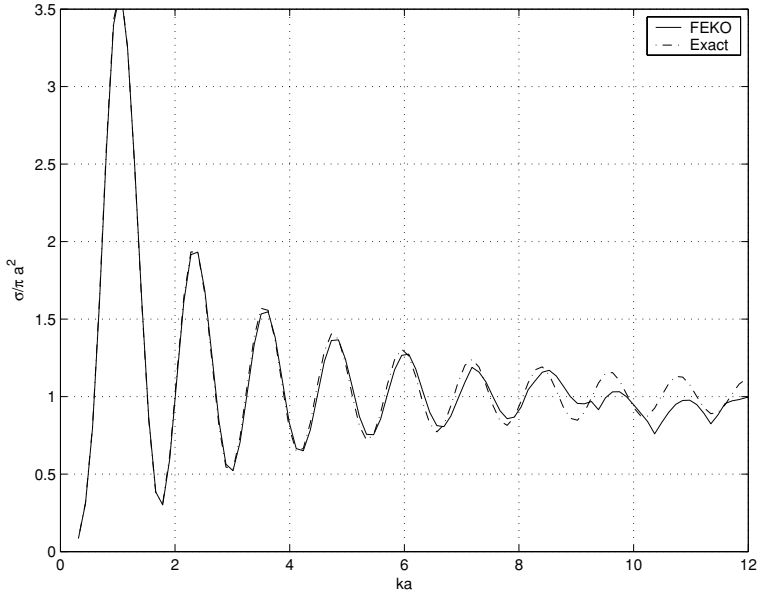


Figure 6.4 Normalized RCS of a PEC sphere plotted against circumference in wavelengths ka. Results were computed exploiting symmetry.

Fig. 6.3. However, at the high-frequency end, the mesh is too coarse even with this model, as is clear by comparison with the analytical solution.

Modelling hints – modelling spheres

All meshers generate some approximation of the actual spherical surface; in the case of FEKO, the triangular mesh is inscribed within the sphere. (FEKO provides the `KU` card to generate a spherical section or a sphere, which makes modelling the sphere very straightforward.) The model can be improved by using a slightly larger radius, chosen to provide the same surface area as the sphere. Conveniently, FEKO computes the surface area of the triangles; for the first model, the area was 0.03096 m^2, whereas the surface of a 0.05 m radius sphere should be 0.03142 m^2. Increasing the radius by 1.007, the square root of this ratio, should provide a slightly better model.

A couple of closing comments on this study would be in order. Firstly, because a sphere is rotationally symmetric, we could have used a field incident from *any* angle. The choice of the $\hat{x}$-polarized field, traveling in the $-\hat{z}$ direction, was however convenient. Note that if a different incident field were used, results would (or should) be very similar, but would not be identical, since the mesh is slightly

"directional." Similar comments apply if one compares results computed using a sphere created using symmetry to results from a sphere created directly in its entirety.

Finally, an important point about the physics and engineering of scattering should be made. We computed the RCS in only one direction – straight back in the direction of the incident field. In applied physics, this parameter is generally known as the back-scatter cross-section; in radar engineering, this is called the monostatic RCS, and is the parameter usually used in radar systems analysis. It is (normally) the parameter appearing in the radar range equation. Most radars are monostatic, which means that they use the same antenna for transmit and receive, or at least the transmitter and receiver are located very close to one another. As already mentioned, the monostatic RCS *of a sphere* is not a function of angle – note that this is the only structure of which this is true! However, there is another type of RCS, *bistatic* RCS. In this case, the transmit and receive antennas are *not* in the same location, and the angles of incidence and reflection are no longer the same. (Very few bistatic radars have been built, even fewer – if any – deployed operationally.) Although the monostatic RCS of a sphere is not angle dependent, the bistatic RCS is. The bistatic RCS can also be computed efficiently using MoM codes, since it requires only a different excitation vector to be computed.

6.3.2 The analytical solution

The exact solution of scattering from a PEC sphere, plotted in Figs. 6.3 and 6.4, is one of the classic analytical solutions in electromagnetics, dating back to the turn of the previous century. Nonetheless, despite the venerable status of the solution, there are some points which are worth making about it, and indeed about analytical solutions in general.

A brief historical aside – Mie scattering

The analytical solution for scattering from a PEC sphere was originally derived by Mie and published in 1908, and the solution bears his name to this day. Debye undertook a very similar study, published in 1909. For details, see [6, p. 415]; for elegant sketches of the fields for the first four modes, reproduced from Mie's paper, see [6, p. 567]. Stratton's book is unfortunately difficult to obtain nowadays; the derivation may also be found in somewhat more recent texts, such as [7, Chapter 6], and a particularly detailed derivation is given in [5, Section 11.8].

The monostatic RCS is given by the following expression:

$$\sigma = \frac{\lambda^2}{4\pi}\left|\sum_{n=1}^{\infty}\frac{(-1)^n(2n+1)}{\hat{H}_n^{(2)\prime}(ka)\hat{H}_n^{(2)}(ka)}\right|^2 \tag{6.10}$$

with a the radius of the sphere and k the free-space wavenumber. The function $\hat{H}_n^{(2)}(ka)$ is the alternative spherical Hankel function. It is related to the regular cylindrical Hankel function by [5, p. 938]

$$\hat{H}_n^{(2)}(x) = \sqrt{\frac{\pi x}{2}}H_{n+1/2}^{(2)}(x) \tag{6.11}$$

The prime in $\hat{H}_n^{(2)\prime}(ka)$ indicates differentiation with respect to the argument.

This would really appear to be a relatively straightforward formula to implement. MATLAB provides only the regular cylindrical Hankel function, but the scaling required by Eq. (6.11) is very easy to implement. For FORTRAN implementations, routines are available in [8, Chapter 6], although one will need to build the Hankel function from its constitutive Bessel functions of the first and second kinds, viz. $H_p^{(2)}(x) = J_p(x) - jY_p(x)$. The derivative requires some simple manipulation to evaluate, using the rule for the differentiation of products applied to Eq. (6.11), and the standard identity [5, p. 936]

$$\frac{d}{dx}[H_p^{(2)}(\alpha x)] = -\alpha H_{p+1}^{(2)}(\alpha x) + \frac{p}{x}H_p^{(2)}(\alpha x) \tag{6.12}$$

to obtain:

$$\hat{H}_n^{(2)\prime}(x) = \frac{1}{2}\sqrt{\frac{\pi}{2x}}H_{n+1/2}^{(2)}(x) + \sqrt{\frac{\pi x}{2}}\left[-H_{n+3/2}^{(2)}(x) + \frac{n+\frac{1}{2}}{x}H_{n+1/2}^{(2)}(x)\right] \tag{6.13}$$

Hence, Eq. (6.10) can be implemented within a few lines of code. However, one needs to be cautious! Routines to compute Bessel functions (by which we include Hankel functions) are not bulletproof. In particular, when the argument (ka in this case) or the order (n) becomes very large, the results lose accuracy. Good implementations should warn of such problems: MATLAB, for instance, provides five different error flags, ranging from warnings of possible loss of precision to outright error messages and not returning a numeric value at all. *One must check such error flags!* In the present case, exceeding some hundred terms or so is sufficient to trigger error messages.

Needless to say, the infinite sum in Eq. (6.10) must also be truncated at some point. In Fig. 6.5, results are shown for the RCS for the sphere as the maximum number of terms is increased; this has been graphed on a semi-logarithmic scale, so that the variation is more easily seen. Plotting against ka is especially insightful,

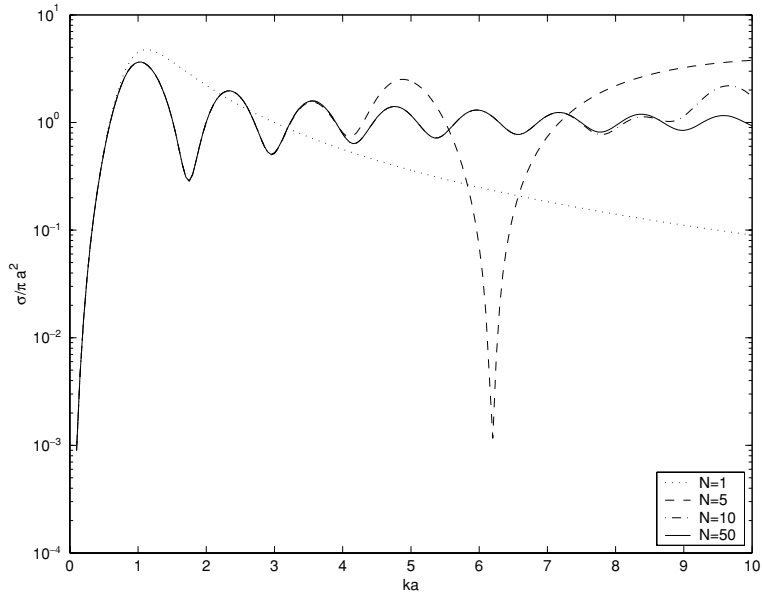


Figure 6.5 Convergence of the analytical solution for the RCS of a PEC sphere, as a function of the number of terms used.

since it is clear that the number of terms required is approximately equal to this product. (This is not coincidental: these terms correspond to circumferentially varying modes, and modes with significantly more rapid variation than ka contribute primarily to the reactive near-field only.)

For electrically large spheres, Eq. (6.10) is clearly going to be problematic to evaluate directly, and one needs to use asymptotic forms to retain accuracy.

A philosophical aside – on "exact" analytical solutions

The above discussion raises a number of interesting points about the nature of "exact" analytical solutions. Critics of our present-day reliance on numeric codes sometimes forget that even pristine analytical solutions are usually approximate in reality, when it comes to evaluating them; such solutions, derived from separation of variables and suitable special functions, usually involve infinite summations which must in practice be truncated. Furthermore, the evaluation of the special functions is almost always done computationally nowadays, and as we have commented, this process is by no means always reliable. (Even tables of functions are not always error free.) It is perhaps the ultimate irony that the author verified his MATLAB implementation of Eq. (6.10) by comparing the results to FEKO computations ...

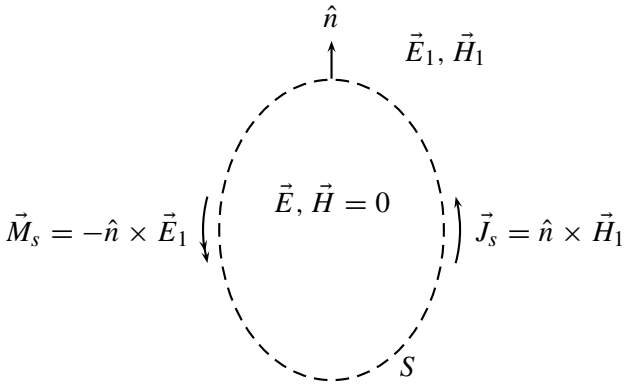


Figure 6.6 Love's form of the equivalence principle.

6.4 Modelling homogeneous material bodies using equivalent currents

In the preceding section, we discussed modelling structures consisting of PEC (perfect electric conductor) material.[3] The current which the MoM computes in this case is the real, physical current, and is what would be measured were one to probe the surface current using a loop, for instance. However, there is another interesting application of surface currents: modelling homogeneous material bodies, that is, dielectric (or magnetic) regions.

All of these rest on the application of the *surface equivalence theorem*, first introduced in 1936 by Schelkunoff. It states that the fields outside an imaginary closed surface can be obtained by placing, over the closed surface, suitable electric and magnetic current densities that satisfy the boundary conditions. Furthermore, the fields inside the surface can be chosen essentially arbitrarily, since the problem is only "equivalent" in the exterior region. When this imaginary surface coincides with a real surface, interesting physics emerges with specific choices of the internal fields. For PEC modelling, the form of the equivalence principle which is generally used is *Love's equivalence principle*, illustrated in Fig. 6.6 for a general surface. With this form, the fields inside the body are assumed zero; since the boundary condition at a PEC surface requires that the tangential total electric field be zero (and hence also the magnetic surface current), only the electric surface current is non-zero and since it is equal to $\hat{n} \times (\vec{H}_{\text{tot}} - 0)$, where $\vec{H}_{\text{tot}}$ is the total magnetic field just above the surface, and the 0 represents the internally zeroed fields, it is also the *actual* current. It is also very convenient because since the field has been chosen as zero in the internal region, the material in this region can be replaced arbitrarily; usually, it is chosen to have the same value as the exterior region, which

[3] The approach can be extended to work for highly conducting structures.

is usually free space in antenna problems.[4] This is *very* important, since it permits the use of the free space Green function – we usually apply this without fully discussing the underlying justification.

In passing, note that there is another variant of this principle which one quite often encounters in the theoretical analysis of aperture antennas. In this case, instead of replacing the internal region with free space, one uses a PEC body. If this is a half-space, one can then use image theory and hence the Green function for free space (again) to solve the problem. There are yet other forms which are useful in specific circumstances.

When the material body is an homogeneous dielectric or magnetic structure, we can apply the same approach as with the PEC body; there are two differences, however. Firstly, the currents are now fictitious (in other words, one would not be able to measure them with some cleverly devised experiment), and secondly, both electric and magnetic equivalent surface currents are required.

6.5 Scattering from a dielectric sphere

Having just discussed a PEC sphere, it is now an interesting exercise to repeat the analysis for a dielectric sphere. The model is very similar to the PEC sphere. Results are shown in Fig. 6.7. A moderate value, $\epsilon_r = 4$, has been chosen for the relative permittivity, otherwise the sphere has a very low signature. Results are normalized to the asymptotic limit for the PEC sphere, πa^2. It is interesting that the RCS of the dielectric sphere exceeds that of the PEC sphere for $ka > 1.5$. Both unfortunately and surprisingly, there do not appear to be computed data for this particular problem of RCS versus ka widely available in the literature, although the analytical solution has long been well known. In [9, Fig. 6a], results are given for the bistatic scattering from a $ka = 3$ sphere, and the similarly normalized result for $\theta = 0$ is around 25; the FEKO result is a little smaller but in the same region at $ka \approx 3.3$.

To validate this computation, we can use another approach for modelling dielectrics available within FEKO, namely equivalent *volumetric* currents. In this case, the entire volume is meshed using cubical cells – this permits the material properties to vary from cell to cell, but at much higher computational cost (we will discuss this shortly). Results computed using the volumetric approach, as well as a surface current model using a slightly finer mesh, are also shown on Fig. 6.7. The agreement between the surface and volume formulations is very good up to just above $ka = 2$, which is about the point at which the mesh density drops below

[4] Note that the whole argument also works in reverse for the interior region: in this case, it is the fields in the exterior region which are arbitrary. This is not very useful in antenna modelling, however.

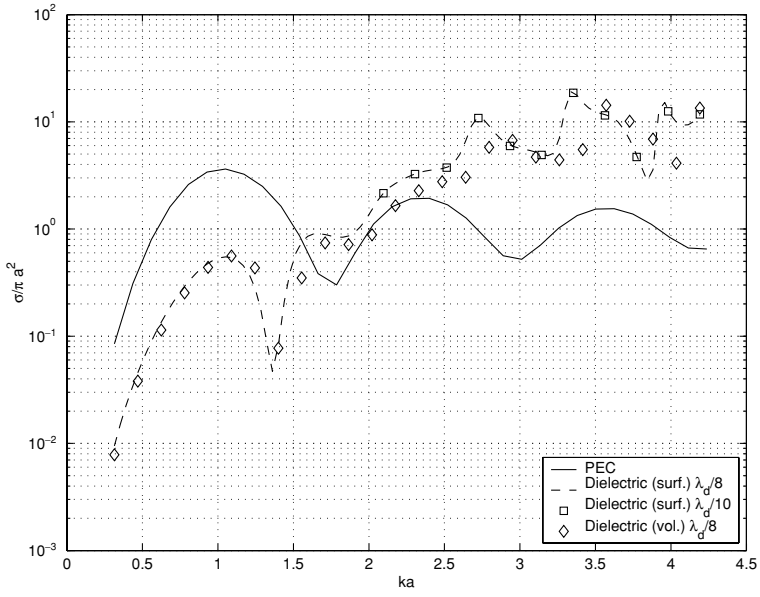


Figure 6.7 RCS of a dielectric sphere, radius a, $\epsilon_r = 4$, compared to a PEC sphere. All results are normalized by πa^2. k is the free-space wavenumber.

$\lambda_d/10$, where λ_d is the wavelength in the dielectric . (Because we are effectively modelling fields in an electrically denser medium, it is the wavelength *in the dielectric* which concerns us.) Since the volume approach meshes the sphere with small rectangular cubes, as opposed to a conforming triangular surface mesh, one can expect the volume approach to be slightly less accurate geometrically, in particular at higher frequencies. This is confirmed by a calculation using a slightly smaller edge length for the surface mesh; the agreement between the two surface current meshes is good.

A note of caution here: such intracode validation is usually questionable, but in this case, FEKO is using two quite different techniques to compute the RCS, so we can place some faith in this result.

Although the equivalent surface current model is probably the most computationally efficient available for general problems,[5] the requirement to treat both the equivalent electric and magnetic surface currents doubles the number of unknowns, and hence quadruples the amount of memory, and increases the run-time by between four and eight, depending on the problem size, when compared to a PEC sphere. (Eight is the asymptotic limit, for problems with a very large number of unknowns where the matrix solution dominates the run-time – we discuss

[5] For the dielectric sphere, the Green function is known analytically, so for this special case only, one could develop a faster solver.

Table 6.1 *Comparison of computational requirements for the PEC versus dielectric sphere*

	PEC	Dielectric (surface)		Dielectric (volume)	
		$\lambda_d/8$	$\lambda_d/10$	$\lambda_d/8$	$\lambda_d/10$
N	663	$2 \times 663 = 1326$	$2 \times 1008 = 2016$	$3 \times 772 = 2316$	$3 \times 1370 = 4110$
Memory (Mbyte)	27.6	108	249	329	1033
Relative run-time	1	4.8	11.5	11.5	N/A

N is the number of unknowns. Run-time is per frequency point. The edge lengths are given for $ka = \pi$.

this shortly.) A summary of the computational requirements is given in Table 6.1. The run-times are given normalized to the PEC case. Note how execution time increases by a factor of about five for the dielectric sphere using surface equivalence, and more than ten when using the volumetric mesh. (The N/A indicates that the problem was too large to run with the available resources.)

Also shown in this table is the effect of refining the discretization. Changing the edge length from $\lambda_d/8$ to $\lambda_d/10$, with the corresponding frequency in this case chosen as that corresponding to $ka = \pi$ (towards the upper end of the frequency band), results in an enormous change in computational requirements. Indeed, the $\lambda_d/10$ volumetric discretization was too large for a typical laptop or desktop PC at the time of writing, indicated by N/A in the table.

6.6 Computational implications of surface and volume modelling with the MoM

As has just been seen with the analysis of the sphere, modelling surfaces is far more computationally expensive than modelling wires. As already discussed in Chapter 4, for a typical wire model the number of unknowns N is linearly related to the length of the wire. We will use the product kd to characterize this, with k the wavenumber and d the length of wire. There are two time-consuming operations required by an MoM code with N unknowns, viz. matrix filling and factoring. The former is of $\mathcal{O}(N^2)$, the latter $\mathcal{O}(N^3)$ when using direct solvers (iterative solvers will be discussed later). However, the constants associated with matrix filling can be quite large (that of the matrix solve is close to unity) and in practice one often finds that MoM codes are in the pre-asymptotic region as far as timing goes, spending more time filling than factoring the matrix. Since N is proportional

to kd, we have an asymptotic cost of $\mathcal{O}([kd]^3)$. To store the interaction matrix $[Z]$, N^2 memory locations are required. Hence the amount of memory required is $\mathcal{O}([kd]^2)$ for wires. These properties are also known as the *frequency scaling* behavior of the algorithm.

For a surface, although triangles using the vector RWG basis functions are the approach generally used in practice, when doing a frequency scaling analysis it is easier to consider square patches. To model a surface of size $kd \times kd$, it is clear that the number of unknowns will now be $M = N \times N$; thus the asymptotic computational cost is clearly $\mathcal{O}([kd]^6)$. (The asymptotic analysis neglects the fact that when modelling a *surface*, one needs to approximate the *two* components of current on each patch – so in practice surface modelling is costly. As for the one-dimensional case however, the matrix fill tends to dominate the run-time for many problems, with a somewhat lower asymptotic behavior.) In terms of memory, the requirements are $\mathcal{O}([kd]^4)$ for surfaces.

To give a concrete example, consider doubling the size of ground plane in the helix example discussed in Chapter 5; equivalently, double the frequency – the product kd expresses this product of wavenumber and size succinctly. The run-time will increase by between $2^4 = 16$ and $2^6 = 64$, and the amount of memory required will increase by a factor of $2^4 = 16$. (This is approximate since the helix must also be modelled more finely, but as a wire structure, the frequency scaling is somewhat better; however, the requirements of the ground plane increasingly dominate the considerations.) A factor of 64 is almost precisely the difference between minutes and hours and one should appreciate that modelling surfaces may require powerful computers and take considerable time. Fortunately, there are some methods available to assist in this regard, which we will discuss shortly.

Modelling volumes is even more costly. To model a volume of size $kd \times kd \times kd$, it is clear that the number of unknowns will now be $M = N \times N \times N$; thus the asymptotic computational cost is clearly $\mathcal{O}([kd]^9)$. (Again, the asymptotic analysis neglects the fact that when modelling a volume, one needs to approximate the components of current on each cell – now three of them. On the other hand, once again the matrix fill tends to dominate the run-time for many problems.) In terms of memory, the requirements are $\mathcal{O}([kd]^6)$ for volumes. We saw these effects clearly at work in Table 6.1, where a slight change in edge length for the volumetric case meant that we were unable to solve the problem in a reasonable time, or indeed even run it all due to memory limitations.

6.7 Hybrid MoM/asymptotic techniques for large problems

This section is based on a review paper originally published as [10].

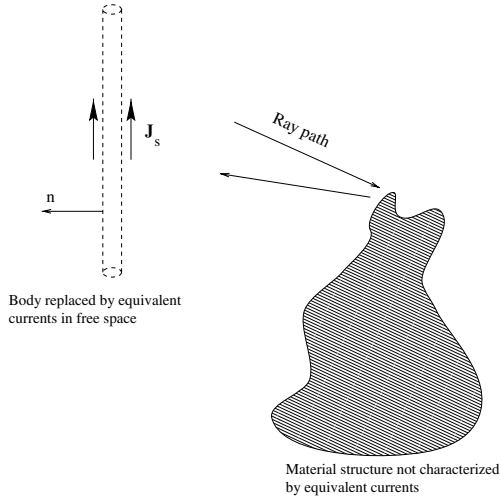


Figure 6.8 Wire radiator together with an electromagnetically large scatterer, whose effect is taken into account using a hybrid formulation. (After [10], ©1999 SAIEE.)

6.7.1 Introduction

Any combination of CEM techniques can be termed a hybrid. Here it is convenient to distinguish between *exact* and *approximate* hybrid approaches. In the former, also known as MoM/Green's function hybrids, special Green's functions are used to take the effect of the scatterer in Fig. 6.8 into account *implicitly*. Although very powerful for appropriate problems, the restricted number of special Green's functions available limits the generality of this approach. In the latter case, high-frequency methods such as physical optics are used to describe approximately the interaction between parts of the structure far removed from one another.

Probably the best known of the exact hybrids is the Sommerfeld potential treatment for radiators near, on or within stratified media. This will be discussed subsequently in this book. For slotted waveguide array analyses, the appropriate waveguide Green function has been widely used in MoM formulations. Another special Green function that has been used is that for layered spheres [11].

Deriving such Green functions is a formidable task: [5] gives a good introduction to the process of deriving a Green function, but for more advanced purposes a detailed description of dyadic Green functions may be found in [12]. A review of this type of hybrid method may be found in [13].

We use the term "exact" hybrid method for this approach since the only approximations made involve the conventional MoM discretization of the current on the radiator/scatterer. There is some disagreement about the use of the term "hybrid" for the MoM/Green function method; we follow the nomenclature of [13]

here. As regards the use of "exact", we have already commented that many special Green's functions involve theoretically infinite series expansions or, as we will see, pose challenging integration problems in the complex plane, as is the case with Sommerfeld potentials.

6.7.2 Moment method/asymptotic hybrids

Hybridizations of the MoM with various asymptotic techniques are approximate in the sense that in addition to the conventional MoM discretization, assumptions are made that are only exact in the high-frequency limit. (The MoM is sometimes described as a "numerically exact" formulation, in that the only approximations are those required to produce a linear system. This type of hybrid is no longer numerically exact – even if the equations could be solved exactly, without any errors introduced by discretization or numerical evaluation of integrals, the method is still approximate.) However, these methods are potentially more generally applicable than the MoM/Green function hybrids outlined above and we will now review physical optics for this purpose.

6.7.3 Physical optics and MoM hybridization

Physical optics (PO) is a well established concept in electromagnetic theory [5, Section 7.10]. The essence is that the equivalent surface current on a smooth conducting surface is given by:

$$\mathbf{J}_s = 2\hat{n} \times \mathbf{H}_i \tag{6.14}$$

We have already seen in Section 6.1 that this is an approximation of the MFIE. It may also be seen as an application of the equivalence principle, with the following approximations for a sufficiently large structure: firstly, $\mathbf{H}$ can be replaced by $2\mathbf{H}_i$ (this essentially assumes no end effects); secondly, currents can be "locally" imaged (hence the factor 2). Note that unlike a ray-based method, integration over the surface current is still required – but the current in the integrand is now *known*, as opposed to the MoM where the current is *unknown* .

In terms of hybridization with the MoM, PO has an enormous advantage in being *current based* – most asymptotic methods (UTD etc.) are *field* based, and this leads to a rather natural MoM/PO hybridization process. The essential idea is to use the MoM on small, resonant structures, and in regions near edges, and to use PO on large, smooth areas. If applied appropriately, smooth "blending" should occur between these regions. The overview in this section closely follows the development presented by Jakobus and Landstorfer [14] and retains their notation.

The mechanics of hybridization require a brief review of basic MoM theory using linear operator notation. The scattered fields are set up by currents on surfaces ($\mathbf{J}^{\mathrm{MM}}$) and wires ($\mathbf{I}^{\mathrm{MM}}$):

$$\begin{aligned}\mathbf{E}_s &= \mathcal{L}_J^E \mathbf{J}^{\mathrm{MM}} + \mathcal{L}_I^E \mathbf{I}^{\mathrm{MM}} \\ \mathbf{H}_s &= \mathcal{L}_J^H \mathbf{J}^{\mathrm{MM}} + \mathcal{L}_I^H \mathbf{I}^{\mathrm{MM}}\end{aligned} \tag{6.15}$$

$\mathcal{L}_J^E$, $\mathcal{L}_J^E$ etc. are linear operator short-hand for the actual integrodifferential operators (for example, the EFIE and the MFIE as in Section 6.1). Standard MoM basis functions are used:

$$\begin{aligned}\mathbf{J}^{\mathrm{MM}} &= \sum_{n=1}^{N_J^{\mathrm{MM}}} \alpha_n \mathbf{f}_n \\ I^{\mathrm{MM}} &= \sum_{n=1}^{N_I^{\mathrm{MM}}} \beta_n g_n\end{aligned} \tag{6.16}$$

Jakobus and Landstorfer use piecewise linear basis functions for g_n and $\mathbf{f}_n$; the latter are the Rao–Wilton–Glisson triangular vector functions for surfaces as already discussed. For a PEC, the standard boundary condition $\mathbf{E}_{\tan} = 0$ is applied, resulting in:

$$\begin{aligned}-\mathbf{E}_{\tan}^i = &\sum_{n=1}^{N_J^{\mathrm{MM}}} \alpha_n (\mathcal{L}_J^E \mathbf{f}_n)_{\tan} + \\ &\sum_{n=1}^{N_I^{\mathrm{MM}}} \beta_n (\mathcal{L}_I^E g_n)_{\tan}\end{aligned} \tag{6.17}$$

Either collocation or weighted residuals can be used to solve for the unknown coefficients α_n and β_n (in total, $N_J^{\mathrm{MM}} + N_I^{\mathrm{MM}}$ of them).

Now, in the region of the scatterer *not* treated by the MoM, the PO surface current is approximated *using the same surface patch treatment as in the MoM region* as

$$\mathbf{J}^{\mathrm{PO}} = \sum_{n=N_J^{\mathrm{MM}}+1}^{N_J^{\mathrm{MM}}+N_J^{\mathrm{PO}}} \gamma_n \mathbf{f}_n \tag{6.18}$$

with $\mathbf{f}_n$ as before and γ_n coefficients of surface current in the PO region. It is *very important* to note that γ_n are known (in terms of the α_n and β_n coefficients) from

the PO approximation, as shown later. Hence they are *not* obtained by the solution of a linear system – thus the matrix *size* remains $N_J^{\mathrm{MM}} + N_I^{\mathrm{MM}}$.

In the PO region, the PO current $\mathbf{J}^{\mathrm{PO}}$ is given by:

$$\begin{aligned}\mathbf{J}(\mathbf{r})^{\mathrm{PO}} &= 2\delta_i \cdot \hat{n} \times \mathbf{H}_i(\mathbf{r}) \\ &\quad + \sum_{n=1}^{N_J^{\mathrm{MM}}} 2\alpha_n \delta_{J,n} \cdot \hat{n} \times \mathcal{L}_J^H \mathbf{f}_n \\ &\quad + \sum_{n=1}^{N_I^{\mathrm{MM}}} 2\beta_n \delta_{I,n} \cdot \hat{n} \times \mathcal{L}_I^H g_n \end{aligned} \tag{6.19}$$

$\delta_{J,n}$ and $\delta_{I,n}$ account for possible shadowing, with values of ± 1 or 0; the optical basis of the method will be recalled.

When currents in the PO region are included as well, the equation from the boundary condition *in the MoM region* becomes:

$$\mathcal{L}_J^E \mathbf{J}^{\mathrm{MM}} + \mathcal{L}_I^E \mathbf{I}^{\mathrm{MM}} + \mathcal{L}_J^E \mathbf{J}^{\mathrm{PO}} = -\mathbf{E}^{i,\tan} \tag{6.20}$$

Note that there are *two different* PO/MoM coupling mechanisms.

1. The currents in the MoM region contribute to the PO currents via Eq. (6.19) (via the summation terms).
2. The currents in the PO region in turn contribute to the fields in the MoM region and thus impact on the boundary condition represented by Eq. (6.20).

It might appear that this would require some iterative process for self-consistency, but the "feedback" effects can be taken into account in closed form. The PO currents can be found *in terms of the unknown MoM currents* as

$$\gamma_k = \tau_{i,k} + \sum_{n=1}^{N_J^{\mathrm{MM}}} \alpha_n \cdot \tau_{J,n,k} + \sum_{n=1}^{N_I^{\mathrm{MM}}} \beta_n \cdot \tau_{I,n,k} \tag{6.21}$$

with

$$\begin{aligned}\tau_{i,k} &= (\hat{t}_k^+ + \hat{t}_k^-) \cdot (\delta_i \hat{n} \times \mathbf{H}_i) \\ \tau_{J,n,k} &= (\hat{t}_k^+ + \hat{t}_k^-) \cdot (\delta_{J,n} \hat{n} \times \mathcal{L}_J^H \mathbf{f}_n) \\ \tau_{I,n,k} &= (\hat{t}_k^+ + \hat{t}_k^-) \cdot (\delta_{I,n} \hat{n} \times \mathcal{L}_I^H g_n) \end{aligned} \tag{6.22}$$

$\hat{t}_k^+$ and $\hat{t}_k^-$ are unit vectors associated with the kth triangle edge; see [14] for further details. It is important to note that *all* the terms in the above equation are *known*, being either derived from the geometry of the problem, the discretization or the chosen basis function.

The central idea here is that these PO currents *in terms of the MoM unknowns* can now be substituted into Eq. (6.20). The final result is the following:

$$\sum_{n=1}^{N_J^{\mathrm{MM}}} \alpha_n \cdot \left[(\mathcal{L}_J^E \mathbf{f}_n)_{\tan} + \sum_{k=N_J^{\mathrm{MM}}+1}^{N_J^{\mathrm{MM}}+N_J^{\mathrm{PO}}} \tau_{J,n,k} \cdot (\mathcal{L}_J^E \mathbf{f}_k)_{\tan} \right] +$$
$$\sum_{n=1}^{N_I^{\mathrm{MM}}} \beta_n \cdot \left[(\mathcal{L}_I^E g_n)_{\tan} + \sum_{k=N_J^{\mathrm{MM}}+1}^{N_J^{\mathrm{MM}}+N_J^{\mathrm{PO}}} \tau_{I,n,k} \cdot (\mathcal{L}_J^E \mathbf{f}_k)_{\tan} \right]$$
$$= -\mathbf{E}_{i,\tan} - \sum_{k=N_J^{\mathrm{MM}}+1}^{N_J^{\mathrm{MM}}+N_J^{\mathrm{PO}}} \tau_{i,k} \cdot (\mathcal{L}_J^E \mathbf{f}_k)_{\tan} \tag{6.23}$$

The above equation summarizes the MoM/PO interaction: the effect of the PO is to *alter* the MoM matrix entries. Note that each MoM entry is modified by contributions from *all* the PO currents; this can become computationally expensive and can be neglected under certain conditions, usually when the PO and MoM regions are physically separated. (An example is a reflector antenna, where the feed is treated with the MoM and the reflector with the PO.) Note further that the boundary condition of zero tangential **E** is only rigorously enforced in the MoM region.

In the basic MoM/PO hybridization outlined above, edge effects are *not* taken into account by the PO. It is possible to use Fock theory to account for these effects; see for example [14]. The approach used is related to Umfitsev's physical theory of diffraction.

For very large structures, the integration over the entire structure can still become very time consuming – although the $\mathcal{O}(f^6)$ dependence of the MoM is reduced enormously, the PO asymptotic dependence is still $\mathcal{O}(f^2)$.

Hodges and Rahmat-Samii have shown recently that the MoM/PO hybrid can be seen as a special case of a more general EFIE/MFIE hybridization, with the MoM/PO as the first term in an iterative Neumann series technique [15]. They show good results for two monopoles mounted on opposite sides of a cylinder, and thus in each other's shadow region. However, the use of the MFIE restricts the method to smooth closed bodies.

6.7.4 A FEKO example using the MoM/PO hybrid

The above theory is available within FEKO, and we will now consider an example of its applications. For this example, one of the simplest (and also most effective)

applications of the MoM/PO hybrid will be chosen. We will mount a $\lambda/2$ dipole antenna horizontally above a finite ground plane, of $1\lambda \times 1\lambda$ in size. From basic image theory, the "image" in the ground plane is out of phase, so the distance above the ground plane should be an odd multiple of $\lambda/4$ above the ground plane to produce constructive interference.

Modelling hints – symmetry

Once again, symmetry can be exploited to build the model and improve the computational efficiency. By mounting and feeding the dipole symmetrically about the $y = 0$ and $z = 0$ planes, magnetic and electric symmetry can be used. Note that the quarter-ground plane is imaged first in the $y = 0$ plane before the half-dipole is added; one does not want to image a wire on top of itself! Following this, the half-ground plane and half-dipole are then imaged in the $z = 0$ plane to create the whole model, and the feed segment is then added.

Two approaches have been used to solve this problem: firstly, the MoM has been used for the entire problem; then, the MoM is applied to the dipole only, and the effect of the reflector is approximated using the PO. The FEKO models for both are shown in Fig. 6.9 – note that the models appear identical, since it is the mathematical approach, rather than the geometrical model, which differs. Results comparing the far-field H-plane ($z = 0$) radiation patterns computed using the two approaches are shown in Fig. 6.10.

The results shown in Fig. 6.10 compare favorably. Using some advanced methods within FEKO which correct the PO currents at the edge of the reflector, it is possible to do even better. However, a caution is in order. It must be appreciated that the MoM/PO hybrid is *approximate*; how good the approximation is relies quite heavily on the experience of the user. As such, it is useful to build confidence by initially comparing results using MoM/PO hybrids with full MoM solutions as far as possible. Efficient use of symmetry usually allows the solution of quite electrically large MoM problems, although these may of course take some time to compute. Once one is reasonably confident of the level of accuracy for a particular class of applications, one may then do production runs investigating changes to and optimization of the structure, etc. It would, however, be very unwise to base major design decisions on an MoM/PO hybrid solution which one has not carefully evaluated beforehand. (Of course, this is true in general of computed solutions, but even more so in this case.) Problems which generally lend themselves very well to the MoM/PO hybrid approach are reflector-type problems, since the radiating feed element is largely decoupled from the reflector, and ray-tracing issues are not problematic.

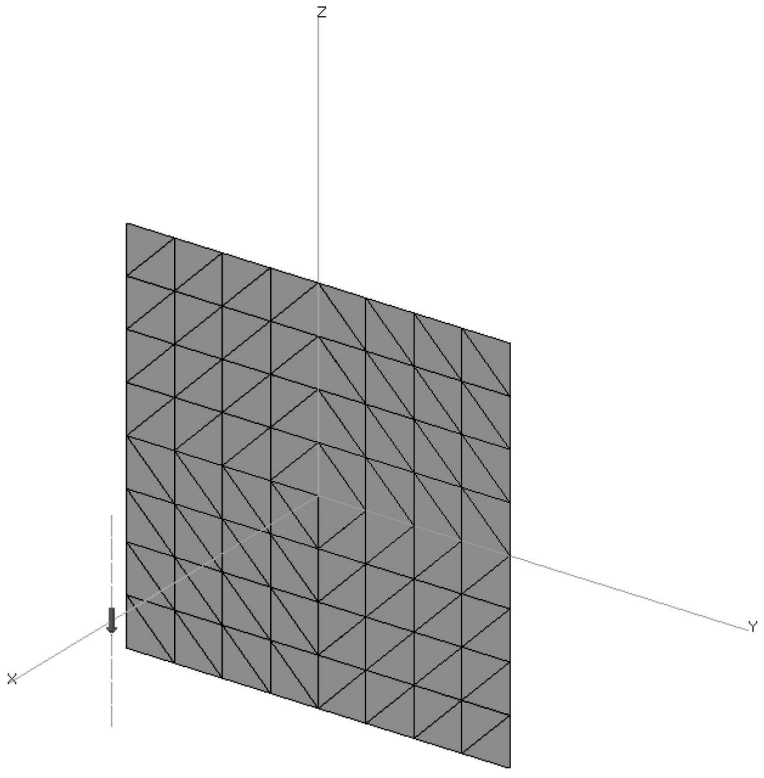


Figure 6.9 FEKO model of a dipole in front of a reflector.

Modelling hints – using the MoM/PO hybrid within FEKO

FEKO is the only commercial code offering this functionality at the time of writing, so the following discussion only applies to FEKO. Physical optics is controlled using the `PO` card, which offers a number of parameters which require some brief discussion. The first parameter, requiring a label, is obvious; the PO is applied to the structures with this label. The second parameter controls *ray tracing*. Because the PO is an optics-based method, in general one needs to ray trace to determine whether a triangle is in the "lit" or "shadow" region relative to the source. In this case, it is clear that all triangles are illuminated, and ray tracing may be switched off to save time. The third parameter relates to the use of symmetry in ray tracing and is irrelevant here since ray tracing has been deactivated. The fourth parameter controls MoM–PO coupling, as described in the previous theoretical section; here, the full treatment is applied and the regions are fully coupled. The fifth parameter is another optics based one; it determines

the number of multiple reflections to be taken into account. In this case, none are required. The final parameter is for specialized use and the default should be used here.

FEKO offers additional functionality to improve PO modelling. The `KA` card permits one to define the boundary of the PO region, and "fringe wave" currents are then used in this region to improve the approximation. The `VS` card allows one to specify "visibility" information, to reduce the time required when multiple reflections are present. The `FO` card uses Fock theory to improve the PO surface current.

6.8 Other approaches for the solution of electromagnetically large problems

6.8.1 Background

By the late 1980s, research on the MoM was confronted with the basic problem of the high asymptotic cost of the method – $\mathcal{O}(N^3)$ in terms of number of unknowns, or $\mathcal{O}([kd]^6)$ for surfaces, as we have seen for direct solvers. Little can be done to improve this further, apart from the application of high-performance computing (of

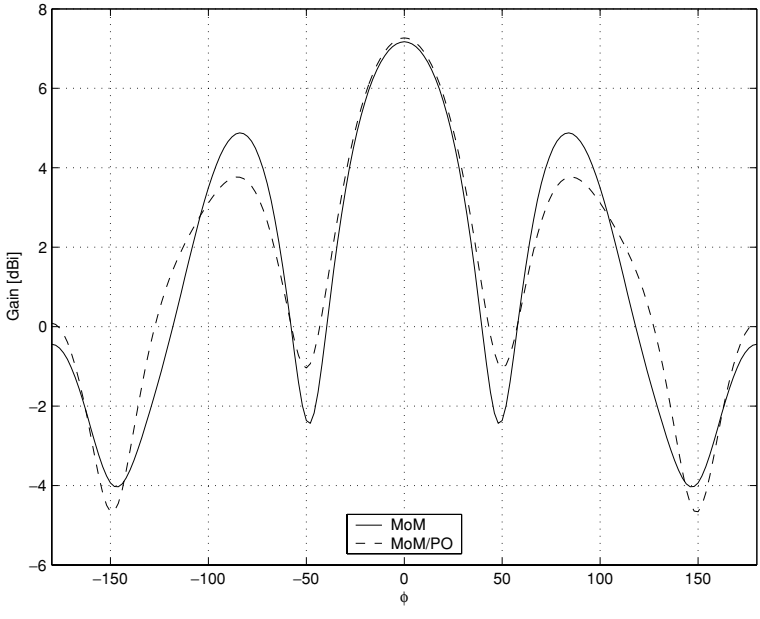


Figure 6.10 A comparison of the H-plane far-field patterns computed using the MoM and MoM/PO hybrid.

which more anon). Iterative solvers started attracting much attention in CEM the late 1980s – even though the basic algorithms, in particular the conjugate gradient (CG) algorithm, have been known since the 1940s – since the computational cost is $\mathcal{O}(N^2)$ per iteration, with overall cost $\mathcal{O}(N_{\text{iter}}N^2)$ for N_{iter} iterations. Clearly, if N_{iter} can be kept well below N, algorithms with better scaling properties are possible. It has to be said here that unfortunately, the considerable experience accumulated by many researchers over the years has indicated that it is very difficult to predict N_{iter} for arbitrary problems; testing the algorithms on canonical problems, such as spheres, has frequently resulted in highly over-optimistic predictions. (The reason is the relatively simple eigenvalue structure of such problems; since the iterative methods usually used variants of the CG method, the rate of convergence is heavily determined by the eigenvalue spectrum.) So, using iterative techniques alone is not sufficient – and in any case, this does nothing to the $\mathcal{O}(N^2)$ memory requirements of the method, which is frequently as serious a problem as computational cost.

From a slightly different perspective, the integral equation formulations which we have worked with are essentially *convolutions* of the Green function with the currents. Familiarity with signal processing methods immediately suggests that convolution in one domain may be more easily implemented by multiplication in the Fourier transform domain; we will exploit this idea in Chapter 7, although for a slightly different purpose. But for now, the idea that one could use a Fourier transform immediately suggests the use of the *fast Fourier transform* (FFT), and indeed, this was one of the first successful "fast" methods in electromagnetics. However, it was limited in terms of application to general structures with arbitrary meshes. An extension of this concept, the adaptive integral method, removes this restriction. However, it is an alternative approach, the fast multipole method (FMM), which provided the theoretical breakthrough in the early 1990s. In its most powerful multi-level form it reduced the asymptotic cost from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$, and it is the most popular of the fast methods today. It was a breakthrough as significant as Berenger's PML absorber,[6] although the theory is rather more complex, and efficient implementation in particular is challenging. (By comparison, the PML is really quite straightforward to code.) Despite the complexity of the theory underlying the FMM, since it is starting to be offered by commercial codes at the time of writing,[7] an elementary introduction is certainly appropriate at this stage. Before looking at fast techniques, however, we will briefly discuss

[6] Hopefully, this comparison will not cause confusion: the PML and FMM are entirely different methods, with quite different aims.

[7] FEKO appears to have been the first publicly available commercial code to incorporate the FMM; the frequently referenced Fast Illinois Solver Code (FISC) has numerous restrictions on its distribution, especially outside the USA, due to US military funding during its development.

high-performance computing, which is also an important topic when the solution of large electromagnetic problems is considered.

6.8.2 High-performance computing

All the methods and technologies described in this section had their genesis in the late 1980s. One approach to the problem of high computational cost, and one which is still bearing fruit today, was exploitation of the emerging technology of parallel processing. Parallel processing – or indeed high-performance computing (HPC) in general – simply provides more computational power, it does not address the fundamental algorithmic issue of computational cost, but can significantly push the envelope of any particular computational technique. At its heart, there are only two ways of making a given computation faster: either increase the rate at which a computer can process information, or do more operations at the same time. The former of course has been the dominant technological drive through several decades, manifested by clock speeds which, for typical personal computers, have increased from some tens of MHz at the start of the 1990s to some GHz by the millennium, only a decade later. The latter has spawned a variety of methods; historically, parallel processing originally split into *pipelining* and *replication*.

Pipelining involves overlapping parts of operations in time and was the approach taken by the vector supercomputers, such as the early CRAY machines (the first of which was installed in 1976). Replication provides more than one functional unit (e.g. CPU), permitting operations to be performed simultaneously, and was the competing approach taken by large processing arrays. Another nomenclature encountered in the earlier literature was single instruction multiple data (SIMD) and multiple instruction multiple data (MIMD) machines. This taxonomy was was introduced by in the early 1970s [16]; a MIMD system described a computer consisting of a number of nodes, each with at least a processing element, operating independently on its own local instruction stream and data, whereas a SIMD system performed the same operation in lockstep to all data. Machines were also characterized in terms of how data were exchanged; many of the early experimental systems used were local memory, message passing systems. In these, all memory was divided up locally amongst the available processors, and a processor could only directly access its own memory. Access to the memory on other processors was done by explicit message passing, which was *much* slower than direct memory access. However, the problem of memory contention that complicated the other main competing approach to memory allocation, namely global memory, was removed with this approach. Technological advances have however blurred many of the traditional distinctions. Even the ubiquitous CPUs encountered in personal computers contain significant elements of pipelining and replication,

and increasingly sophisticated architectures now blur the global/local memory dichotomy.

The basic concept of parallel processing was, and still remains, to provide P processors or processing elements, and by splitting the computational load, reduce the overall run-time by a factor as close to P as possible. Several methods have been proposed to characterize parallel computers, but the most widely used are speed-up and efficiency. Speed-up, S, is the ratio of time taken by an equivalent serial algorithm running on one processor, T_s, to the time taken by the parallel algorithm using P processors, T_p. Efficiency, ϵ, is the speed-up normalized by the number of processors. Formally,

$$S = \frac{T_s}{T_p} \tag{6.24}$$

$$\epsilon = \frac{S}{P} \tag{6.25}$$

S is usually bounded from above[8] by P, and ϵ is hence usually bounded from above by 1.

Some algorithms can be parallelized very easily and efficiently: examples are the FDTD and iterative methods. Some, such as LU decomposition, are rather less obvious, but can nonetheless be very efficiently parallelized with some clever data decomposition techniques. All the major algorithms in CEM have been parallelized with varying degrees of success over the last decade; perhaps the most problematic one has been the FEM, due to the large, unstructured, but highly sparse matrix characterizing the method. Examples of measured efficiencies on a transputer array are shown in Fig. 6.11. (The results are shown for slightly different numbers of processors; this was due to different interconnection topologies used for the algorithms.) These data were measured in the early 1990s, hence the problem sizes are small by contemporary standards, but nonetheless, establish the principle.

An historical aside – the transputer

In the late 1980s, PCs were limited by the 640 kB limitation on RAM imposed by the then dominant operating system, DOS, and clock speeds were low. Supercomputers were (and for that matter still are) extremely expensive. A British company, INMOS, introduced the transputer, one of the first "computers on a chip," incorporating a CPU, floating point unit, memory and communication links. (This was to become quite standard later, but at the time was

[8] Sometimes, architectural quirks resulted in "superlinear" improvement on specific problems, i.e. a speed-up in excess of P; usually, this was a result of the cache design.

revolutionary.) The transputer came in several different variants – the T800 model was the one widely used in parallel processing.

The transputer was a 32-bit RISC[a] design, capable of internal operation at up to 30 MHz – again, this must be seen from the viewpoint of the technology of the time! One T800 transputer was able to produce a peak floating point throughput of 1.5 Mflop/s. A novel feature, still not widely seen on other systems to this day, was the provision of four serial links providing comparatively high-speed communication either with a host processor or with other transputers. Additionally, *all* components could execute concurrently; each of the four links and the floating point processor could perform useful work while the other elements were executing other instructions.

The transputer was a very powerful processor in its own right when introduced, out-performing the microVax, which was then the usual system of choice for numeric computations in universities and most research laboratories (outside US government research laboratories). However, it was ideally suited for application in parallel processing applications, in particular due to the on-chip links, and a number of experimental prototypes and some commercial products incorporating transputers were produced around the world.

The relentless advance of clock speeds in personal computer CPUs during the 1990s, combined with an over-dependence on a novel but ultimately commercially unsuccessful language-cum-operating system, Occam, eventually consigned the transputer to historical notes such as this. However, its role as an innovative catalyst in affordable parallel processing should not be underestimated; its do-it-yourself bargain-basement philosophy, if not technology, inspired a generation of computational scientists working at institutions unable to afford the extremely expensive supercomputers of the time, and still resonates today in current systems using Linux clusters.

[a] Reduced instruction set computer.

In this context, it is necessary to mention Amdahl's "law,"[9] which states that if an algorithm contains both a serial and a parallel part, the relative time taken by the serial part increases as parallelization reduces that of the parallel part, and a law of diminishing returns holds: further parallelization has increasingly little influence on run-time. While this observation is perfectly true, for many problems the ultimate aim is to increase the problem size that can be handled. Thus as more parallelization is made available, larger problems are tackled and the overall

[9] As with Moore's "law" – that the number of transistors in integrated circuits doubles approximately every 18 months – this is really an observation rather than a law in the sense as used in physics.

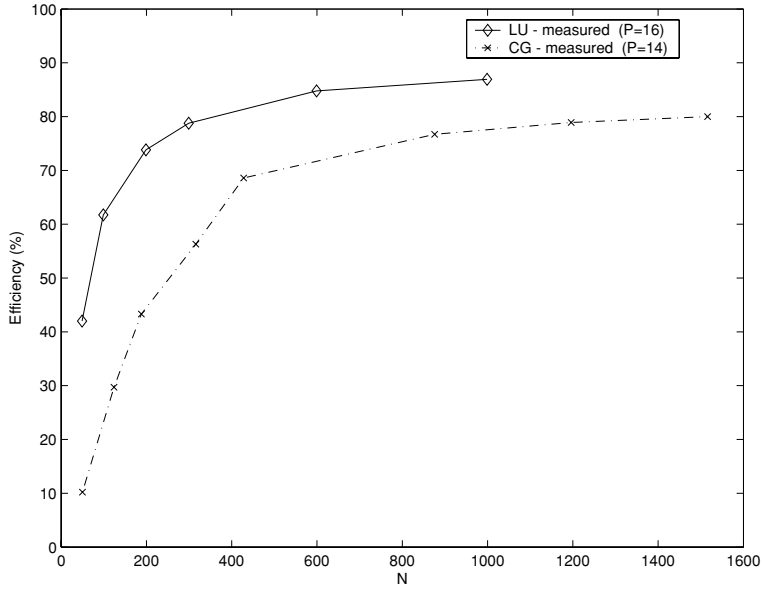


Figure 6.11 Comparison of measured efficiencies of parallel CG and LU algorithms on a transputer array, for an MoM problem with a total of N unknowns running on P processors. (Adapted from [17, Fig. 18] and [18, Fig. 12]).

serial/parallel split remains fairly constant. In particular, the efficiency of many parallel algorithms is a function of grain size – the number of unknowns per processor, N/P. An example of this is shown in Fig. 6.12, which indicates that for a particular grain size, the algorithm has approximately constant efficiency.

When HPC first came to the attention of the CEM community, it was often accompanied by highly specialized hardware, frequently purpose built, such as the transputer-based arrays mentioned here. However, relatively mainstream environments are now the norm, reflecting a degree of maturity in the field. It is also notable that the old SIMD-MIMD classification has largely fallen away – HPC environments now are generally classified either as SMP (symmetric multiprocessor), MPP (massively parallel processor) or distributed processing environments. The first is currently epitomized by systems such as the Silicon Graphics Origin; the number of processors is typically fairly modest, but memory is essentially shared. The second is epitomized by the Cray T3-E, with a large number of processors accessing distributed memory, and the last by heterogeneous networks of standard workstations, again with distributed memory but much slower communication networks than the purpose-built ones incorporated into MPPs. (The T3-E actually combines elements of both SMP and MPP paradigms, since it also contains a globally addressable memory subsystem.) At the time of writing, yet

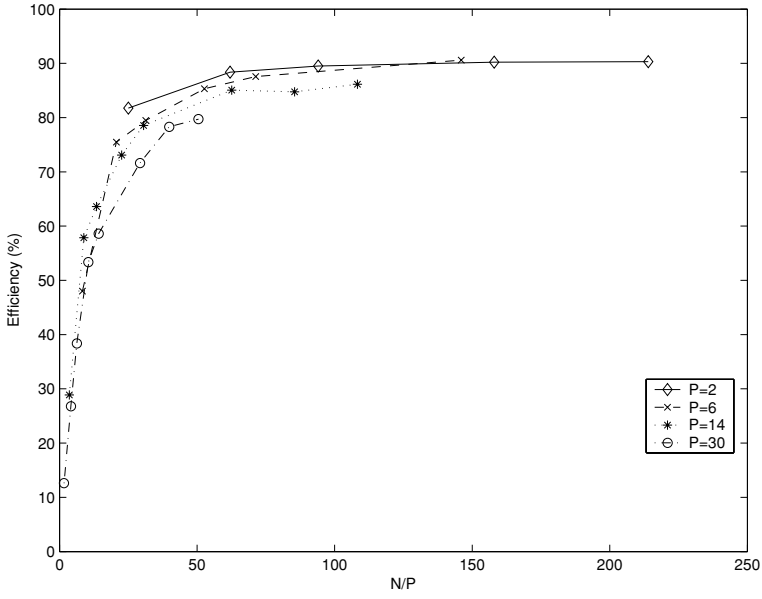


Figure 6.12 Measured efficiency of a parallel CG algorithm on a transputer array, for an MoM problem with a total of N unknowns running on P processors. (After [17, Fig. 7], ©1993 ACES, reprinted with permission.)

another new paradigm, "grid computing," is emerging, with the aim of using the Internet as a global computer.

During the 1990s, there was also a major shake-out in the HPC sector; a number of the machines (and manufacturers) referenced in papers at that time have long ceased trading. Thinking Machines Corp. and their Connection Machines (CM-2 and CM-5), which were some of the few truly deserving the *massively* parallel tag, with thousands of SIMD processors, are gone. Kendall Square Research, whose machines had some innovative features, not least a physically distributed memory which was accessed as shared memory by application programs, using a system of multi-level caches, has also long ceased to function commercially. Cray Inc. and Silicon Graphics remain arguably the most influential commercial vendors in this field at the time of writing.

A noteworthy aspect of the work reported in the literature on parallel processing is that no new specifically parallel algorithms have arisen in computational electromagnetics. Well over a decade back, when parallel computing first attracted serious interest, there was speculation in some quarters that the rise of massively parallel computers would trigger entirely new algorithms that were only feasible in massively parallel computing environments. With hindsight, such claims appear as primarily marketing "hype." Additionally, it has to be commented that at the

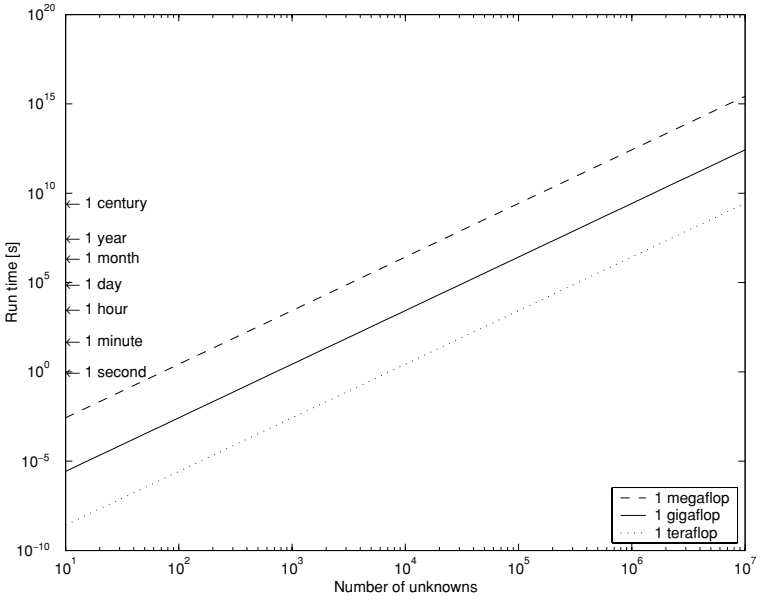


Figure 6.13 Run-times for LU decomposition, compared for systems capable of sustaining 1 megaflop, 1 gigaflop and 1 teraflop.

time of writing, the usability of current high-performance computing platforms remains disappointing: a major bottleneck has been the inadequacy of parallel compilers and system software. Although considerably improved over the systems of a decade ago (where such system software was sometimes entirely absent), fundamental items such as parallel I/O and easy-to-use parallel debuggers have not appeared. What is encouraging has been the emergence of two standardized "harnesses" – parallel virtual machine (PVM) and message passing interface (MPI). These provide standardized high-level communication routines (via libraries) to route data between processes, removing, or at least greatly reducing, the hardware dependent implementations which characterized earlier work.

Nonetheless, despite implementation issues which remain challenging, parallel processing has emerged as a very useful enabling technology; several commercial codes (such as FEKO) are available in parallelized versions for various platforms. Whilst one does not always appreciate the impact of incremental increases in performance, when compounded over decades the results are deeply impressive. In Fig. 6.13, the time required for direct matrix solution (LU decomposition) on systems capable of sustaining 1 megaflop, 1 gigaflop and 1 teraflop respectively are compared.[10] Comparing a 1 megaflop (typical of the late 1980s) and a 1 gigaflop

[10] The operation count for LU decomposition for a matrix of dimension N with complex valued entries is approximately $8/3N^3$ floating point operations.

machine (typical of current systems), one sees that for a problem with around 1000 unknowns, the time has dropped from around an hour to a few seconds. A similar improvement is noted for a 10 000 unknown problem when comparing a 1 gigaflop and a 1 teraflop machine.[11]

6.8.3 FFT-based methods

If we refer back to the very simple introductory thin-wire example of Chapter 4, specifically to Eqs. (4.15) and (4.16), we note that Z_{mn} is a function of only $m - n$ and Δ. With a uniform discretization, as used there, the latter is constant, and hence we actually only need to compute one row of the matrix. This is known as *Toeplitz* (or translational) symmetry. The reason that this observation is important is that in this case, the product of this matrix with a vector can be implemented as a discrete convolution.

In general, a discrete convolution is an operation of the form

$$e_m = \sum_{n=0}^{N-1} j_n g_{m-n} \tag{6.26}$$

or in matrix form

$$\begin{bmatrix} g_0 & g_{-1} & g_{-2} & \cdots & g_{1-N} \\ g_1 & g_0 & g_{-1} & \ddots & \\ g_2 & g_1 & g_0 & \ddots & \\ \vdots & & & & \\ g_{N-1} & g_{N-2} & g_{N-3} & \cdots & g_0 \end{bmatrix} \begin{bmatrix} j_0 \\ j_1 \\ \vdots \\ j_{N-1} \end{bmatrix} = \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{N-1} \end{bmatrix} \tag{6.27}$$

The $N \times N$ matrix in the above is a general *Toeplitz* matrix; all the elements of this matrix are described by the $2N - 1$ entries in the first row and column. If the elements repeat with period N, so that

$$g_{n-N} = g_n, \qquad n = 1, 2, \ldots, N-1 \tag{6.28}$$

then the operation is known as a *circular* discrete convolution, and the $N \times N$ matrix above is *circulant*. Otherwise, the operation is a *linear* discrete convolution. Any linear discrete convolution of length N can be embedded into a circular discrete convolution of length $2N - 1$ by extending the original sequence g to repeat

[11] In November 1998, a CRAY T3-E became the first supercomputer to sustain the latter rate of computation on a real-world computation.

with period $2N-1$, zero padding the sequence to length $2N-1$ and changing the upper limit of summation in Eq. (6.26) to $2N-2$.

The discrete convolution theorem states that if Eq. (6.26) is a circular discrete convolution, it is equivalent to

$$\tilde{e}_n = \tilde{j}_n \tilde{g}_n, \qquad n = 0, 1, \ldots, N-1 \tag{6.29}$$

where the $\tilde{e}$ is the N-point discrete Fourier transform (DFT) of e, and similarly $\tilde{j}_n$ and $\tilde{g}_n$. The DFT will of course be implemented using the FFT algorithm. If Eq. (6.26) is a linear discrete convolution, then embedding as described above is used.

Hence, the matrix-vector product of Eq. (6.27) can be efficiently implemented as

$$e = \mathrm{FFT}_N^{-1} \{\mathrm{FFT}_N(j)\mathrm{FFT}_N(g)\} \tag{6.30}$$

In the MoM context, with a Toeplitz matrix, the matrix-vector product is thus expressed as

$$\sum_{i=1}^{n} Z_{mn} I_n = Z_m \otimes I_m \tag{6.31}$$

where $Z_m = Z_{m1}$ and $\otimes$ indicates cyclic convolution, evaluated as

$$[Z]\{I\} = \mathrm{FFT}_N^{-1} \{\mathrm{FFT}_N(Z_m)\mathrm{FFT}_N(I)\} \tag{6.32}$$

Usually, $\{I\}$ is an approximation of the current, typically $\{I\}_k$ at the kth iteration of an iterative solver.

Note that the convolution has become the Hadamard, or outer, product (i.e. element-by-element) and hence for an iterative algorithm, the $\mathcal{O}(N^2)$ cost of the matrix-vector product (usually required once or twice per iteration) has been reduced to $\mathcal{O}(N \log N)$. Also very importantly, the memory requirement is reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$.

This can of course be extended to two and three dimensions, using two- and three-dimensional FFTs as appropriate; the requirement remains that the grid should be a regular Cartesian one. Indeed, three-dimensional FFT-based methods provide quite efficient ways of dealing with the volume integral MoM discretizations.

The adaptive integral method is an extension of this idea to triangular surface grids. In this case, the triangular subdomain basis functions are projected onto a rectangular grid so that the FFT can be applied for the matrix-vector product.

A mathematical aside – what makes the fast Fourier transform (FFT) fast?

The FFT must rate as one of the top numerical algorithms of the twentieth century. Although first popularized by J. W. Cooley and J. W. Turkey in the mid 1960s, perhaps as many as a dozen individuals had independently discovered, and in some cases implemented, efficient methods for evaluating the discrete Fourier transform (DFT), starting with no less a figure than Gauss in 1805. As usual, the treatment in [8] is both highly entertaining and informative, and the following is a summary thereof.

Firstly, until the mid 1960s, the standard method for evaluating an N-point DFT of the discrete function h_k,

$$H_n \equiv \sum_{k=0}^{N-1} h_k \, e^{2\pi ikn/N} \tag{6.33}$$

was to define the complex number W as (note that $i = \sqrt{-1}$, the unit imaginary number, not a counter!)

$$W \equiv e^{2\pi i/N} \tag{6.34}$$

and then the DFT can be written as

$$H_n = \sum_{k=0}^{N-1} W^{nk} h_k, \qquad n = 0, 1, \ldots, N-1 \tag{6.35}$$

Clearly, for each n, this is the product of a matrix of size $N \times N$ (whose (n, k)th entry is W to the power of $n \times k$) times a vector of length N; this must be done N times (for each value of n) yielding an $\mathcal{O}(N^2)$ algorithm.

One of the "rediscoveries" of the algorithm which provides one of the clearest derivations of the FFT is that of Danielson and Lanczos in 1942. The DFT is written as the sum of two DFTs, each of length $N/2$. One is formed from the even-numbered points, one from the odd-numbered points. Mathematically,

$$\begin{aligned} F_k &= \sum_{j=0}^{N-1} e^{2\pi ijk/N} f_j \\ &= \sum_{j=0}^{N/2-1} e^{2\pi ik(2j)/N} f_{2j} + \sum_{j=0}^{N/2-1} e^{2\pi ik(2j+1)/N} f_{2j+1} \end{aligned}$$

$$= \sum_{j=0}^{N/2-1} e^{2\pi ikj/(N/2)} f_{2j} + W^k \sum_{j=0}^{N/2-1} e^{2\pi ikj/(N/2)} f_{2j+1}$$

$$= F_k^e + W^k F_k^o \tag{6.36}$$

F_k^e is the kth component of the Fourier transform of length $N/2$ formed from the even components of the original f_j, and similarly F_k^o is the corresponding transform formed from the odd components. Although in the last line of Eq. (6.36), k varies from 0 to $N-1$, not just $N/2-1$, the transforms F_k^e and F_k^o are periodic in k with length $N/2$, so each is simply repeated through two cycles.[a]

The neat point about this algorithm is that is can be applied *recursively*. For instance, F_k^e can now be subdivided in F_k^{ee} and F_k^{eo}. For N a power of two, this can be continued down to the point where one is left with the transform of length one – which simply copies the input to the output. There are $\log_2 N$ such recursions. These one-point transforms are then combined appropriately. Each such combination takes of order N operations, there are $\log_2 N$ such combinations, hence we have the $\mathcal{O}(N \log_2 N)$ operation count of the FFT.

The above is not a complete description of the algorithm; one still needs to perform some book-keeping to keep track of which one-point transform corresponds to which combination of even–odd subdivisions, e.g. F^{eoe} for an eight-point transform. By *bit-reversing* the binary representation of each index of the input vector, it turns out that this can be done very efficiently. The interested reader can refer to [8, Section 12.2] for the details.

[a] Another way of looking at this is that taking even-numbered points is equivalent to halving the sampling density, hence the aliasing frequency also halves.

6.8.4 The fast multipole method

A two-dimensional FMM prototype

Whereas the FFT-based methods rely on the algebraic properties of the DFT, the fast multipole method (FMM) is based on the analytical properties of the Green function. Before we briefly introduce the full FMM, it is worth discussing a two-dimensional example originally developed by Lu and Chew, which captures the essence of the algorithm in a far more readily accessible form; it is presented in the following form in [19, Section 4.13]

Assume a TM_z PEC scattering problem. In this case, the EFIE is [19, Section 2.1]

$$E_z^{\mathrm{inc}}(t) = jk\eta A_z(t) \tag{6.37}$$

where

$$A_z(t) = \frac{1}{4j} \int J_z(t') H_0^{(2)}(kR)\, dt'$$

$$R = \sqrt{[x(t) - x(t')]^2 + [y(t) - y(t')]^2} \tag{6.38}$$

with t a parametric variable describing position around the contour of the cylinder surface, and the unprimed and primed coordinates indicate source and field points as usual. Using subsectional pulse basis functions, as in Chapter 4, one obtains the usual MoM matrix equation, with impedance matrix entries which, for segments small compared to a wavelength, may be approximated by

$$Z_{mn} \approx \frac{k\eta}{4} w_n H_0^{(2)}(kR_{mn})\, dt', \qquad \forall\, m \neq n \tag{6.39}$$

with w_n the width of cell n and

$$R_{mn} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2} \tag{6.40}$$

This, then, is the conventional MoM solution of this problem. We will assume that there are no geometrical properties of the shape of the circumference that we can exploit. (For instance, if it is a right circular cylinder, and the discretization is uniform, we have a Toeplitz matrix and we can apply the FFT approach to reduce the cost.) If we seek the solution of $[Z]\{I\} = \{V\}$ using a conventional iterative solver, the cost per iteration will be $\mathcal{O}(N^2)$. The memory requirement is also $\mathcal{O}(N^2)$.

Now, consider a fast approach for computing the product of the matrix-vector product. As usual, the circumference of the cylinder will be divided into N segments (which need *not* be equal in size in this approach). Now, the new idea: we collect these segments into p *groups* [12] of roughly equal size and number of unknowns. We index the groups as $i = 1, 2, \ldots, p$; there are now N/p segments per group, indexed as $n = 0, 1, \ldots, N/p - 1$ in each group. One segment per group will be centered at a local origin (x_{i0}, y_{i0}), whilst the other segment centroids are denoted by (x_{in}, y_{in}). For source and field cells closely located, the "near-zone," the calculation proceeds as usual. However, for other segments, sufficiently far separated that they are in the "far-zone," an approximation will be used as follows.

[12] In the presentation of [19, Section 4.13], the terms "cells" and "segments" are used respectively. The latter is rather confusing, since a segment in an MoM formulation is usually the sub-domain spanned by one (or sometimes a few) basis functions. The nomenclature used in this section corresponds to typical FMM usage.

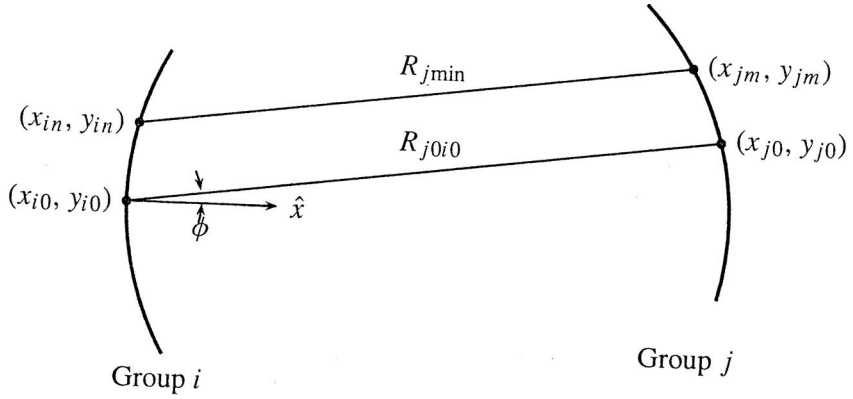


Figure 6.14 Groups i and j, with segments in and jm

Consider the calculation of the field at (x_{jm}, y_{jm}) due to sources on group i (see Fig. 6.14)

$$E_z^{\text{scat}}(x_{jm}, y_{jm}) = -\frac{\omega\mu}{4} \sum_{n=0}^{N/p-1} j_n w_n H_0^{(2)}(kR_{j\min}) \tag{6.41}$$

The distance function $R_{j\min}$ is approximated as

$$R_{j\min} \approx R_{j0i0} + R_{jm} - R_{in} \tag{6.42}$$

where

$$R_{j0i0} = \sqrt{(x_{j0} - x_{i0})^2 + (y_{j0} - y_{i0})^2} \tag{6.43}$$

$$R_{jm} = (x_{jm} - x_{j0}) \cos\phi + (y_{jm} - y_{j0}) \sin\phi \tag{6.44}$$

$$R_{in} = (x_{in} - x_{i0}) \cos\phi + (y_{in} - y_{i0}) \sin\phi \tag{6.45}$$

The angle ϕ denotes the orientation of R_{j0i0} with respect to the x-axis. (This is just the usual far-field approximation used in the derivation of the potential of a two-dimensional dipole.)

Now, the asymptotic form of the Hankel function for large arguments,

$$H_0^{(2)}(k\rho) = \sqrt{\frac{2j}{\pi k\rho}} e^{-jk\rho} \tag{6.46}$$

is applied, yielding

$$H_0^{(2)}(kR_{j\min}) \approx H_0^{(2)}(kR_{j0j0}) e^{-jkR_{jm}} e^{+jkR_{in}} \tag{6.47}$$

and thus Eq. (6.41) can be replaced by

$$E_z^{\text{scat}}(x_{jm}, y_{jm}) \approx -\frac{\omega\mu}{4} H_0^{(2)}(kR_{j0i0}) e^{-jkR_{jm}} \sum_{n=0}^{N/p-1} j_n w_n e^{+jkR_{in}} \tag{6.48}$$

Hence, *all* interactions between the cells in groups i and j can be obtained from a *single* summation over the coefficients j_n, and one Hankel function calculation. This involves $\mathcal{O}(N/p)$ operations. There are approximately p^2 combinations of far-zone groups, so the overall complexity grows as $\mathcal{O}(Np)$. It can be shown that the optimal grouping is $p = \sqrt{N}$, in which case the complexity is $\mathcal{O}(N^{3/2})$.

It is useful to separate the operations contained in Eq. (6.48). First, the sources on group i are *aggregated* together via the summation

$$S^i \approx \sum_{n=0}^{N/p-1} j_n w_n e^{+jkR_{in}} \tag{6.49}$$

Then, *translation* uses the Hankel function

$$E_z^{\text{scat}}(x_{j0}, y_{j0}) \approx -\frac{\omega\mu}{4} H_0^{(2)}(kR_{j0i0}) S_i \tag{6.50}$$

to shift the field to the center of group j. Finally, the scattered field is *disaggregated* throughout group j by a multiplication with the phase correction

$$E_z^{\text{scat}}(x_{jm}, y_{jn}) \approx e^{-jkR_{jm}} E_z^{\text{scat}}(x_{j0}, y_{j0}) \tag{6.51}$$

We find analogous steps in the full FMM.

The full three-dimensional FMM

The FMM rests on two identities. The first, a form of Gegenbauer's addition theorem, states that

$$\frac{e^{-jk_0|\vec{r}+\vec{d}|}}{|\vec{r}+\vec{d}|} = -jk_0 \sum_{l=0}^{\infty} (-1)^l (2l+1) j_l(k_0 d) h_l^{(2)}(k_0 r) P_l(\hat{d} \cdot \hat{r}) \tag{6.52}$$

where $j_l(x)$ is a spherical Bessel function of the first kind, $h_l^{(2)}(x)$ is a spherical Hankel function of the second kind, $P_l(x)$ is a Legendre polynomial, and $d < r$. All the special functions are as defined in standard texts, e.g. [20]. The second identity is a *spectral decomposition* of the product of the Bessel function and the Legendre polynomial, into propagating plane waves:

$$4\pi(-j)^l j_l(k_0 d) P_l(\hat{d} \cdot \hat{d}) = \oint_S e^{-j\vec{k}\cdot\vec{d}} P_l(\hat{d} \cdot \hat{r})\, d^2\hat{k} \tag{6.53}$$

where the integral is over a unit sphere S and $\vec{k} = k_0\hat{k}$. Substituting Eq. (6.53) into Eq. (6.52), and interchanging the order of addition and summation (which has been described as "illegitimate but expedient" [21]), we obtain the approximation

$$\frac{e^{-jk_0|\vec{r}+\vec{d}|}}{|\vec{r}+\vec{d}|} = -\frac{jk_0}{4\pi}\oint_S e^{-j\vec{k}\cdot\vec{d}} T_L(\hat{k}\cdot\hat{r})\, d^2\hat{k} \tag{6.54}$$

with

$$T_L(\hat{k}\cdot\hat{r}) = \sum_{l=0}^{L}(-j)^l(2l+1)h_l^{(2)}(k_0 r)P_l(\hat{k}\cdot\hat{r}) \tag{6.55}$$

The first key point in the FMM is the function $T_L(\hat{k}\cdot\hat{r}) = T_L(\kappa,\theta)$ with $\kappa = k_0 r$ *precomputed* for various values of distance κ and various angles θ. This is a truncated multipole expansion, hence the name: it has been shown semi-empirically that the number of multipoles is approximately $k_0 D + 6(k_0 D)^{1/3}$ (with D the maximum dimension applicable) for an accuracy of 10^{-6}.

The second key point of the FMM is that the interaction matrix is divided into *near* and *far* parts. *Near* interactions are computed as usual with the MoM, and the FMM does not change these at all (by contrast, FFT methods evaluate *all* matrix elements). *Far* interactions are evaluated *approximately*, using the above function T_L. Basis functions in the *far* region are grouped into M localized groups – it has been shown that the optimal value of this is $\sqrt{N}$, with N the number of basis functions.

The third key point in the FMM is that the (approximate) matrix-vector product may be done in $\mathcal{O}(N^{3/2})$ operations. This is done by first computing the far fields of each *group*, then computing the Fourier components of the field in the neighborhood of each group generated by non-near sources, and finally adding the effects of the near- and far-group interactions. These steps are also known as *aggregation*, lumping the fields radiated by a group to the group center, *translation and summation*, which sends the fields from one group to another and then sums them, and finally *disaggregation*, which distributes the received field to each point within the receiving group.

By introducing a recursive hierarchy of groups, the operation count can be further reduced to $\mathcal{O}(N \log N)$; this is known as the multilevel fast multipole algorithm (MLFMA).

The above description is very cursory, and the interested reader is referred to Section 6.9 for references which provide far more detail. We should caution that the constants in the operation counts can be very large, easily on the order of many thousands or more (by contrast, for direct methods or matrix-vector multiplication, the constants are usually on the order of unity) so the FMM and MLFMA are

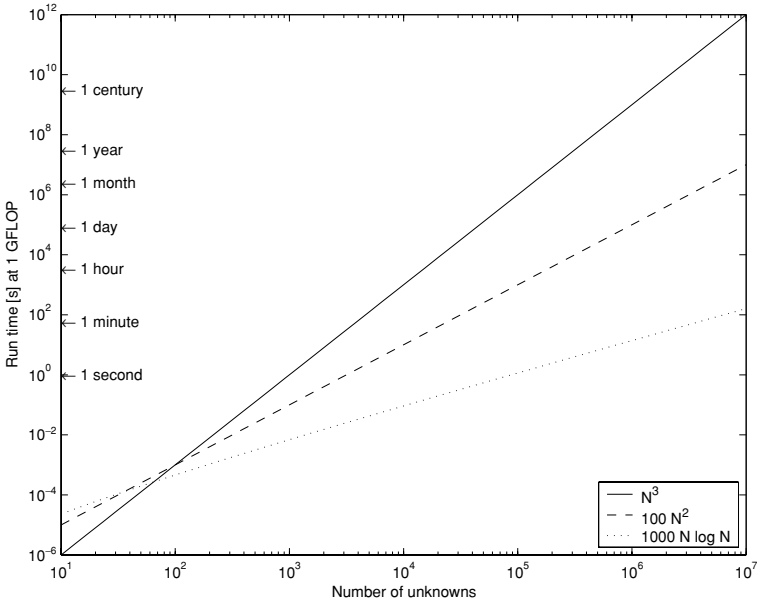


Figure 6.15 Run-times for typical LU decomposition, a very rapidly converging iterative solution and a well-optimized FMM solution on a 1 gigaflop system. (Adapted from [22, Fig.14.8].)

only *asymptotically* "fast"; indeed for small to medium size problems, the FMM will probably be slower than the MoM. Furthermore, for large problems, highly efficient implementation is essential, otherwise the benefits are lost, so an FMM implementation is emphatically *not* a project for beginners.

The impact of a reduction in asymptotic cost is not always immediately apparent. To illustrate this, Fig. 6.15 compares the run-time on a system capable of sustaining 1 gigaflop for N^3, $100N^2$ (as one might hope to obtain with a very rapidly converging iterative solver) and $1000N \log N$, as one might obtain with a very well optimized FMM code, as suggested by [22, Fig. 14.8]. Clearly, the impact of reducing this asymptotic cost is enormously significant for large problems; the difference with the assumed operation counts for 1 million unknowns is that of minutes versus decades! (In reality, the FMM code is likely to run for many hours at least, but the point remains valid.) It must be commented that the constants assumed in both the iterative and FMM cases above may well be extremely optimistic.

The impact on memory is also highly significant; Fig. 6.16 compares the memory required to store the full MoM matrix compared to the storage requirements of a proposed FMM implementation, as suggested by [22, Fig. 14.9]. (Note that each complex word requires 8 bytes to store in single precision on typical systems.)

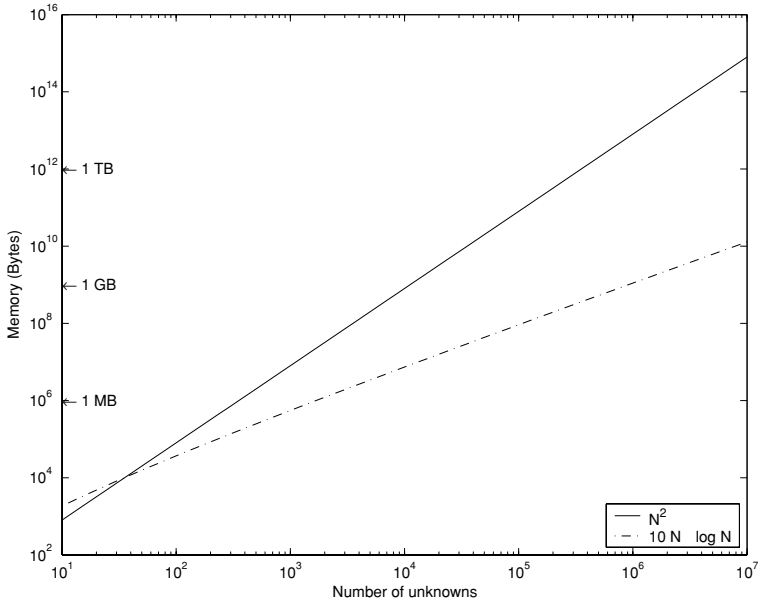


Figure 6.16 Memory required for LU decomposition and a proposed FMM implementation. (Adapted from [22, Fig.14.8].)

Again, one should note that a real FMM implementation is unlikely to be this memory efficient.

6.9 Further reading

The electric and magnetic field integral equations are covered in a number of texts on electromagnetic theory and CEM. There are many equivalent different forms, depending on how the differentials are treated; those in this chapter are based on [23]. An introductory treatment may be found in [5, Chapter 12]. Good treatments will also be found in [22, Chapter 14], with more on the underlying theory in [1, Sections 6–9 and 12–3]. The topic is also discussed in [24]. A point which we have glossed over in this chapter is that both the EFIE and MFIE exhibit a phenomenon known as *interior resonance*. Essentially, a (theoretically) non-radiating *interior* eigenmode is also present in the MoM solution procedure,[13] and due to numerical inaccuracies, the eigenmode incorrectly contributes to radiation. Canning showed that there is a component of the field equations which should annihilate this term, but that this term is slightly "off" in frequency in the discrete MoM solution, hence the problem. He proposed a method using singular value decomposition to

[13] We assume here the usual exterior field problem.

remove this term [25]; although it worked well for canonical problems, Steyn and the present author showed that it was difficult to apply to more general problems [26]. The topic of interior resonances in general has been quite extensively discussed in the literature; in practice, it is usually a very narrowband phenomenon, and for simple problems (in particular ones where the eigenvalues can be predicted analytically) can simply be "smoothed" through, but a rigorous solution requires a combination of both EFIE and MFIE, as the combined field integral equation. A particularly comprehensive discussion of this may be found in [19, Chapter 6].

In the context of equivalent surface current modelling, discussions of the equivalence principle will be found in several standard texts; that in [5, Section 7.8] is especially useful. For the modelling of homogeneous and inhomogeneous material bodies, few textbooks discuss this topic – [1, Chapter 12] being a notable exception – and one will need to refer largely to journal papers. One of the earliest papers to consider this was Richmond's [27], although his formulation was essentially a volume equivalence one. For details of the surface equivalence formulation, [28] provides a comprehensive discussion and an extensive, although not exhaustive, list of references. The discussion of the equivalence principle is often quite cursory; a particularly detailed study has recently been published by Booysen [29].

On hybrid MoM/PO methods, Jakobus and Landstorfer's original papers [14, 30] remain the best reference.

Regarding parallel processing, the present author made some of the earlier contributions in this regard [17, 18, 31]; other early work may be found in [32]. With Cwik, the present author recently summarized much of the state-of-the-art [33]; this special issue contains papers by many of the researchers active in the field in the mid to late 1990s.

There is now a large body of literature dealing with fast techniques in CEM. A very readable introductory treatment will be found in [19, Chapter 4]. Jin provides a detailed, up-to-date and yet succinct overview of fast methods in general in [22, Chapter 14], and this would serve well as a first reference for more detailed study; a fairly extensive list of references complements the technical descriptions. On a historical note, Bojarski is credited with the first use of the FFT method in electromagnetics for this purpose,[14] in a US Air Force technical report of 1971, although the work was only published in the archival open literature a decade later [34]. The application of the FFT to surface and volumetric scattering is well illustrated by the work of Zwamborn and van den Berg, of which [35] is a good example, and also by Borup and Gandhi [36]. For some of the early work on iterative methods, the papers by Sarkar contain useful descriptions of the iterative algorithms ([37] is typical), but it should be noted that there are misconceptions in this and other

[14] He used the term "k-space" in his work rather than CGFFT.

papers about the nature of discrete operators. This led to a lengthy debate in the literature (see [38], for instance, as well as comments in [24, Chapter 1]); this was finally settled by Ray and Peterson [39]. Their closing comment is conclusive:

> While direct iterative methods may be very efficient for some problems, they are no more accurate than their moment-method analogs.

On the FMM, the paper by Coifman, Rokhlin and Wandzura [21] remains a classic; the paper belies its title, providing the essential ideas and outlining the implementation in only six pages. (Note that they use the $e^{-i\omega t}$ convention widely used in physics, so the signs of i are reversed relative to the discussion in Section 6.8.4, and the spherical Hankel function is of the first kind.) Chew and colleagues at Illinois have been prolific users of the method; their recent book provides a detailed discussion of the many applications [40], and their review paper provides a succinct overview of the field [41]. On the question of error control, the paper by Botha and the present author presents a detailed discussion [42].

6.10 Concluding comments

In this chapter, we have studied methods of solving currents on surfaces using the MoM, starting with the electric and magnetic field integral equations. These may be real currents, in the case of a PEC, or fictitious ones, in the case of an homogeneous dielectric (or magnetic) body. Some theoretical background on the RWG surface basis functions has also been provided, since these are widely used in commercial codes. The ability to model homogeneous material bodies using fictitious equivalent surface currents is very useful indeed; some MoM codes, such as FEKO, can also handle *inhomogeneous* material bodies, using an equivalent volume current method, but the computational cost associated with this is extremely high, as we have seen (unless FFT-based methods are used).

The much larger computational requirements of surface modelling as opposed to thin-wire modelling have been discussed comprehensively. A hybrid MoM/PO formulation has been outlined. Although inherently approximate, this permits large structures to be modelled with good accuracy provided caution is exercised; it is particularly useful for what is often called "installed antenna performance modelling," which frequently involves electrically small antennas mounted on electrically large vehicles (used here in the general sense to include aircraft, spacecraft and ships). A commercial implementation of this theory is available and we have shown an example of its use. High-performance computing has also been discussed; this continues to be an important enabling technology driving very large applications of the method. Finally, "fast" methods have been considered, including the original FFT-based methods, extensions in the form of the adaptive integral

method, and of course the fast multipole method. The last in particular rejuvenated the method of moments in the early 1990s and has proven one of the most important theoretical advances in the MoM over the last two decades.

References

[1] A. Ishimaru, *Electromagnetic Wave Propagation, Radiation and Scattering*. Engelwood Cliffs, NJ: Prentice-Hall, 1991.

[2] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, **30**, 409–418, May 1982.

[3] G. S. Smith, *An Introduction to Classical Electromagnetic Radiation*. Cambridge: Cambridge University Press, 1997.

[4] H. A. Haus and J. R. Melcher, *Electromagnetic Fields and Energy*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[5] C. A. Balanis, *Advanced Engineering Electromagnetics*. New York: Wiley, 1989.

[6] J. A. Stratton, *Electromagnetic Theory*. New York: McGraw-Hill, 1941.

[7] R. F. Harrington, *Time-Harmonic Electromagnetic Fields*. New York: McGraw-Hill, 1961.

[8] W. H. Press, S. A. Teukolsky, W. Vettering, and B. R. Flannery, *Numerical Recipes in Fortran: the Art of Scientific Computing*. Cambridge: Cambridge University Press, 2nd edn., 1992.

[9] J. R. Mautz and R. F. Harrington, "Electromagnetic scattering from a homogeneous material body of revolution," *Arch. Elektron. Übertragungstech.*, **33**, 71–80, 1979.

[10] D. Davidson and S. Keunecke, "Hybrid techniques using the MOM and PO/UTD: a tutorial overview," *Trans. S. Afr. Inst. Electr. Eng.*, **90**(2) 69–82, June 1999.

[11] U. Jakobus, I. Sulzer, and F. M. Landstorfer, "Parallel implementation of the hybrid MoM/Green's function technique on a cluster of workstations," in *10th International Conference on Antennas and Propagation*, vol. 1, pp. 1.182–1.185. IEE Conference Publication No. 436, 1996.

[12] C. T. Tai, *Dyadic Green's Functions in Electromagnetic Theory*. New York: IEEE Press, 2nd edn., 1994.

[13] E. H. Newman, "An overview of the hybrid MM/Green's function method in electromagnetics," in *Moment Methods in Antennas and Scattering* (R. C. Hansen, ed.), pp. 449–461. Norwood, MA: Artech House, 1990.

[14] U. Jakobus and F. M. Landstorfer, "Improved PO-MM hybrid formulation for scattering from three-dimensional perfectly conducting bodies of arbitrary shape," *IEEE Trans. Antennas Propag.*, **43**, 162–169, February 1995.

[15] R. E. Hodges and Y. Rahmat-Samii, "An iterative current-based hybrid method for complex structures," *IEEE Trans. Antennas Propag.*, **45**, 265–276, February 1997.

[16] M. J. Flynn, "Some computer organizations and their effectiveness," *IEEE Trans. Comput.*, **21**, 948–60, September 1972.

[17] D. B. Davidson, "Parallel matrix solvers for moment method codes for MIMD computers," *Appl. Comput. Electromagn. Soc. J.*, **8**(2), 144–175, 1993.

[18] D. B. Davidson, "Parallel processing revisited: a second tutorial," *IEEE Antennas Propag. Mag.*, **34**, 9–21, October 1992.

[19] A. F. Peterson, S. L. Ray, and R. Mittra, *Computational Methods for Electromagnetics*. Oxford and New York: Oxford University Press and IEEE Press, 1998.

[20] M. Abramowitz and I. A. Stegun, eds., *Handbook of Mathematical Functions*. New York: Dover, 1972.
[21] R. Coifman, V. Rohklin, and S. Wandzura, "The fast multipole method for the wave equation: a pedestrian prescription," *IEEE Antennas Propag. Mag.*, **35**, 7–12, June 1993.
[22] J. Jin, *The Finite Element Method in Electromagnetics*. New York: Wiley, 2nd edn., 2002.
[23] P. P. Silvester and R. L. Ferrari, *Finite Elements for Electrical Engineers*. Cambridge: Cambridge University Press, 3rd edn., 1996.
[24] J. J. H. Wang, *Generalized Moment Methods in Electromagnetics*. New York: Wiley, 1991.
[25] F. X. Canning, "Singular value decomposition of integral equations of em and applications to the cavity resonance problem," *IEEE Trans. Antennas Propag.*, **37**, 1156–1163, September 1989.
[26] P. Steyn and D. B. Davidson, "A technique for avoiding the EFIE "interior resonance" problem applied to an MM solution of electromagnetic radiation from bodies of revolution," *Appl. Comput. Electromagn. Soc. J.*, **10**(3), 116–128, 1995.
[27] J. H. Richmond, "Scattering by a dielectric cylinder of arbitrary cross section shape," *IEEE Trans. Antennas Propag.*, **13**, 334–341, May 1965.
[28] K. Umashankar, A. Taflove, and S. M. Rao, "Electromagnetic scattering by abritrary shaped three-dimensional homogeneous lossy dielectric objects," *IEEE Trans. Antennas Propag.*, **34**, 758–766, June 1986.
[29] R. Booysen, "Aperture theory and the equivalence principle," *IEEE Antennas Propag. Mag.*, **45**, 29–40, June 2003.
[30] U. Jakobus and F. M. Landstorfer, "Improvement of the PO-MM hybrid method by accounting for effects of perfectly conducting wedges," *IEEE Trans. Antennas Propag.*, **43**, 1123–1129, October 1995.
[31] D. B. Davidson, "A parallel processing tutorial," *IEEE Antennas Propag. Mag.*, **32**, 6–19, April 1990.
[32] T. Cwik and J. Patterson, eds., *Computational Electromagnetics and Supercomputer Architecture*. PIER7 Progress in Elecromagnetics Research. Cambridge, MA: EMW Publishing, 1993.
[33] D. B. Davidson and T. Cwik, guest eds., "Special issue on computational electromagnetics and high-performance computing," *Appl. Comput. Electromagn. Soc. J.*, **13**, 87–225, July 1998.
[34] N. N. Bojarski, "The k-space formulation of the scattering problem in the time domain," *J. Acoust. Soc. Am.*, **72**, 570–584, August 1982.
[35] A. P. M. Zwamborn, P. M. van den Berg, J. Mooibroek, and F. T. C. Koenis, "Computation of three-dimensional electromagnetic-fields distributions in a human body using the weak form of the CGFFT method," *Appl. Comput. Electromagn. Soc. J.*, **7**, 26–42, 1992.
[36] D. T. Borup and O. P. Gandhi, "Fast-fourier transform method for calculation of SAR distributions in finely discretized inhomogeneous models of biological bodies," *IEEE Trans. Microwave Theory Tech.*, **32**, 355–360, April 1984.
[37] T. K. Sarkar, "The conjugate gradient method as applied to electromagnetic field problems," *IEEE Antennas Propag. Soc. Newsletter*, **28**, 5–14, August 1986.
[38] T. K. Sarkar, "Comments on "Comparison of the FFT conjugate gradient method and the finite-difference time domain method for the 2-D absorption problem" and reply by D. T. Borup and O. P. Gandhi," *IEEE Trans. Microwave Theory Tech.*, **36**, 166–170, January 1988.

[39] S. L. Ray and A. F. Peterson, "Error and convergence in numerical implementation of the conjugate gradient method," *IEEE Trans. Antennas Propag.*, **36**, 1824–1827, December 1988.
[40] W. C. Chew, J. Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*. Boston, MA: Artech House, 2001.
[41] W. C. Chew, J. Jin, C. Lu, E. Michielssen, and J. M. Song, "Fast solution methods in electromagnetics," *IEEE Trans. Antennas Propag.*, **45**, 533–543, March 1997.
[42] M. M. Botha and D. B. Davidson, "Application of the fast multipole method to the FE-BI analysis of cavity backed structures with comprehensive FMM error control," *Electromagnetics*, **22**, 405–417, July 2002.

7

The method of moments and stratified media: theory

7.1 Introduction

Modelling *stratified media* is an important application of the MoM. A stratified medium is one consisting of homogeneous layers of material, each layer having different electromagnetic properties. This includes the general category of *printed antennas*, of which microstrip is the best known. (Microstrip technology is discussed in more detail in the next chapter.) It also brings with it the problem of dealing with dielectric materials. Central to this is the issue of the *Green function*[1] for the problem. The MoM relies on an appropriate Green function as the "field propagator." Due to its perceived complexity, the topic of stratified media is generally regarded as an advanced one, and the coverage tends to be highly theoretical, and frequently impenetrable without lengthy study. One reason for this is that historically, analysis focussed on the problem of a dipole above a dielectric half-space. There are a number of complex issues which this raises, requiring quite sophisticated analytical techniques to understand, in particular for the asymptotic cases where interesting radiation physics can be extracted. However, the analysis of a very important special case, namely the grounded single-layer microstrip line (or patch antenna), can be undertaken without undue complexity, at least for most practical cases where the substrate is relatively thin.

In this chapter, a static analysis of a microstrip transmission line is first undertaken, to demonstrate the basic principles of the spectral domain and the derivation of the Green function. Following this, the dynamic analysis is introduced, and the Sommerfeld potentials derived from first principles. Although the work in this chapter is certainly not original, being based on a synthesis of the literature – in particular [1] – the presentation in the present format does not appear to have been thus undertaken in other works to date.

[1] Contemporary usage is "Green function" rather than "Green's function," in line with "Dirac delta function," "Heaviside step function" etc.

7.2 Dyadic Green functions: some introductory notes

The main reason for efficiency of the MoM formulation already discussed is the existence of suitable Green functions. The Green function $G(\vec{r})$ is equivalent to the *impulse response* $h(t)$ of system theory. Just as $h(t)$ gives the response (in time) to a *temporally* impulsive source, so $G(\vec{r})$ gives the response (in space) to a *spatially* impulsive (current) source. The response to a spatially distributed source is obtained by integration, and plays the same role in space that *convolution* in system theory does in time:

$$y(t) = h(t) * x(t) \iff \vec{E}(\vec{r}) = \bar{\bar{G}}(\vec{r}, \vec{r}') * \vec{J}(\vec{r}') \tag{7.1}$$

We have already encountered the *free-space Green function* in our work in Chapter 4, although we made only passing reference to it then. In free space, the function is (moderately) simple:

$$\bar{\bar{G}}(\vec{r}, \vec{r}') = \left(k^2 \bar{\bar{I}} + \nabla\nabla\right) g(\vec{r}, \vec{r}'), \qquad g(\vec{r}, \vec{r}') = \frac{e^{-jkR}}{4\pi R} \tag{7.2}$$

where $R = |\vec{r} - \vec{r}'|$ is the distance from source to field point. Green functions can be obtained for either fields or potentials, and in the above, $\bar{\bar{G}}(\vec{r}, \vec{r}')$ is the electric field Green function for free space, and $g(\vec{r}, \vec{r}')$ is the potential Green function for free space. We will primarily use Green functions for potentials in this chapter. It is worth highlighting that the Green function for free space is given in *closed form* and is trivial to compute (although the singularities which accompany it make an accurate MoM implementation anything but!).

Some new notation has been introduced in the above. The double-overbar notation indicates a *dyad*; this is a mathematical device which after multiplication by a vector, yields a vector. A dyad typically consists of the following terms, when written as a matrix:

$$\bar{\bar{G}} = \begin{bmatrix} G^{xx} & G^{xy} & G^{xz} \\ G^{yx} & G^{yy} & G^{yz} \\ G^{zx} & G^{zy} & G^{zz} \end{bmatrix} \tag{7.3}$$

It is also frequently written out in its component form:

$$\begin{aligned} \bar{\bar{G}} = {} & G^{xx}\hat{x}\hat{x} + G^{xy}\hat{x}\hat{y} + G^{xz}\hat{x}\hat{z} + \\ & G^{yx}\hat{y}\hat{x} + G^{yy}\hat{y}\hat{y} + G^{yz}\hat{y}\hat{z} + \\ & G^{zx}\hat{z}\hat{x} + G^{zy}\hat{z}\hat{y} + G^{zz}\hat{z}\hat{z} \end{aligned} \tag{7.4}$$

The product of a dyad and vector is then computed using normal matrix theory or the usual vector dot-products. $\bar{\bar{I}}$ is the identity dyad. Note that although both

operations $\hat{s} \cdot \bar{\bar{G}}(\vec{r}, \vec{r}')$ and $\bar{\bar{G}}(\vec{r}, \vec{r}') \cdot \hat{s}$ with $\hat{s}$ a unit vector (i.e. $\hat{x}$, $\hat{y}$ or $\hat{z}$) are defined, only the latter has physical meaning as the potential due to an $\hat{s}$-oriented source.

However, for many applications (such as printed antennas, antennas above or buried in a real earth) radiation occurs in a *stratified media* environment, not free space. The presence of the stratified media greatly complicates the analysis. The Green function for an elementary dipole radiating in the vicinity of the stratified medium needs to be worked out. This was done many years ago by Arnold Sommerfeld – in 1909, he determined the field radiated by a short vertical electrical dipole above a dielectric interface. However, the passage of time has not made the theory any easier. In particular, the required integration in the complex plane brings with it a number of complex issues. Finally, the Green functions obtained are *not* given in closed form, and are computationally expensive to compute, so even implementations of seemingly simple problems require some thought.

Before concluding this introductory section, it should be commented that there are a number of MoM formulations for stratified media. This chapter uses the mixed potential integral equation formulation (MPIE), introduced by Mosig and Gardiol [2] and used with great success for MoM formulations by a number of workers. However, before we outline this approach, we will consider a much simpler problem, which illustrates many of the issues: deriving the Green function for stratified media for electrostatics from first principles.

7.3 A static example of a stratified medium problem: the grounded dielectric slab

Central to stratified media formulations is the spectral domain transform. The Fourier transform is used to simplify the problem by transforming the partial differential equation(s) of electromagnetics in the spatial domain into an ordinary differential equation in the spectral domain. (Once again, the analogy with linear systems theory is strong.) To illustrate the basic concepts, we will derive the static spectral domain Green function for a microstrip structure, as shown in Fig. 7.1. This does *not* include radiation effects, which requires the full-wave solution of the problem, the topic of later parts of this chapter. This is still quite useful, nonetheless: the quasi-TEM approach often used for transmission-line analysis renders the problem (quasi-)static. A solution can be used to compute the characteristic impedance and phase constant of the transmission line by making the calculation twice – once with the dielectric present, and once with the dielectric replaced by free space [3, p. 166]. Note that the structure is assumed to be of infinite length, thus there is no variation in y.

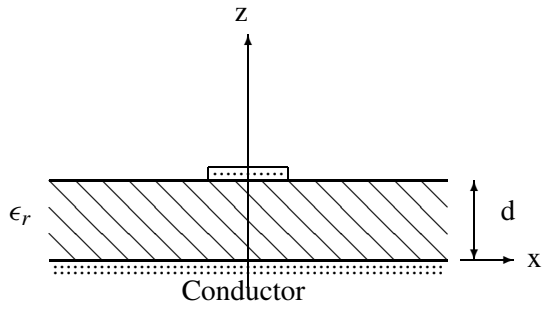


Figure 7.1 Typical microstrip structure.

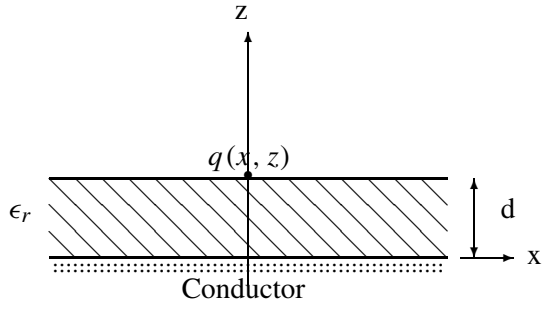


Figure 7.2 Stratified medium equivalent with impulsive source $q(x, z) = \delta(x)\delta(z - d)$.

This formulation appears to have been originally presented in the engineering literature by Yamashita and Mittra [4]. They did not actually derive the Green function; they were formulating a variational expression for the unknown charge distribution on the strip, but the extension is straightforward. Their notation is largely followed here, except that k_x is used as the Fourier transform variable instead of β, and d instead of h for the substrate thickness. Booton provides a similar derivation [5, Section 10.3]. It is interesting to note that an almost identical derivation may be found in Schwinger's lecture notes [6, Chapter 14]; although only recently published, these lectures were originally given in 1976.

To derive the Green function, the Poisson equation for a spatially impulsive source of unit magnitude located at $x = 0, z = d$ must be solved (subsequently, the case $x \neq 0$ is also considered); see Figs. 7.1 and 7.2. Thus the partial differential equation to solve is:

$$\nabla^2 \Phi(x, z) = -\frac{1}{\epsilon}\delta(x)\delta(z - d) \tag{7.5}$$

The equation is transformed into the spectral domain; using the linearity of the Fourier transform, the $\frac{\partial}{\partial x} \Longleftrightarrow jk_x$ transform property, and the Fourier transform

of the Dirac delta function, one obtains

$$\left[-k_x^2 + \frac{d^2}{dz^2}\right]\tilde{\Phi}(k_x, z) = -\frac{1}{\epsilon}\delta(z-d) \tag{7.6}$$

with $\tilde{\Phi}(k_x, z)$ the Fourier transform of the potential, also known as the spectral domain representation:

$$\tilde{\Phi}(k_x, z) = \int_{-\infty}^{\infty} \Phi(x, z) e^{-jk_x x} dx \tag{7.7}$$

Note that this is now an *ordinary* differential equation in $\tilde{\Phi}(k_x, z)$. The homogeneous differential equation (with the inhomogeneous source taken into account via a Neumann boundary condition) is now solved:

$$\left[-k_x^2 + \frac{d^2}{dz^2}\right]\tilde{\Phi}(k_x, z) = 0, \qquad \forall z \neq d \tag{7.8}$$

The boundary conditions are: zero potential at $z = 0$ and $z \to \infty$; continuous potential at the material interface at $z = d$; and flux discontinuous by the source singularity at $z = d$. These boundary conditions transform in a straightforward fashion to the spectral domain. The solution to Eq. (7.8) must be written in the two regions demarcated by the material interface. Note that even if $\epsilon_r = 1$, this two-region approach is still necessary, so that the jump discontinuity can be enforced.

The boundary conditions, transformed into the spectral domain, are:

$$\tilde{\Phi}(k_x, 0) = 0 \tag{7.9}$$

$$\tilde{\Phi}(k_x, \infty) = 0 \tag{7.10}$$

$$\tilde{\Phi}(k_x, d^+) = \tilde{\Phi}(k_x, d^-) \tag{7.11}$$

$$\epsilon_0 \frac{d}{dy}\tilde{\Phi}(k_x, d^+) = \epsilon_0 \epsilon_r \frac{d}{dy}\tilde{\Phi}(k_x, d^-) - 1 \tag{7.12}$$

The solution of Eq. (7.6) is in the form of exponentials in each region:

$$\tilde{\Phi}_1(k_x, z) = A\, e^{-k_x z} + B\, e^{k_x z}, \qquad \forall 0 \leq z < d \tag{7.13}$$

$$\tilde{\Phi}_2(k_x, z) = C\, e^{-|k_x| z} + D, \qquad \forall z \geq d \tag{7.14}$$

Equation (7.10) immediately yields $D = 0$, and Eq. (7.9) yields $A = -B$. Thus

$$\tilde{\Phi}_1(k_x, z) = -2A \sinh k_x z \tag{7.15}$$

Applying Eq. (7.11) in the limit $d^{\pm} \to d$ one obtains

$$A = -C \frac{e^{-|k_x| d^+}}{2 \sinh k_x d^-} \tag{7.16}$$

The d/dz terms in Eq. (7.12), again in the limit $d^{\pm} \to d$, are thus:

$$\left.\frac{d\tilde{\Phi}_1(k_x, z)}{dz}\right|_{z=d^-} = +Ck_x\, e^{-|k_x|d} \coth k_x d^+$$

$$\left.\frac{d\tilde{\Phi}_2(k_x, z)}{dz}\right|_{z=d^+} = -C|k_x|\, e^{-|k_x|d^-}$$

Equation (7.12) yields:

$$C = \frac{e^{|k_x|d}}{\epsilon_0|k_x|[1 + \epsilon_r \coth|k_x|d]} \tag{7.17}$$

where the even property of the product of $\coth(k_x d)$ and $k_x d$ has been used to make the required simplification $k_x \coth k_x d = |k_x| \coth|k_x|d$ (assuming $d \geq 0$). The solution for $\tilde{\Phi}_2(k_x, z)$, valid in the limit $d^+ \to d$ for $z \geq d$ is thus:

$$\tilde{\Phi}(k_x, z) = \frac{e^{|k_x|(d-z)}}{\epsilon_0|k_x|[1 + \epsilon_r \coth|k_x|d]} \tag{7.18}$$

We have dropped the subscript 2 since we are now on the interface. Note that for $z = d$, this reduces to:

$$\tilde{\Phi}(k_x, d) = \frac{1}{\epsilon_0|k_x|[1 + \epsilon_r \coth|k_x|d]} \tag{7.19}$$

This can also be written as:

$$\tilde{\Phi}(k_x, d) = \frac{\sinh|k_x|d}{\epsilon_0|k_x|\{\sinh|k_x|d + \epsilon_r \cosh|k_x|d\}} \tag{7.20}$$

(An interesting special case can be identified, viz. $\epsilon_r = 1$. For this case, by expanding the hyperbolic terms in the denominator, Eq. (7.20) reduces to

$$\tilde{\Phi}(k_x, d) = \frac{1}{\epsilon_0} \frac{e^{-|k_x|d} \sinh|k_x|d}{|k_x|} \tag{7.21}$$

This can be useful in asymptotic analysis, where the Green function for a homogeneous dielectric is used.)

Equation (7.20) is the spectral domain Green function for a source located on the z-axis. The Green function is then the inverse Fourier transform of this:

$$G(x, 0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \tilde{\Phi}(k_x, d)\, e^{jk_x x}\, dk_x \tag{7.22}$$

and for the general case of the source located at x', this becomes

$$G(x, x') = \frac{1}{2\pi} \int_{-\infty}^{\infty} \tilde{\Phi}(k_x, d)\, e^{jk_x(x-x')}\, dk_x \tag{7.23}$$

The required integral equation for the potential in terms of the charge distribution $\rho(x, d)$ is thus

$$\Phi(x, d) = \int_{-\infty}^{\infty} G(x, x')\rho(x', d)\, dx' \tag{7.24}$$

This, then, is the spectral domain static Green function for a grounded dielectric slab. Unfortunately, we note that it must first be inverse Fourier transformed to the spatial domain, and doing this for each possible value of the argument $x - x'$ is very time consuming, since numerical integration is required. Interpolation tables are often used to accelerate the evaluation of the functions. Another approach is to formulate the entire MoM problem in the spectral domain, by using basis functions which have analytical Fourier transforms. This is described in detail for the quasi-static microstrip analysis problem in [7]. However, we will not pursue this further here. Instead, we turn our attention to the full-wave case, after first revising some concepts from electromagnetic theory regarding scalar and vector potential representations.

7.4 The Sommerfeld potentials

7.4.1 A brief revision of potential theory

Before confronting the full-wave stratified medium problem, we will briefly revise some basic electromagnetic theory, in particular, potential theory. It is often useful to represent fields in terms of potentials. Classic elementary electrostatics uses $\vec{E} = -\nabla\Phi$. For high-frequency electromagnetics the electrostatic potential is of course incomplete, and a very widely used set of potentials is

$$\vec{E} = -\nabla\Phi - \frac{\partial \vec{A}}{\partial t} \tag{7.25}$$

$$\vec{B} = \nabla \times \vec{A} \tag{7.26}$$

It will be recalled that there is considerable arbitrariness surrounding the choice of potential (as is well known, a potential $\vec{A}' = \vec{A} + \nabla\phi$ with ϕ any suitable scalar function results in the same set of fields); this is usually resolved via a *gauging* process. The most widely used in RF engineering is the "Lorenz gauge," with

$$\nabla \cdot \vec{A} = -(1/c^2)\frac{\partial \Phi}{\partial t} \tag{7.27}$$

A and ϕ must to be worked out from

$$\nabla^2\phi - \frac{1}{c^2}\frac{\partial^2\Phi}{\partial t^2} = -\frac{\rho}{\epsilon} \tag{7.28}$$

$$\nabla^2\vec{A} - \frac{1}{c^2}\frac{\partial^2\vec{A}}{\partial t^2} = -\mu\vec{J} \tag{7.29}$$

In the frequency domain, these become

$$\left(\nabla^2 + k^2\right)\Phi = -\frac{\rho}{\epsilon} \tag{7.30}$$

$$\left(\nabla^2 + k^2\right)\vec{A} = -\mu\vec{J} \tag{7.31}$$

and these solutions – for differential current elements $d\rho$ and $d\vec{J}$ – are the potential Green functions.

We have already commented that within one potential representation, the potentials are not unique. There is also more than one possible potential representation. Another set involving *only* electric and magnetic *vector* potentials may be used; this was originally introduced by Hertz. In this case, the potentials satisfy the following Helmholtz equations:

$$\left(\nabla^2 + k^2\right)\vec{A} = -\mu\vec{J} \tag{7.32}$$

$$\left(\nabla^2 + k^2\right)\vec{F} = -\epsilon\vec{M} \tag{7.33}$$

where $\vec{M}$ is the (fictitious) magnetic current. These are also sometimes written as $\vec{\Pi}^e = \frac{\vec{A}}{j\omega\mu\epsilon}$ and $\vec{\Pi}^h = \frac{\vec{F}}{j\omega\mu\epsilon}$. For the Hertz potentials, the fields in the spatial domain are given as:

$$j\omega\mu\epsilon\vec{E} = k^2\vec{A} + \nabla\cdot\nabla\vec{A} - j\omega\mu\nabla\times\vec{F} \tag{7.34}$$

$$j\omega\mu\epsilon\vec{H} = k^2\vec{F} + \nabla\cdot\nabla\vec{F} + j\omega\mu\nabla\times\vec{A} \tag{7.35}$$

7.4.2 The Sommerfeld potentials

Preliminaries

In the stratified medium case, at least two approaches using potentials have been used. The former uses the field components normal to the interface as potentials. We will retain the convention of the preceding sections that the interfaces are in planes of constant z; hence, in this case, the potentials would be E_z and H_z. Another possibility is the use of the (Hertz) potentials, of both electric ($\vec{A}$) and magnetic ($\vec{F}$) type. If only z-directed components A_z and F_z are retained, this choice is

traditionally called the Hertz–Debye potentials. The final possibility, and the one we will investigate since it is the most popular, is the Sommerfeld potentials.

The Sommerfeld potentials, in the absence of magnetic currents, assume $\vec{F} = 0$. A vertical electric dipole (VED), i.e. z-directed in our convention, needs only the A_z component. A horizontal electric dipole (HED) (i.e. parallel to the x–y plane) will require a component parallel to the source. Hence, the dyadic in this approach will have only five non-zero terms:

$$\bar{\bar{G}}_A = (\hat{x}G_A^{xx} + \hat{z}G_A^{zx})\hat{x} + (\hat{y}G_A^{yy} + \hat{z}G_A^{zy})\hat{y} + \hat{z}G_A^{zz}\hat{z} \tag{7.36}$$

In order to find these terms, we first need some additional background on the spectral domain.

The spectral domain transform

In the static case discussed previously, no $\hat{y}$ variation was assumed, and the Fourier transform was the usual one-dimensional one. For a general structure, we cannot make this assumption, and the transform (and inverse) becomes two dimensional:

$$\tilde{f}(k_x, k_y) = \frac{1}{2\pi}\int\int_{-\infty}^{\infty} f(x, y)\, e^{-jk_x x}\, e^{-jk_y y}\, dx\, dy \tag{7.37}$$

$$f(x, y) = \frac{1}{2\pi}\int\int_{-\infty}^{\infty} \tilde{f}(k_x, k_y)\, e^{jk_x x}\, e^{jk_y y}\, dk_x\, dk_y \tag{7.38}$$

It is useful to introduce the polar vector $\rho = x\hat{x} + y\hat{y}$ (this is simply the usual radius vector in cylindrical coordinates, $|\vec{\rho}| = \sqrt{x^2 + y^2}$) and the radial spectral variable $\vec{k}_\rho = k_x\hat{x} + k_y\hat{y}$. This permits the "del" operator ∇ to be split into its transverse and normal parts as $\nabla = \nabla_t + \frac{\partial}{\partial z}\hat{z}$. In the spectral domain, this becomes

$$\tilde{\nabla} = j\vec{k}_p + \frac{\partial}{\partial z}\hat{z} \tag{7.39}$$

Since the only spatial derivative remaining in the spectral domain is with respect to z, the shorter dot notation for derivatives will frequently be used in the following, for example $\partial\tilde{\Psi}/\partial z = \dot{\tilde{\Psi}}$. Using the Bessel function J_0, the above transforms may be written as

$$\tilde{f}(k_\rho) = \int_0^{\infty} J_0(k_\rho\,\rho)\, f(\rho)\, \rho\, d\rho \tag{7.40}$$

$$f(\rho) = \int_0^{\infty} J_0(k_\rho\,\rho)\, \tilde{f}(k_\rho)\, k_\rho\, dk_\rho \tag{7.41}$$

This is known as the Fourier–Bessel or Hankel integral transform pair. These are best known amongst RF and microwave engineers as Sommerfeld integrals.

As in the two-dimensional static case, the introduction of these transforms permits the spatial domain differential equation (the Helmholtz, rather than the Laplace of the static case)

$$\left(\nabla^2 + k^2\right)\Psi = 0 \tag{7.42}$$

to be written in the spectral domain as the solution of an ordinary differential equation

$$\left(\frac{\partial^2}{\partial z^2} - u^2\right)\tilde{\Psi} = 0 \tag{7.43}$$

where the parameter u in the traditional notation of Sommerfeld is given by

$$u^2 = -k_z^2 = k_x^2 + k_y^2 - k^2 = k_\rho^2 - k^2 \tag{7.44}$$

The spectral variable k_ρ is complex valued, and by convention written as $k_\rho = \lambda + j\nu$. λ in this context is the real part of k_ρ, and should not be confused with wavelength.

Normal component representation

One possibility for stratified media is the use of the normal fields E_z and H_z as potentials. The normal components satisfy Eq. (7.42) or (7.43) in the spatial or spectral domain respectively. In the spectral domain, the transverse components are given by:

$$k_\rho^2 \tilde{E}_x = jk_x \dot{\tilde{E}}_z - \omega\mu k_y \tilde{H}_z \tag{7.45}$$

$$k_\rho^2 \tilde{E}_y = jk_y \dot{\tilde{E}}_z + \omega\mu k_x \tilde{H}_z \tag{7.46}$$

$$k_\rho^2 \tilde{H}_x = jk_x \dot{\tilde{H}}_z + \omega\epsilon k_y \tilde{E}_z \tag{7.47}$$

$$k_\rho^2 \tilde{H}_y = jk_y \dot{\tilde{H}}_z - \omega\epsilon k_x \tilde{E}_z \tag{7.48}$$

As in the static case, the boundary conditions transform in a straightforward fashion to the spectral domain. Hence, tangential field continuity across the layers is satisfied if $\epsilon\tilde{E}_z$, $\dot{\tilde{E}}_z$, $\mu\tilde{H}_z$ and $\dot{\tilde{H}}_z$ are continuous. Rather importantly, this means that the boundary conditions do *not* introduce coupled equations in $\tilde{E}_z$ and $\tilde{H}_z$. From the viewpoint of the Green functions, the potentials are the normal components, but we will not pursue this further now. The Sommerfeld potentials make use of some normal components, hence the discussion here.

Sommerfeld potentials

In the absence of magnetic currents,[2] the Sommerfeld approach assumes $\vec{F} = 0$. A VED requires only the A_z component, obtained from the spectral domain relationship

$$j\omega\mu\epsilon\tilde{E}_z = k_\rho^2 \tilde{A}_z \tag{7.49}$$

This is obtained from the spectral domain equivalent of Eq. (7.34). $\tilde{E}_z$ is obtained as above. The other components may be computed from the spectral domain equivalents of Eqs. (7.34) and (7.35). It may be shown that one obtains the following in terms of the normal component representation:

$$\tilde{G}_A^{xx} = -\frac{\mu\tilde{G}_H^{zx}}{jk_y} \tag{7.50}$$

$$k_\rho\tilde{G}_A^{zx} = j\omega\mu\epsilon\tilde{G}_E^{zx} + \frac{k_x\mu\dot{\tilde{G}}_H^{zx}}{k_y} \tag{7.51}$$

$$\tilde{G}_A^{yy} = \frac{\mu\tilde{G}_H^{zy}}{jk_x} \tag{7.52}$$

$$k_\rho\tilde{G}_A^{zy} = j\omega\mu\epsilon\tilde{G}_E^{zy} - \frac{k_y\mu\dot{\tilde{G}}_H^{zy}}{k_x} \tag{7.53}$$

$$k_\rho\tilde{G}_A^{zz} = j\omega\mu\epsilon\tilde{G}_E^{zz} \tag{7.54}$$

Regarding boundary conditions at the interface, it may be shown – from Eqs. (7.34) and (7.35) – using these Sommerfeld potentials, that transverse field continuity implies that $\tilde{A}_z$ and $\tilde{A}_z/\epsilon$ must be continuous for a VED. For an x-directed HED, $\tilde{A}_x$, $\dot{\tilde{A}}_x$, $\tilde{A}_z$, and $\nabla\cdot\tilde{\vec{A}}/\epsilon$ must be continuous, and a similar expression holds for a y-directed HED. The last condition couples normal and transverse components of the Green function, which hence cannot be independently computed. For this reason, it is usually easier to work with the normal field components, as will be done shortly.

Symmetry also results in the following expressions, which we note although we will not use them further:

$$\tilde{G}_A^{xx} = \tilde{G}_A^{yy} \tag{7.55}$$

$$\frac{\tilde{G}_A^{zx}}{jk_x} = \frac{\tilde{G}_A^{zy}}{jk_y} \tag{7.56}$$

[2] As an aside, it should be noted that it is possible to have non-zero $\vec{F}$ even with zero magnetic current $\vec{M}$, due to the amount of arbitrariness in the potentials.

7.4.3 An example: derivation of G_A^{xx} for single-layer microstrip

General multi-layered substrates are best handled using a matrix formulation. Within each substrate, the normal field components are computed for a unit Hertz dipole embedded in the layered medium. The boundary conditions are handled using "chain" matrices. A particularly complete description may be found in [1]. However, for the simple but very important case of a single-layer microstrip, we can directly compute the potentials in a fashion very similar to that described in Section 7.3. Once again, Fig. 7.2 is relevant, although now the impulsive source is a horizontal Hertzian dipole, and for convenience the air–dielectric interface, rather than the ground plane, is at $z = 0$ (and hence the ground plane is located at $z = -d$). In general, the derivation must be repeated for the five non-zero components of the Green function, viz. Eq. (7.36), but we will only derive one of these here – the x-directed magnetic Green function. We also restrict the derivation to non-magnetic lossy dielectric substrates, i.e. $\mu_1 = \mu_0$ and $\epsilon_1 = \epsilon_0\epsilon_r'(1 - \tan\delta)$. We will use $\epsilon_r = \epsilon_r'(1 - \tan\delta)$ to represent the complex relative permittivity in the following; it is useful to be able to distinguish between ϵ_r and ϵ_r'.

The source-free ODE to be solved for the normal magnetic field in the spectral domain is of the form of Eq. (7.43), repeated here for the H_z case:

$$\left(\frac{\partial^2}{\partial z^2} - u^2\right)\tilde{H}_z = 0 \tag{7.57}$$

The solution in each region may either be written as the sum of exponentials, as in Section 7.3, or as hyperbolic functions. In the upper region $z \geq 0$, the solution is of the form

$$\tilde{H}_z = a_0\, e^{-u_0 z} \tag{7.58}$$

which already incorporates the boundary condition at infinity. In the dielectric region, the solution is of the form

$$\tilde{H}_z = a_1 \cosh u_1(z + d) + b_1 \sinh u_1(z + d) \tag{7.59}$$

The remaining boundary conditions on $\tilde{H}_z$ are:

$$\mu_0\mu_r \tilde{H}_z|_{z=0-} = \mu_0 \tilde{H}_z|_{z=0+} \tag{7.60}$$

$$\dot{\tilde{H}}_z|_{z=0-} = \dot{\tilde{H}}_z|_{z=0+} \tag{7.61}$$

$$\dot{\tilde{H}}_z|_{z=-d} = 0 \tag{7.62}$$

The last boundary condition may not be immediately apparent. The perfect electric conductor at $z = -d$ imposes a zero tangential *electric* field condition, implying zero *normal* derivative of magnetic field.

Table 7.1 *Values of the amplitude coefficients U_i and L_i associated with the upper and lower parts of the layer containing the source (after [1, Table 1, p. 150])*

	G_H^{zx}	G_H^{zy}	G_E^{zx}	G_E^{zy}	G_E^{zz}
U_i	$-jk_y/4\pi u_0$	$jk_x/4\pi u_0$	$-jk_x/4\pi j\omega\epsilon$	$-jk_y/4\pi j\omega\epsilon$	$k_\rho^2/4\pi j\omega\epsilon u_0$
L_i	U_i	U_i	$-U_i$	$-U_i$	U_i

The above are for the source-free case. In Section 7.3, the effect of the source was introduced via a boundary condition. Here, we will introduce another method of dealing with this. For a layer with a source inside it, this can be taken into account by adding a solution ψ^∞, which is the particular solution corresponding to the source embedded in an unbounded homogeneous medium. In the spectral domain, the solution can be written as

$$\psi_i^\infty = \begin{cases} U_i\, e^{-u_i(z_i-D)} & D \le z_i \le d_i \\ L_i\, e^{+u_i(z_i-D)} & 0 \le z_i < D \end{cases} \tag{7.63}$$

for a source at $z_i = D$, with $z_i = z + d_i$ the local normal coordinate in each layer. The amplitude coefficients U_i and L_i depend on the physical quantity represented by ψ, and are tabulated in Table 7.1. (In the spectral domain, the transform of an HED of unit magnitude, $\delta(x)\delta(z=-D)$, is $1/2\pi$. The table takes this and other factors into account.) In the present case, this source will be located in the upper medium (free space) at $D > 0$; the limit case $D \to 0$ will be considered subsequently.

In the free-space region then, the solution is

$$\tilde{H}_z = a_0\, e^{-u_0 z} - \frac{jk_y}{4\pi u_0} e^{+u_i(z_i-D)}, \qquad \forall\, d \le z < D \tag{7.64}$$

in the region just above the interface, and for the rest of the region

$$\tilde{H}_z = a_0\, e^{-u_0 z} - \frac{jk_y}{4\pi u_0} e^{-u_i(z_i-D)}, \qquad \forall\, z \ge D \tag{7.65}$$

It is tempting to set D to zero and use this latter equation immediately, but it yields the incorrect solution.

We now apply the boundary conditions and eliminate the three unknown coefficients, a_0, a_1 and b_1. Application of Eq. (7.62) immediately yields $a_1 = 0$. Applying Eq. (7.60) for the non-magnetic substrate case ($\mu_r = 1$) in the limit $D \to 0$ yields

$$b_1 = \frac{a_0 - \frac{jk_y}{4\pi u_0}}{\sinh u_1 d} \tag{7.66}$$

Application of Eq. (7.61), again in the limiting case, gives

$$a_0 = \frac{-\frac{jk_y}{4\pi} + \frac{jk_y}{4\pi u_0} u_1 \coth u_1 d}{D_{\mathrm{TE}}} \tag{7.67}$$

where

$$D_{\mathrm{TE}} = u_0 + u_1 \coth u_1 d \tag{7.68}$$

The D_{TE} term (and a similar D_{TM} term, to be defined shortly) are written in this specific notation because they are linked to *surface waves*. These can be important as a mechanism both for loss, and for increasing coupling between elements in a microstrip patch array. Neither is usually desirable. We will return to this later.

The last coefficient, b_1, may now be obtained, and we find for the fields in the dielectric that

$$\tilde{H}_z = -\frac{jk_y}{2\pi \sinh u_1 d} \frac{1}{D_{\mathrm{TE}}} \sinh u_1 (z + d) \tag{7.69}$$

For the case where both source and observer lie on the air–dielectric interface, $z \to 0$ and this reduces to

$$\tilde{H}_z = -\frac{jk_y}{2\pi} \frac{1}{D_{\mathrm{TE}}} \tag{7.70}$$

What has now been computed is the spectral domain normal magnetic field due to an elementary x-directed dipole, i.e. $\tilde{G}_H^{zx}$. From Eq. (7.50), we find that

$$\tilde{G}_A^{xx} = -\frac{\mu \tilde{G}_H^{zx}}{jk_y} = \frac{\mu_0}{2\pi} \frac{1}{D_{\mathrm{TE}}} \tag{7.71}$$

The other components required for a HED may be derived in a similar fashion. The results are given in Table 7.2. Here, the subscript 1 has been dropped on u, since it clearly refers to the substrate. For convenience, the spectral domain parameters u and u_0 are also listed.

7.4.4 The scalar potential and the mixed potential integral equation

The third entry in Table 7.2 lists a term which requires a brief comment, viz. $\tilde{G}_V$. In Section 7.4.1, the usual "mixed potential" formulation, Eq. (7.25) (which is valid for $F = 0$) was presented. It is actually by no means obvious that the usual scalar potential,

$$V(\vec{r}) = \int_S G_V(\vec{r}, \vec{r}') q_s(\vec{r}')\, dS' \tag{7.72}$$

Table 7.2 *Spectral domain Green's functions for a single-layer grounded microstrip structure*

Sommerfeld potentials	
$\frac{2\pi\tilde{G}_A^{xx}}{\mu_0}$	$= \frac{1}{D_{\mathrm{TE}}}$
$\frac{2\pi\tilde{G}_A^{zx}}{\mu_0}$	$= \frac{jk_x(\epsilon_r-1)}{D_{\mathrm{TE}}D_{\mathrm{TM}}}$
$2\pi\epsilon_0\tilde{G}_V$	$= \frac{u_0+u\tanh ud}{D_{\mathrm{TE}}D_{\mathrm{TM}}}$
$D_{\mathrm{TE}} = u_0 + u\coth ud,$	$D_{\mathrm{TM}} = \epsilon_r u_0 + u\tanh ud$
$u^2 = k_\rho^2 - k^2,$	$u_0^2 = k_\rho^2 - k_0^2$

Both source and observer are on the air–dielectric interface (after [1, Table 2, p. 153]). k_0 is the wavenumber in free space, and k is the wavenumber in the dielectric.

can be extended to a layered medium under dynamic conditions. Fortunately, in the case of horizontal conducting surfaces, it can be shown that this is indeed valid, and further that the required scalar Green function is given in the spectral domain by [1, Section 3.3]

$$\tilde{G}_V = \frac{j\omega}{k_\rho^2}\left(\frac{\tilde{\dot{G}}_E^{zx}}{jk_x}\right) - \left(\frac{k}{k_\rho}\right)^2\left(\frac{\tilde{G}_H^{zx}}{jk_y\epsilon}\right) \tag{7.73}$$

for the Sommerfeld potentials.

Once the potentials are known, the fields can be computed from the potentials, as in Section 7.4.1. Before proceeding, it is worthwhile reminding the reader that the Green functions we have obtained are *spectral domain* representations; the spatial domain equivalents are of course defined by:

$$G_A^{xx}(\vec{\rho}\mid\vec{\rho}'=0) \equiv A_x(\vec{\rho}) = \frac{\mu_0}{2\pi}\int_0^\infty J_0(k_\rho\rho)\frac{k_\rho}{D_{\mathrm{TE}}}dk_\rho \tag{7.74}$$

$$G_V(\vec{\rho}\mid\vec{\rho}'=0) \equiv V(\vec{\rho}) = \frac{1}{2\pi\epsilon_0}\int_0^\infty J_0(k_\rho\rho)k_\rho\frac{u_0+u\tanh ud}{D_{\mathrm{TE}}D_{\mathrm{TM}}}dk_\rho \tag{7.75}$$

and these are the functions we require. Again, as a reminder, ρ is radial distance on the patch surface, $\sqrt{x^2+y^2}$; k_ρ is the integration variable; by convention, $z=0$ is the air–dielectric interface; and $J_0(x)$ is the Bessel function of the first kind of order zero

$$J_0(x) \equiv \frac{1}{\pi}\int_0^\pi \cos(x\ \sin\ \psi)\,d\psi \tag{7.76}$$

Note also that these are the Green functions for a source located at $\rho' = 0$; due to the translation symmetry, for sources located at a point other than the origin, all we need do is interpret the radial parameter as the distance from the observer to the source, i.e. $\rho = \sqrt{(x - x')^2 + (y - y')^2}$. This is also sometimes expressed as

$$G(x, y|x', y') = G(x - x', y - y'|0, 0) \tag{7.77}$$

Equipped with these Sommerfeld potentials, we can now write the mixed potential integral equation (MPIE) for the x-directed HED:

$$\vec{z} \times \vec{E}^{\text{inc}} = \vec{z} \times \left[j\omega \int_S \bar{\bar{G}}_A \cdot \vec{J}_S \, dS' + \nabla \int_S G_V q_S \, dS' + Z_S \vec{J}_S \right] \tag{7.78}$$

The vector potential $\bar{\bar{G}}_A$ and scalar potential G_V are as in the preceding section and are of course known, even if difficult to compute, as is the excitation $\vec{E}^{\text{inc}}$.

7.4.5 Surface waves

We commented earlier that the D_{TE} and D_{TM} terms are written in this specific form since they can be interpreted as *surface waves*. It can be shown that these expressions are the characteristic equations for the surface waves of, respectively, TE and TM waves propagating in a dielectric layer backed by a perfect conductor [1, Section 6]. Surface waves can decay as slowly as $1/\sqrt{\rho}$, and hence can be an important coupling mechanism between patches in a microstrip patch array. In the integrals required to compute the spatial domain Sommerfeld potentials, Eqs. (7.74) and (7.75), these enter in the denominator of the integrand, and zeros in D_{TE} and D_{TM} hence represent poles in the kernel, complicating the integration process. Fortunately, if $k_0 d\sqrt{\epsilon_r' - 1} < \pi/2$, then D_{TE} has no zeros and D_{TM} has only one, corresponding to the dominant zero-cutoff TM surface wave. This condition is equivalent to the restriction:

$$f[\text{GHz}] \leq \frac{75}{d[\text{mm}]\sqrt{\epsilon_r' - 1}} \tag{7.79}$$

For practical substrates, this condition is generally satisfied over most of the microwave band. Only in the case of a thick substrate of high dielectric constant need one be concerned with this requirement.

The position of the pole is also required for the integration process. For lossless substrates, the pole is real ($k_\rho = \lambda_{p0}$) and lies inside the segment of the

real axis $1 < \lambda_{p0}/k_0 < \sqrt{\epsilon_r}$. For thin substrates, an approximation of its position is [1, Section 6]:

$$\lambda_{p0}/k_0 \approx 1 + (k_0 d)^2 \frac{(\epsilon_r - 1)^2}{2\epsilon_r^2} \tag{7.80}$$

This expression also holds for low-loss substrates, although the pole then migrates below the real axis, as in Figure 7.3:

$$\begin{aligned} \lambda_p &\approx \lambda_{p0} \\ \nu_p &\approx (\epsilon_r' - 1) \tan\delta \left(\frac{k_0 d}{\epsilon_r'} \right)^2 \end{aligned} \tag{7.81}$$

7.5 Evaluating the Sommerfeld integrals

7.5.1 Approximate evaluation of the Sommerfeld integrals

In general, the semi-infinite integrals in the spatial domain Sommerfeld potentials, Eqs. (7.74) and (7.75), have no closed-form solution and numerical evaluation, the topic of this section, is required. In certain cases, however, approximate solutions can be used, and one useful one in the present context is for the magnetic vector potential A_x for the HED case. Equation (7.74) does not contain the TM pole, with the result that the vector potential can be approximated by the vector potential for the homogeneous region $\epsilon_r = 1$. (Physically, the argument is that this is the *magnetic* vector potential, which should not be much affected by thin dielectric sheets.) In this case, the approximation is

$$\frac{4\pi}{\mu} A_x = \frac{e^{-jk_0 R_0}}{R_0} - \frac{e^{-jk_0 R_1}}{R_1} \tag{7.82}$$

with $R_0^2 = \rho^2$ and $R_1^2 = \rho^2 + (2d)^2$. The latter is of course the distance from the image of the HED in the ground plane, and we recognize this expression as that of a dipole and its (reversed) image. Although not generally valid, this is a useful approximation, especially for thin substrates of moderate dielectric constant. Although an approximation of the scalar potential is also available [1, Section 7.2], it turns out to be far less useful in this case and will not be discussed here.

Before proceeding further, the very important point must be made that the techniques to be discussed here emphasize simplicity, frequently exploiting knowledge of the specific problem: for instance, we restrict the analysis to the case of a single pole, and concentrate largely on the lossless substrate case. General-purpose

programs using the Sommerfeld potentials have to handle potentially far more complex problems, and research still continues on efficient and robust implementations.

A mathematical aside – integration on the complex plane

The Sommerfeld integrals involve integration on the complex plane, $k_\rho = \lambda + j\nu$ in the present context, or more usually $z = x + iy$ in mathematical notation which we will use in this brief note. A few refreshers might be useful here. Firstly, a function $f(z)$ is *analytic* (or *regular*) in a region of the complex plane if it has a unique derivative at every point of the region. This is a far stronger condition in the complex plane than on the real line, since an analytic function has derivatives of *all* orders. (Many real functions have only derivatives to a certain order.) The *Cauchy–Riemann* conditions can be used to test whether a function is analytic in a region. A singularity is a point where $f(z)$ is not analytic; in the present context, it usually corresponds to an infinite value of the function.

Cauchy's theorem, and the resulting integral formula, are crucial: the theorem states that on a closed contour[a] C:

$$\oint_C f(z)\,dz = 0$$

provided that the function is analytic on and inside C.

A very important consequence of this is that if $C = C_1 + C_2$, then $\int_{C_1} f(z)\,dz = \int_{C_2} f(z)\,dz$. This is so important in the context of the Sommerfeld potentials that it is worth reiterating: provided that the function is analytic, *different integration paths between two points in the complex plane yield the same result*.

Cauchy's integral formula states that under the same limitations as above, the value of $f(z)$ at $z = a$, a inside C, is given by

$$f(a) = \frac{1}{2\pi i}\oint_C \frac{f(z)}{z-a}dz$$

We usually apply this in reverse: for a function analytic except for a simple pole at $z = a$, the above theorem permits us to evaluate the integral. Combined with Laurent's theorem, this produces the residue theorem, which states that for

[a] There are some limitations on the form of C – it must not cross itself, and only a finite number of corners are permitted.

isolated singularities within C,

$$\oint_C f(z)\,dz = 2\pi i \,\Sigma_k R_k$$

where R_k are the residues of $f(z)$ inside C. We will discuss finding the residues subsequently.

7.5.2 Numerical integration in the spectral domain

The spatial domain Sommerfeld potentials, Eqs. (7.74) and (7.75), require integration over the real positive axis λ.[3] We also note that since the integration is in the complex plane, the theory of complex functions permits deformation of the integration path, and a number of approaches avoid the pole(s), deforming the integral into the first quadrant. (The reason that the deformation takes this route is as follows. As already noted, for a lossy dielectric, the pole lies below the real axis, and the integration (along the real axis) lies above it. In the limit, as the loss tends to zero, the integration path must remain above the pole.) However, the most straightforward approach for the case of a simple pole is to to integrate along the real positive axis and this is the approach discussed here. There are, however, two points along the axis that require special care – the branch cut and the pole – and an asymptotic case needing caution.

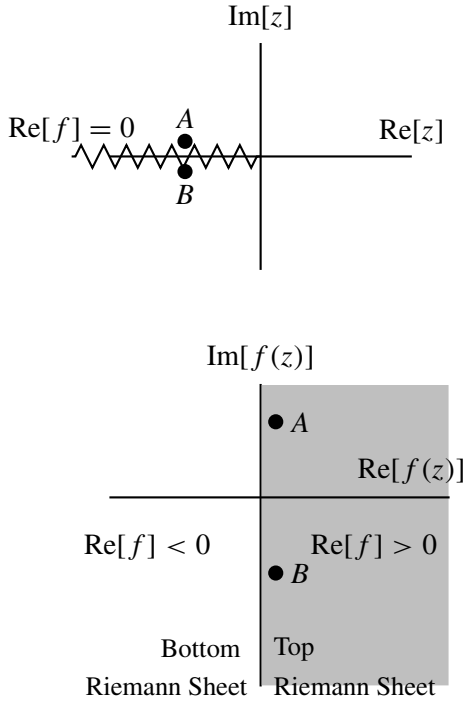
Firstly, at $k = k_0$, the function $u_0 = \sqrt{k_\rho^2 + k_0^2}$ introduces a *branch point*. This is due to the multi-valued nature of the complex valued square root function. Which value to choose is mathematically described as the process of selecting the correct *Riemann sheet*. Fortunately, all we need note here is that we should choose $\mathrm{Re}[u_0] \geq 0$; since the integrand remains bounded at this point, we can integrate straight through the branch point.

A mathematical aside – branch points and branch cuts

Branch points and cuts arise due to multi-valued functions in the complex plane. The branch cut is used to demarcate "Riemann sheets," which resolve the ambiguities. As a simpler example, consider $f(z) = z^{1/2}$. Obviously, with $z = Ce^{i\theta}$, $f(z) = \sqrt{C}e^{i\theta/2}$. This is periodic, but with period 4π, and this is where the problems arise. For instance, consider $\theta = 3\pi/2$ and $\theta = -\pi/2$, the *same* point on the complex plane. Now, the two solutions for $f(z)$ are $\sqrt{C}e^{i3\pi/4}$ and $\sqrt{C}e^{i\pi/4}$, clearly *not* the same point anymore.

[3] Once again, readers are reminded that in this context, $\lambda = \mathrm{Re}[k_\rho]$. Since we will continue to use λ_0 as the free-space wavelength, the potential for confusion is present, but we follow the notation of the literature in this context.

Riemann sheets adopt some convention to resolve this ambiguity. In this case, $f(z)$ for $-\pi < \theta < \pi$ is associated with the "top" Riemann sheet, and $f(z)$ for $\pi < \theta < 3\pi$ with the "bottom" Riemann sheet. This is best illustrated as below:



The negative real axis forms the *branch cut* in the z-plane, which opens up to define the boundary between the Riemann sheets in the $f(z)$ plane. By alternating between Riemann sheets, the function $f(z)$ can be made continuous. For instance, as one moves from $\theta = \pi^-$ (on the top Riemann sheet) to $\theta = \pi^+$, one must move onto the *bottom* Riemann sheet, which effectively resolves the ambiguity of which value of $\sqrt{-1}$ to choose, since we now know we must use π^+ and not $-\pi^-$ when evaluating the function with this convention. In this case, there were only two Riemann sheets. Other multi-valued functions, such as $\ln z$, can have infinitely many values and require an infinite number of Riemann sheets.

Which Riemann sheet one must work in the present context of Sommerfeld integrals often requires physical arguments, such as the radiation condition. This, and related issues, have caused many problems in the history of Sommerfeld potentials, with incorrect choices having led to unphysical artifacts and much debate in the literature. An extended discussion may be found in [8, 2.2].

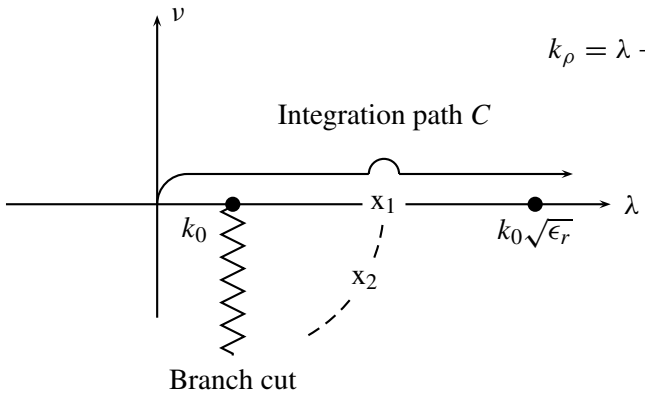


Figure 7.3 Topology of the complex plane for a thin grounded substrate, showing the branch cut, pole positions and the integration path C. For a lossless dielectric, the pole is on the real axis (x_1); when loss is present, it migrates into the fourth quadrant (x_2). (Adapted from [1, Fig. 5].)

The second point requiring attention is the pole, due to the TM surface wave. This introduces a rapidly varying integrand. Here, we follow [1, Section 8] and integrate through the pole (which lies on the real positive axis in the case of a lossless substrate), using a special method to extract the singularity which we will describe shortly. Note that for the HED, and assuming that the inequality of Eq. (7.79) holds (i.e. only the TM pole is present) it is *only* the scalar potential V which is thus affected.

The final point which one must bear in mind is that the oscillating integrands have an envelope which converges very slowly in the asymptotic case $\lambda \to \infty$. All these issues are summarized in Fig. 7.3.

In Fig. 7.4, the general properties of the function to be integrated are shown for a rather thick substrate with relatively large dielectric contrast; this has been done for clarity, to separate clearly the pole and the branch point, which in many practical cases lie close to one another. This figure shows the integrand of the scalar potential, Eq. (7.75), written in the following as:

$$V(\vec{\rho}) = \frac{1}{2\pi\epsilon_0}\int_0^\infty F(\lambda)\,d\lambda \tag{7.83}$$

$$\begin{aligned} F(\lambda) &= J_0(\lambda\rho)\,\lambda\,\frac{u_0 + u\tanh ud}{D_{\mathrm{TE}}D_{\mathrm{TM}}} \\ &= J_0(\lambda\rho)\,f(\lambda) \end{aligned} \tag{7.84}$$

where we have used $k_\rho = \lambda + j\nu$ since the integration is on the real axis.

It has been proposed [1, Section 8] that the real axis be split into three subintervals, namely $[0, k_0]$, $[k_0, k_0\sqrt{\epsilon_r}]$ and $[k_0\sqrt{\epsilon_r}, \infty]$, and we will follow this

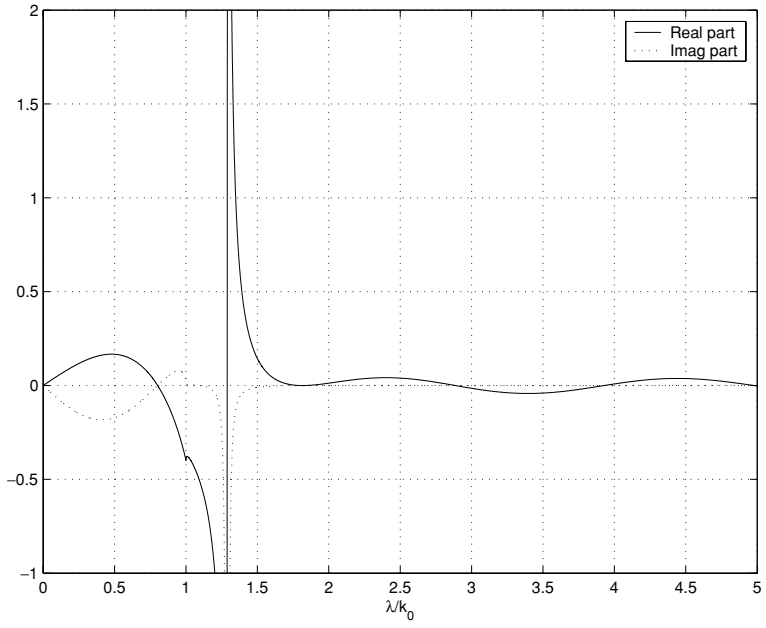


Figure 7.4 Properties of integrand associated with the scalar potential V for an HED. Parameters as for [1, Fig. 11]: $\epsilon_r' = 5$; $k_0 d = 0.2\pi$; $k_0\rho = 3$; $\tan\delta = 0.01$. Note the omission of π in the expression for $k_0 d$ [1].

approach here. We will investigate only the scalar potential V, since as mentioned above, the vector potential does not contain the TM pole and can be approximated using Eq. (7.82) for the case we will study. In each region, we proceed as follows.

Region 1 $[0, k_0]$

No special care is needed in this region, since the function is well behaved, apart from an infinite derivative at $\lambda = k_0$. A change of variables $\lambda = k_0 \cos t$ suffices to make the function very smooth and easily integrated using standard procedures. Hence, in region 1, the integral to evaluate numerically is:

$$\int_0^{\pi/2} F(k_0 \cos t) k_0 \sin t \, dt \tag{7.85}$$

Note that the minus sign present in the differential $d\lambda = -k_0 \sin dt$ is cancelled by the interchange of the lower and upper limits of integration required.

The numerical integration in this and all the remaining regions can be performed in MATLAB using the `quad` function, which implements adaptive Simpson quadrature. (Simpson quadrature, the classic numerical integration routine, fits a

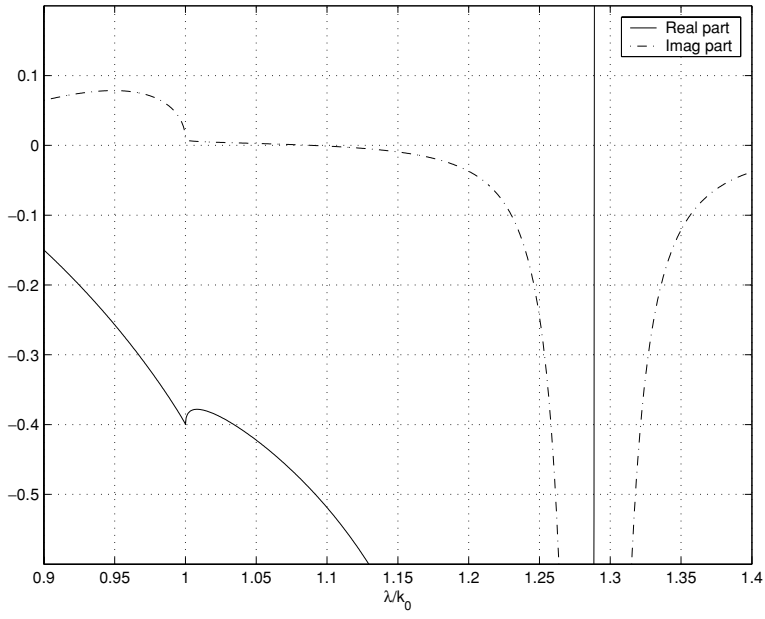


Figure 7.5 Detail of Fig. 7.4 in the region $k_0 \in [0.9k_0, 1.4k_0]$.

quadratic polynomial to the data points to be integrated; due to symmetry, it is exact to third order. The adaptive variant recursively divides the intervals until the difference between successive evaluations is less than some specified tolerance.) Many other types of numerical integration are available and can be applied; see, for instance, [9, Chapter 4] for an especially entertaining discussion.

Region 2 $[k_0, k_0\sqrt{\epsilon_r}]$

In this region, enlarged in Fig. 7.5, the singularity caused by the pole is clearly present. Strictly speaking, with finite loss this is a numerical singularity (or a quasi-singularity), since the pole is now slightly below the real axis and the value of the function is not truly infinite at the pole; however, for practical situations with low-loss substrates, the values are numerically so large that the effect is that of a singularity; furthermore, for a lossless substrate, this is a true mathematical singularity.

The approach used here is widely used for dealing with singular and quasi-singular integrands in integral equations. To the integrand is added and subtracted a function containing the singularity, whose integral can be evaluated analytically. In this case, the following is a suitable function:

$$F(\lambda) = \left[J_0(\lambda\rho)\, f(\lambda) - F_{\text{sing}}\right] + F_{\text{sing}} \tag{7.86}$$

where

$$F_{\text{sing}} = \frac{R}{\lambda - (\lambda_p - j\nu_p)} \tag{7.87}$$

Here $\lambda_p - j\nu_p$ is the complex pole (with $\nu_p > 0$) and R is the *residue* of the integrand at the pole. (We will discuss how to compute this shortly.) To simplify matters, we will limit ourselves to the case of a lossless substrate, hence the pole is on the real axis at $\lambda = \lambda_p$; the extension to the low-loss case is moderately straightforward, however. In this case, the integral in this region of the singular function may be found as [1, Eq. (110)][4]

$$I_s = \int_{k_0}^{k_0\sqrt{\epsilon_r'}} \frac{R}{\lambda - \lambda_p} d\lambda = R \ln\left(\frac{k_0\sqrt{\epsilon_r'} - \lambda_p}{\lambda_p - k_0}\right) - j\pi R \tag{7.88}$$

It is worth noting that this is the sum of the principal value (or Cauchy principal value) of the integral, and the contribution of the pole. (The principle value of a singular integral avoids the singularity.) The result for lossy materials is useful [1, Eq. (109)]

$$I_s = \frac{R}{2} \ln\left(\frac{v_p^2 + (k_0\sqrt{\epsilon_r'} - \lambda_p)^2}{v_p^2 + (k_0 + \lambda_p)^2}\right) + jR \arctan\frac{k_0\sqrt{\epsilon_r'} - \lambda_p}{\nu_p} + jR \arctan\frac{\lambda_p - k_0}{\nu_p} \tag{7.89}$$

In Fig. 7.6, the original function F, the singular function F_{sing} and the difference function have been plotted. The last is clearly smooth and readily integrated numerically. The smoothness has been enhanced by the change of variables $\lambda = k_0 \cosh t$. The integral in this region is the sum of I_s, the analytically integrated singular function as above, and I_d, the numerically integrated difference function:[5]

$$\begin{aligned} I_d &= \int_{k_0}^{k_0\sqrt{\epsilon_r'}} \left[F(\lambda) - F_{\text{sing}}\right] d\lambda \\ &= \int_0^{\operatorname{arccosh}\epsilon_r'} \left[F(k_0\cosh t) - F_{\text{sing}}(k_0\cosh t)\right] k_0 \sinh t \, dt \end{aligned} \tag{7.90}$$

One point that should be mentioned here is that for $\lambda = k_0\sqrt{\epsilon_r'}$, $u = 0$, and the $\coth ud$ term in D_{TE}, in the denominator of the integrand, results in a zero at this

[4] Note that this reference incorrectly includes the $j\pi R$ term, $j\pi P$ in their notation, on the left-hand side as well. Alternatively, the integral on the left-hand side should be a principal value integral.

[5] When performing the change of variables, recall that the derivative of $\cosh t$ is $+\sinh t$!

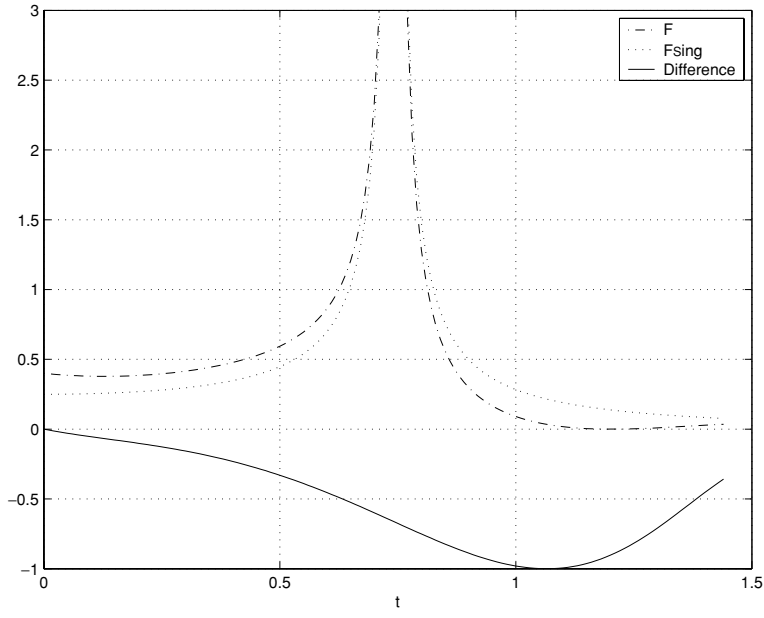


Figure 7.6 The original function (F), the singular function F_{sing} and the difference function, with the change of variables $\lambda = k_0 \cosh t$. All parameters as in Fig. 7.4, except that $\tan\delta = 0$.

point. Attempting to evaluate this numerically is inadvisable, and the upper integration limit should be set fractionally below this value. (Since this is a zero and not a pole, this simple remedy suffices.)

One final point requiring discussion is the evaluation of the residue. For a function of a complex variable z, with simple pole at $z = z_p$, which is the case we have here, the residue can be computed by multiplying the function by $z - z_p$ and evaluating the result at $z = z_p$. It is instructive to attempt this numerically, as shown in Fig. 7.7. The theoretical value is $R = 15.1107$; if the numerical result is interpolated through the pole, one will obtain a value very close to this. The reason that the curve in Fig. 7.7 exhibits a linear decay to zero in a small region around the pole is no doubt due to numerical approximations made (by MATLAB, in this case) when evaluating extremely large-valued functions.

The residue may be found rigorously noting that the integrand is of the form $g(z)/h(z)$, with $h(z_p) = 0$, but $h'(z_p) \neq 0$ and $g(z_p) \neq 0$. In this case, the residue may be computed from

$$R(z_p) = \frac{g(z_p)}{h'(z_p)} \tag{7.91}$$

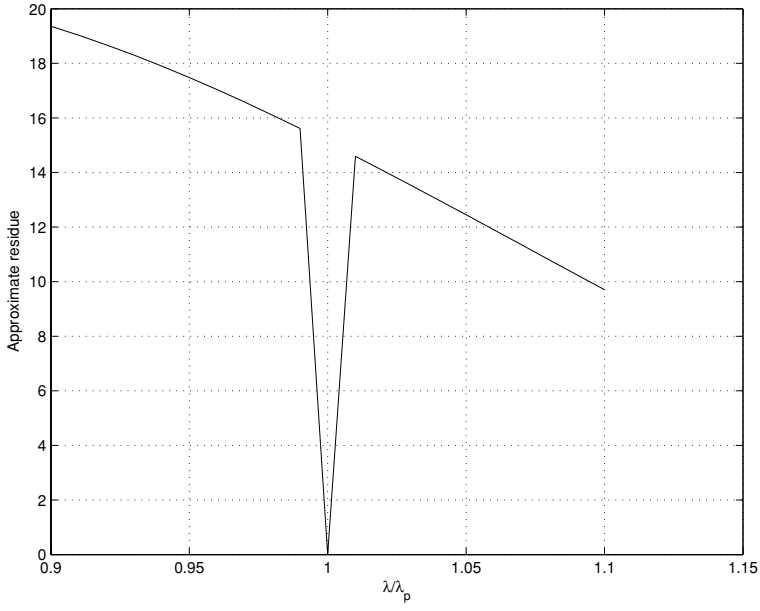


Figure 7.7 Result of attempting to evaluate the residue at the pole numerically. Parameters as in Fig. 7.6.

For the TM pole, the result is:

$$R(\lambda_p) = \frac{J_0(\lambda_p \rho)\lambda_p (u_0 + u \tanh ud)}{D_{\mathrm{TM}} \frac{d}{d\lambda} D_{\mathrm{TE}} + D_{\mathrm{TE}} \frac{d}{d\lambda} D_{\mathrm{TM}}} \tag{7.92}$$

with

$$\frac{d}{d\lambda} D_{\mathrm{TE}} = \frac{\lambda}{u_0} + \frac{\lambda}{u} \coth ud - \lambda\, d\, \mathrm{csch}^2 ud \tag{7.93}$$

$$\frac{d}{d\lambda} D_{\mathrm{TM}} = \epsilon_r \frac{\lambda}{u_0} + \frac{\lambda}{u} \tanh ud + \lambda\, d\, \mathrm{sech}^2 ud \tag{7.94}$$

In deriving this result, note that $du/d\lambda = \lambda/u$ and $du_0/d\lambda = \lambda/u_0$.

Region 3 $[k_0\sqrt{\epsilon_r}, \infty]$

In this region, the function has no singularities or branch points, but contains a slowly converging integrand, as shown in Fig. 7.8. To accelerate the convergence, the static term

$$\frac{J_0(\lambda\rho)}{1 + \epsilon_r}$$

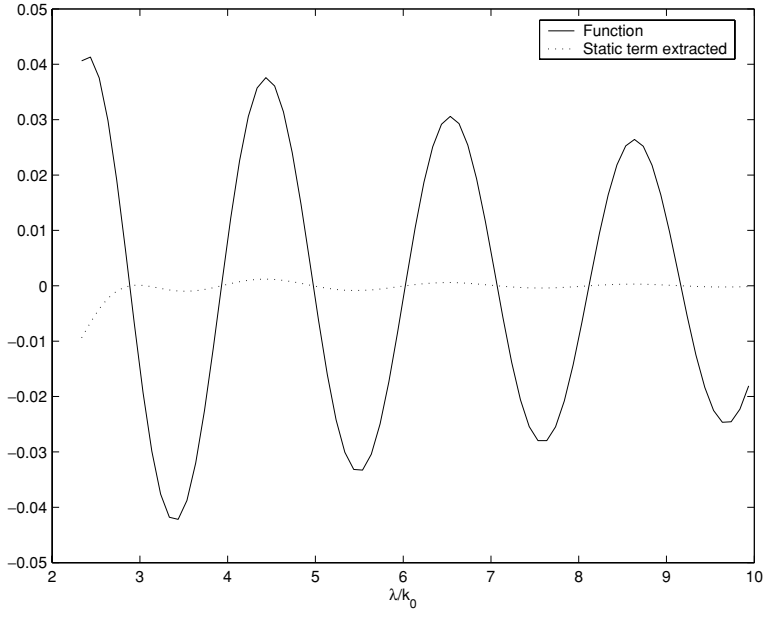


Figure 7.8 The integrand in region 3, before and after subtraction of the static term. Parameters as in Fig. 7.6.

is extracted. Beyond a certain point $\lambda > \lambda'$, the resulting integral is negligible. Using the standard result (for example, [10, Eq. 24.92])

$$\int_0^\infty J_0(\lambda\rho)\, d\lambda = \frac{1}{\rho}$$

one obtains

$$\int_{\sqrt{\epsilon_r'}k_0}^{\infty} F(\lambda)\, d\lambda \approx \int_{\sqrt{\epsilon_r'}k_0}^{\lambda'} \left[F(\lambda) - \frac{J_0(\lambda\rho)}{1+\epsilon_r} \right] d\lambda + \frac{1}{\rho(1+\epsilon_r)} - \frac{1}{1+\epsilon_r} \int_0^{\sqrt{\epsilon_r'}k_0} J_0(\lambda\rho)\, d\lambda \qquad (7.95)$$

The question of how large to set λ' can be determined iteratively. The results to be shown started with $\lambda' = 10k_0$; the resulting integral was evaluated, as well as the integral with $\lambda' = 20k_0$. The difference, normalized by the integral in region 2, between the integrals was then compared, and if too large, the procedure was repeated with the upper limits doubled. (The integral in region 2 is usually the largest contributor to the integral, since it includes the contribution of the pole, and hence was used to normalize this result.) This process is not especially robust, and more sophisticated procedures are available [1, Section 8.2].

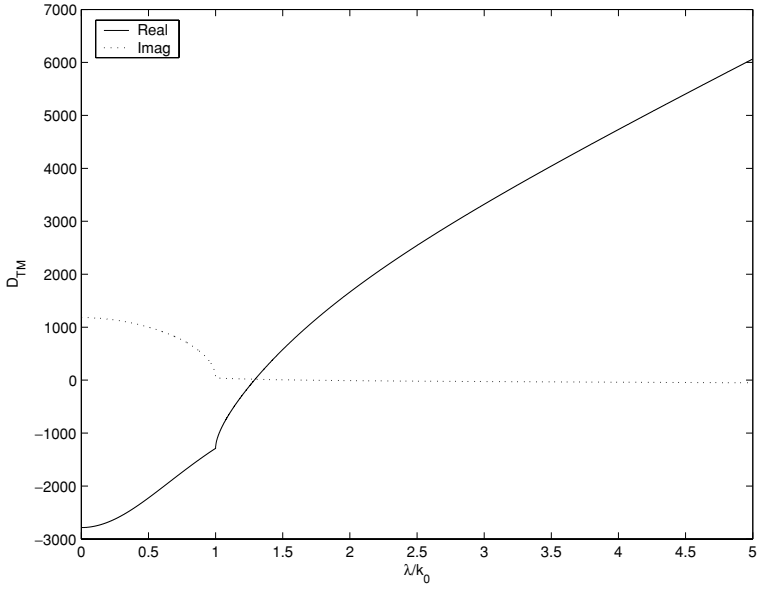


Figure 7.9 Plot of D_{TM}. Parameters as in Fig. 7.6.

7.5.3 Locating the pole

The position of the pole must of course be found with considerable accuracy for the above process to work properly, in particular in region 2. An approximation of its position has already been given in Eq. (7.80), but this is not sufficient for the singularity extraction procedure. Finding the pole is equivalent to locating the roots of D_{TM}. In general, finding the roots of a non-linear function is a very challenging problem, but in the case under consideration, the pole is known to be single, and located on the real axis in the interval $[k_0, \sqrt{\epsilon_r'}k_0]$. Furthermore, as Fig. 7.9 shows, the function is purely real valued for $\lambda > k_0$ (the branch point) and changes sign in this interval $[k_0, \sqrt{\epsilon_r'}k_0]$. A very simple algorithm, such as interval bisection, yields the root easily. Interval bisection starts with an interval containing a root, with the function having opposite signs at the interval limits. The function is then evaluated at the midpoint of the interval, which then replaces whichever limit has the same sign. This proceeds until the root is found with satisfactory precision. Despite its simplicity, the algorithm is failsafe in the present case – since it will always find at least one root, and there is only one. The method also converges linearly which is more than sufficient. The algorithm is so simple as not to require listing; details can be found in any book on numerical analysis, such as [9].

Slightly lossy materials can also be accommodated, although the root finder must now work with complex values; fortunately, although D_{TM} is now complex-

valued in the search interval, the overall shape of the function remains very similar to that of Fig. 7.9, and the imaginary part is small in the search interval.

Here we should comment that all the above holds *only* for the case of the single pole. As soon as more than one pole is present, the pole finding becomes far more complex. It is this type of complexity which makes robust, general-purpose codes so time-consuming to develop. Further details may be found in [1].

7.5.4 General source locations

The above potentials all assume that the source is located at $(x' = 0; y' = 0)$, i.e. $\rho' = 0$. For sources at other locations, all that is required is to substitute $\rho = \sqrt{(x - x')^2 + (y - y')^2}$. This is sometimes written as $V(\rho|\rho')$.

7.5.5 Some results for the Sommerfeld potentials

Now that the question of the integration of the potentials has been addressed, we can turn our attention to the potentials themselves. Results are shown in Figs. 7.10 and 7.11, which illustrate the variation due to different substrate thicknesses; note

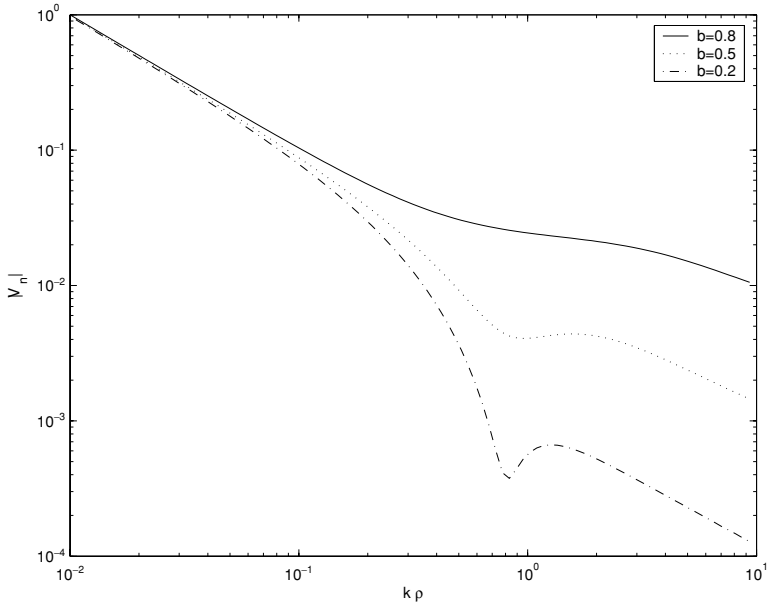


Figure 7.10 The modulus of the normalized scalar potential for various normalized thicknesses as a function of normalized distance. $\epsilon_r = 10$; $b = 2k_0 d\sqrt{\epsilon_r - 1}/\pi$.

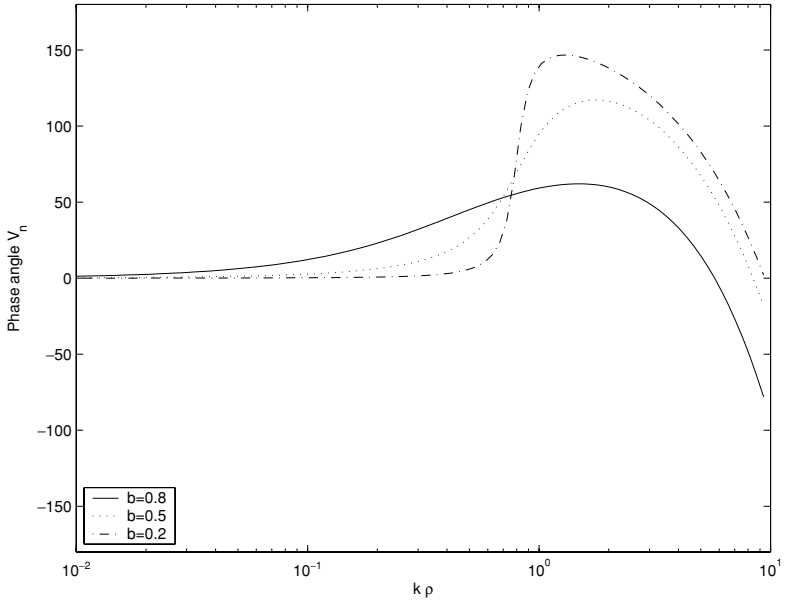


Figure 7.11 The phase of the scalar potential for various normalized thicknesses as a function of normalized distance. $\epsilon_r = 10$; $b = 2k_0 d\sqrt{\epsilon_r - 1}/\pi$.

that the results all converge for very small distances; this is the quasi-static limit. Figure 7.12 shows the effect of various dielectric constants. It is interesting to note the "knee" which sets in at progressively smaller distances as the dielectric constant increases; this corresponds to the transition from static to surface wave behavior. It will be noted that the potential decays at a slower rate once the surface wave sets in. It will also be noted that the surface wave is absent in the case of $\epsilon_r = 1.01$; this is essentially free space, which does not support a surface wave. (To avoid problems in the routines used, a value slightly larger than unity was used.) The effect of increasing dielectric constant has already been noted; for practical antenna design, this means that high-ϵ_r substrates are likely to have more problems with mutual coupling between array elements. The same effect is also present as the substrate thickness is increased.

These results are very similar to [1, Figs. 19–21] and serve to validate the implementation thus far.

7.6 MoM solution using the Sommerfeld potentials

Now that the potentials are available, the MoM discretization of the MPIE, Eq. (7.78), can be undertaken. Before we do this, it is useful to identify a suitable

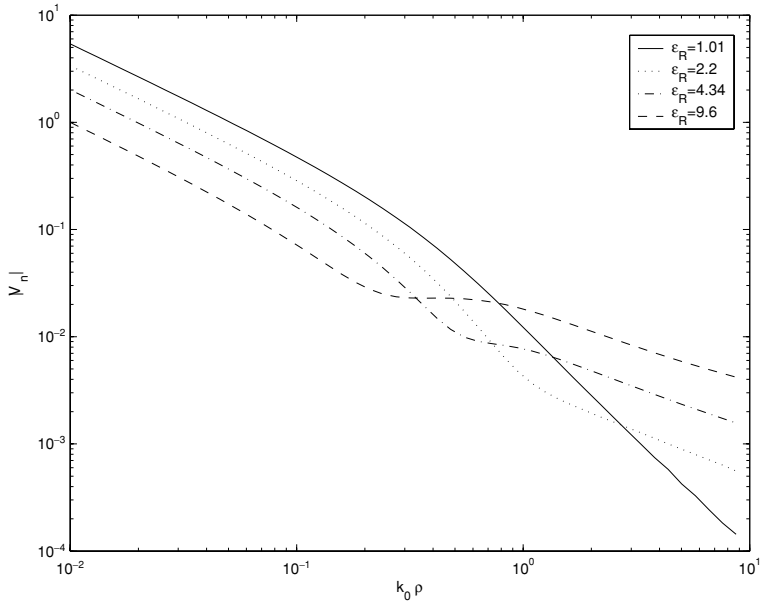


Figure 7.12 Effect of the dielectric constant on the scalar potential. $d/\lambda_0 = 0.05$.

problem. Although microstrip patch antennas[6] are the dominant application of this theory at present, they require a surface discretization, supporting vector currents (that is, the basis function must be able to support both $\hat{x}$- and $\hat{y}$-directed currents). A printed dipole is a rather easier problem, since so long as the dipole is relatively thin, the current flows essentially along the axis of the structure, much as for the thin dipole in free space that we have already studied in Chapter 4. A printed dipole is also easily simulated using a commercial code, such as FEKO.

With a suitable problem identified, various possibilities arise with the MoM. Perhaps the most popular, especially for "do-it-yourself" research codes, have been "rooftop" basis functions, defined on rectangular elements.[7] More sophisticated codes generally use the Rao–Wilton–Glisson element. For testing functions, Galerkin procedures have been widely used; another popular option has been a pulse-doublet testing function. Collocation techniques have also been used. We will take the opportunity to do something a little different (although also used in the literature), namely utilize *entire domain* basis functions. A very obvious one here is a Fourier series expansion; for a symmetrically excited dipole (e.g. center fed) only a cosine series is needed, and only the odd numbered terms.

[6] Readers not familiar with this technology should note that some more background on these antennas is presented in Chapter 8.

[7] The term *patch* instead of *element* is frequently encountered in the literature; the potential for confusion with the patch antenna is obvious and hence element is used here.

It is useful to develop the MoM equations from basic principles as another example of the application of the method. Referring back to the mixed potential integral equation, Eq. (7.78), repeated here, but with the last term dropped:

$$\vec{z} \times \vec{E}^{\text{inc}} = \vec{z} \times \left[j\omega \int_S \bar{\bar{G}}_A \cdot \vec{J}_S \, dS' + \nabla \int_S G_V q_S \, dS' \right] \tag{7.96}$$

the current is expanded as

$$\vec{J} \approx \sum_{m=1}^{N} \alpha_m \vec{F}_m \tag{7.97}$$

From the continuity equation, the charge is therefore expanded as

$$q_S \approx \sum_{m=1}^{N} \alpha_m \frac{-\nabla \cdot \vec{F}_m}{j\omega} \tag{7.98}$$

Note that the basis functions are effectively scalar in this case.

Introducing testing functions W_n and carrying out the weighted residual process as usual, we obtain:

$$\begin{aligned} \vec{z} \times \int_S \vec{W}_n \cdot \vec{E}^{\text{inc}} dS = \vec{z} \times \sum_{m=1}^{N} \alpha_m \Bigg[j\omega \int_S \vec{W}_n \int_S \bar{\bar{G}}_A \cdot \vec{F}_m \, dS' \, dS \\ - \frac{1}{j\omega} \int_S \vec{W}_n \cdot \nabla \int_S G_V \nabla' \cdot \vec{F}_m \, dS' \, dS \Bigg] \end{aligned} \tag{7.99}$$

One subtlety worth commenting on here is the manipulation of the second surface integral on the right-hand side of the above equation. Using the vector identity $\nabla(a\vec{b}) = a\nabla \cdot \vec{b} + \vec{b}\nabla a$, and identifying $\vec{b} = \vec{W}$ and a as the inner integral, one obtains

$$\begin{aligned} \vec{z} \times \int_S \vec{W}_n \cdot \nabla \int_S G_V \nabla \cdot \vec{F}_m \, dS' dS = \vec{z} \times \int_S \nabla \left[\vec{W}_n \int_S G_V \nabla' \cdot \vec{F}_m \, dS' \right] dS - \\ \vec{z} \times \int_S \nabla \cdot \vec{W}_n \int_S G_V \nabla' \cdot \vec{F}_m \, dS' dS \end{aligned} \tag{7.100}$$

The first term on the right-hand side in the above may be eliminated by applying a variant of the divergence theorem, known as the *surface divergence theorem*. For an open surface S bounded by contour C, this states that for a vector function $\vec{f}$

$$\int_S \nabla_s \cdot \vec{f} = \oint_C \hat{m} \cdot \vec{f} \, dC \tag{7.101}$$

with $\hat{m}$ the unit vector normal to contour C, but tangential to surface S [11, p. 712]. $\nabla_s \cdot$ is the divergence operator in the surface, and this is precisely what $\vec{z}\times$ selects.[8] Note that *unlike* Stoke's theorem, the contour integral in this case evaluates *normal* fields on the boundary. Hence this term can be written in terms of a contour integral of a quantity related to current, normal to the bounding contour. Since normally directed current should go to zero at the edge of the dipole, this term is zero. Strangely, few references on this topic explain this point.

The MPIE thus results in the standard MoM matrix equation $[Z]\{I\} = \{V\}$. For convenience, it is useful to split the impedance matrix in two

$$Z_{mn} = a_{mn} + v_{mn} \tag{7.102}$$

with matrix and vector entries as follows:

$$\begin{aligned} a_{mn} &= j\omega \int_S \vec{F}_m(\rho) \cdot \int_{S'} \bar{\bar{G}}_A \cdot \vec{F}_n \, dS' \, dS \\ v_{mn} &= \frac{1}{j\omega} \int_S \nabla \cdot \vec{F}_m(\rho) \cdot \int_{S'} G_V \nabla' \cdot \vec{F}_n \, dS' \, dS \\ b_m &= \int_S F_m(\rho) \cdot \vec{E}^{\text{inc}} \, dS \end{aligned} \tag{7.103}$$

For the case of a thin printed dipole, we will make a number of assumptions similar to those of our earlier work on the thin-wire dipole. It will be assumed that the current flows only in the $\hat{x}$-direction, and that the surface integrals can be approximated as line integrals. In this case, the integral in the transverse direction, $\hat{y}$, simply results in a constant W, present in both $[Z]$ and $[V]$, and thus cancelling. Further, the equations (7.103) can be rewritten in scalar form. The result is the following:

$$\begin{aligned} a_{mn} &= j\omega \int_\ell F_m(x) \int_{\ell'} A_x(|x - x'|) F_n(x') \, dx' \, dx \\ v_{mn} &= \frac{1}{j\omega} \int_\ell \frac{\partial}{\partial x'} F_n(x') \int_{\ell'} V(|x - x'|) \frac{\partial}{\partial x} F_n(x) \, dx' \, dx \\ b_m &= \int_\ell F_m(x) E_x^{\text{inc}} \, dx \end{aligned} \tag{7.104}$$

As already mentioned, we intend using entire domain basis functions. In this case, the source (primed coordinates) and field integrals are over the same domain, namely the length of the wire. Assuming that we center the wire at the origin,

[8] The surface divergence operator can be defined in terms of general curvilinear coordinates for curved surfaces, but in the present case it is unnecessary.

suitable entire domain basis functions are:

$$F_m = \cos\left(\frac{m\pi x}{L}\right), \qquad m = 1, 3, \ldots \tag{7.105}$$

Note that by this choice, the current goes to zero at the ends of the wire ($x = \pm L/2$) as required – for *all* the basis functions and hence also for their sum. With the above geometrical assumptions and these basis functions, and noting that the domain of integration is the same for both source and field points (the length of the wire), the matrix entries become:

$$\begin{aligned} a_{mn} &= j\omega \int_{-L/2}^{L/2} \cos\left(\frac{m\pi x}{L}\right) \int_{-L/2}^{L/2} A_x(|x - x'|) \cos\left(\frac{n\pi x'}{L}\right) dx'\, dx \\ v_{mn} &= \frac{1}{j\omega} \frac{mn\pi^2}{L^2} \int_{-L/2}^{L/2} \sin\left(\frac{m\pi x}{L}\right) \int_{-L/2}^{L/2} V(|x - x'|) \sin\left(\frac{n\pi x'}{L}\right) dx'\, dx \\ b_m &= \int_{-L/2}^{L/2} \cos\left(\frac{m\pi x}{L}\right) E_x^{\text{inc}}\, dx \end{aligned} \tag{7.106}$$

For the source, we will assume a very short feed section, of length Δs. The incident (impressed) electric field is thus V_s/Δ_s, where V_s is the source voltage. The result is that

$$b_m \approx V_s \tag{7.107}$$

It is interesting to note that the same result is obtained by assuming an infinitely thin Dirac delta source, with $E_x^{\text{inc}} = V_s \delta(x)$.

The code can now be developed. The integration required must be performed numerically. In this case, a simple trapezoidal scheme will suffice (implemented in MATLAB as `trapz`). An issue which requires a little care is that of singularities; both the vector and scalar potentials exhibit singularities at the origin. Fortunately, the singularities are of low order – this is one of the appealing features of the MPIE. The rigorous method for handling this extracts the singular component (which in both cases is the static limit), integrates this analytically and the remaining part is integrated numerically, in a fashion already applied in region 2 when evaluating the scalar Sommerfeld potential. This works very well for subdomain MoM methods and is relatively easy to implement, since it need only be applied to the "self" term; unfortunately, with entire domain basis and testing functions, it is rather more difficult to use. Because the singularity is of relatively low order, it can be side-stepped numerically, by using integration points for the field and source point integrals which are slightly offset from one another. If there are N equally spaced

integration points $\Delta = L/N$ apart, instead of sampling at

$$x_j = -L/2 + \Delta/2 + \Delta(j-1)$$

(and similarly for x'_k) one can use, for instance,

$$x_j = -L/2 + \Delta/3 + \Delta(j-1)$$

and

$$x'_k = -L/2 + 2\Delta/3 + \Delta(j-1)$$

which offsets the points by $\Delta/3$.

To keep things simple, it will also be assumed that the substrate is thin enough that the low-frequency approximation of the magnetic vector potential may be used, namely Eq. (7.82).

One other issue which requires attention is computational efficiency. Usually, the first implementation of a new method can be done with little regard for this. However, the Sommerfeld potentials are sufficiently time consuming to evaluate that if some thought is not given to this, even simple problems take far too long to solve. Because of the dependence on wavenumber, the potentials are frequency dependent, and nothing can be done about this. However, for a particular antenna geometry at a specific frequency, the potentials are only a function of radial distance ρ (and in this one-dimensional case, $|x - x'|$) and a widely used approach is to pre-compute the potentials and use interpolation when constructing the MoM matrices. This significantly reduces the time required to fill the impedance matrix.

Results for a MATLAB implementation are shown in Fig. 7.13. The printed dipole has length $L = 0.39\lambda_0$ and width $W = 0.002\lambda_0$, with relative permittivity $\epsilon_r = 2.55$, as in [12]. This dipole was designed as an element in a very large array, with λ_0 the free-space wavelength corresponding to the center frequency. For this simulation, this was chosen as 10 GHz, well into the microwave band and a typical frequency where microstrip is an attractive technology. (Because this is a single element, one can expect the actual center frequency to differ from this value; it turns out to be around 0.9 of the design value.) The substrate used in [12] is very thick (although only the TM mode propagates), and the approximation of the magnetic vector potential with its static value is insufficiently accurate, so the simulation here used a thinner substrate, $h = 0.12\lambda_0$ thick.

Figure 7.13 shows three results: one computed using FEKO ($h/\lambda = 50$ discretization), and two computed with a MATLAB code based on the formulation developed here. The "coarse" result was computed using only 1 mode, with 32 integration points; the "fine" result used 5 modes and 128 integration points. The reflection coefficient is computed in a $Z_0 = 50\,\Omega$ system. (It should be commented

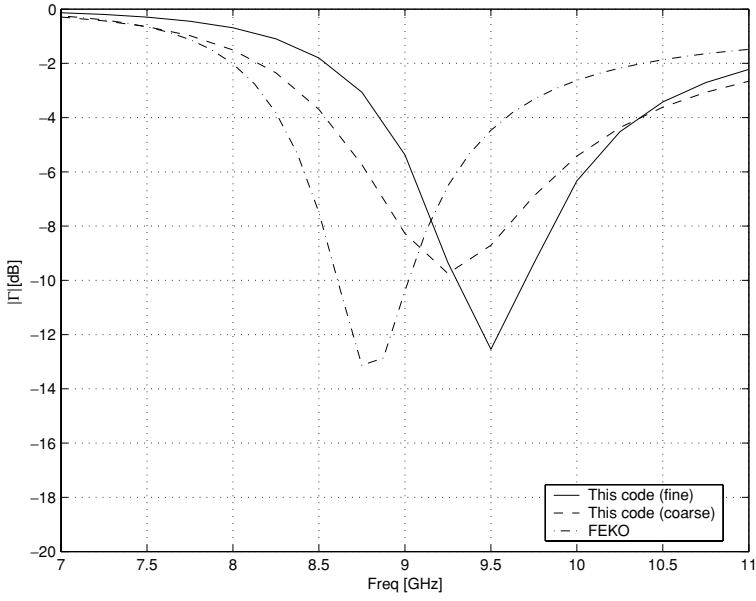


Figure 7.13 Reflection coefficient of a thin printed dipole.

that this antenna is not very well matched: the reason is that the substrate is not thick enough to provide sufficient spacing between the antenna and its image in the ground plane.) Improving the MoM model (the "fine" result) produces a value for minimum S_{11} very similar to the FEKO result, although at a frequency some 8% higher.

This is not very accurate – certainly not sufficient for engineering design purposes – but provides verification of our formulation and implementation. The aim of this section has not been to develop an accurate engineering tool per se, but rather to demonstrate the basic operation of the Sommerfeld approach and this has been achieved. Nonetheless, there are various things one could to to improve this scheme. Firstly, the magnetic vector potential should be implemented as a full Sommerfeld integral, rather than approximated by its low-frequency value as at present. Secondly, the integration scheme used in region 3 of the Sommerfeld integral would benefit from some refinement. Thirdly, the singularities in the MoM impedance matrices should be properly addressed; a subdomain MoM scheme might make this easier. We will not, however, pursue this here. There are sufficient problems remaining in this field that entire books can (and have) been written on this topic. Instead, we turn our attention in the next chapter to the use of a commercial package which has a very comprehensive implementation of this theory [13].

Coding hints – coping with complexity

The implementation of the Sommerfeld formulation discussed here is one of the most complex coding tasks in this book. The author's implementation used one main MATLAB m-file and some nine or ten functions, each of course in its own file. The total code ran to around 300 lines of MATLAB. This sounds quite modest, but MATLAB is particularly terse due to the vector nature of much of the code, implicit typing and high-level functions available (e.g. matrix solution, Bessel functions), so this would probably run to several thousand lines of code in languages such as FORTRAN, C, C++ or Java. How does one cope with the complexity that this brings? Here are some tips gleaned from twenty years of coding.

- Firstly, *start modestly*. Do not try to develop a general-purpose program from the start – unless of course this is one's job description. (Even then, one would be advised to code a simple implementation first, to learn the basics of the method if one is not familiar with it.) Writing general-purpose software is astonishingly difficult and time consuming, which is why good CEM software is not cheap.
- Secondly, *use existing packages* where possible. Writing an LU factorization routine is really unnecessary: there are industrial strength routines available in the excellent public domain LAPACK suite. Evaluating special functions is also more complex than it appears; books such as [9] offer routines[a] for Bessel functions, root finding etc.
- Thirdly, *use a proper scientific programming environment*. Writing one's own code for complex numbers is absurd – find an environment which supports this, or at least has proper libraries. Life is too short to code $a + jb$! By and large, computer scientists appear to prefer to ignore complex numbers, and it usually takes some time for whatever the latest fashionable programming language is to include this. This remains one of the strengths of FORTRAN – complex numbers are a built-in datatype. MATLAB is especially suited for the type of development discussed in this book, due to the very large number of high-level routines available, excellent support for complex numbers, and ease of graphing. For CEM coding, systematic, disciplined and modular work usually leads to far better code than supposedly state-of-the-art advances in languages.
- Fourthly, *modularity* is a key to successful code development. Whilst languages such as C++ have taken this concept much further with object orientation, the basic idea of this is common sense: test sections of the code independently as far as possible. It is much easier to locate the problem in the evaluation of, for instance, the scalar Green function in region 2 when this is implemented and tested separately, than to track this down as part of a complex code.

[a] Be warned that these are *not* public domain codes.

- Fifthly, *debug intelligently*. This is best discussed by anecdote. The key question is: what are the symptoms of the bug? Two which caused problems in the present case were: evaluation of a special function at a singular point (fortunately, MATLAB warns of this, and then it was just a question of locating the specific call and value of the argument); and overlooking the factor 2 in $R_1^2 = \rho^2 + (2d)^2$ in Eq. (7.82). The symptoms in the latter case were an incorrect reactance, which was traced to incorrect $[Z]$ elements; since the contributions from the scalar potential had already been validated, the error probably lay in the vector potential, and thus the bug was located. This latter case is an example of a strange phenomenon of bugs: they are frequently located in some part of the code which should be very simple. Perhaps it is human nature to concentrate on the hard tasks and pay insufficient attention to the straightforward ones?
- Finally, *validate* your code carefully. This is very important, and we have emphasized this on several occasions.

7.7 Further reading

The development in this chapter is largely based on that of Mosig [1]. A similar, although not quite as comprehensive, treatment may be found in [14], and most of the key equations are also available in this source. Both of these contain quite extensive lists of references for further reading. For the specific development of an MoM code for microstrip antennas using the Sommerfeld potentials, these are the key references, containing a wealth of detail of implementation issues. Another contemporary publication was the monograph by Hansen [15]; this is somewhat more general in scope, addressing not just microstrip structures, but also computational issues in detail.

The formulation as discussed in this chapter addressed only single-layer grounded lossy dielectrics. It can be be extended to include multi-layer substrates and superstrates, with conductors of finite conductivity, so complex microstrip antenna arrays can be accurately modelled; details may be found in [1, 14]. (The half-space problem can also of course be addressed – this was the subject of Sommerfeld's original investigations.) Microstrip antennas can be fed via feed pins, side feeds, or aperture coupling; the first two are readily implemented within the electric field MPIE MoM as in this chapter. It is possible to extend the formulation to include magnetic currents as well, which permits aperture coupling to be modelled efficiently.

For other, more general, treatments of stratified media, Chew's work is particularly lucid [8]. Chew takes a slightly different approach, developing the Sommerfeld integral as a sum (spectrum) of cylindrical waves, and using plane-wave theory to handle stacked layers. His treatment is oriented more at buried

antennas (or scatterers) than microstrip structures, reflecting a geophysical background. Ishimaru and Kong both provide coverage of these stratified media in their textbooks [16, 17]. (The transmission matrix formulation widely used for multilayered media was formulated by Kong in an earlier book.) The latter is especially concise, perhaps too much so for introductory reading. Again, the emphasis is on half-space problems rather than microstrip structures. None of these references considers the numerical evaluation of the integrals in any detail.

Work continues to be published on quite fundamental issues on this topic. Work on wires penetrating interfaces between different media was published by Burke and Miller [18] and was implemented in NEC-3 and NEC-4. An important generalization of this was Michalski and Zheng's work [19, 20], which permitted arbitrary conducting objects to penetrate the interfaces between dielectrics, using the RWG basis functions for the surface discretization. A very comprehensive invited review paper by Michalski on handling the "tails" of Sommerfeld integrals appeared quite recently [21]. Improved methods for efficient evaluation of the functions also continue to appear [22]. Some aspects of the extension of the MPIE discussed in this chapter to problems involving both electric and magnetic surface currents are discussed in [23]; an attractive feature of this treatment is that it permits very efficient modelling of slots in ground planes.

References

[1] J. R. Mosig, "Integral equation technique," in *Numerical Techniques for Microwave and Millimetre-wave Passive Structures* (T. Itoh, ed.), Chapter 3. New York: Wiley, 1989.

[2] J. Mosig and F. Gardiol, "General integral equation formulation for microstrip antennas and scatterers," *Proc. IEE (H)*, **132**, 424–432, December 1985.

[3] D. M. Pozar, *Microwave Engineering*. New York: Wiley, 2nd edn., 1998.

[4] E. Yamashita and R. Mittra, "Variational method for the analysis of microstrip lines," *IEEE Trans. Microwave Theory Tech.*, **16**, 251–256, April 1968.

[5] R. C. Booton, *Computational Methods for Electromagnetics and Microwaves*. New York: Wiley, 1992.

[6] J. Schwinger, L. L. DeRaad, K. A. Milton, and W.-Y. Tsai, *Classical Electrodynamics*. Reading, MA: Perseus Books, 1998.

[7] D. B. Davidson and J. T. Aberle, "Introducing students to spectral domain method of moments formulations," *IEEE Antennas Propag. Mag.,* **46**(3), 11–19, June 2004.

[8] W. C. Chew, *Waves and Fields in Inhomogeneous Media*. New York: van Nostrand Reinhold, 1990.

[9] W. H. Press, S. A. Teukolsky, W. Vettering, and B. R. Flannery, *Numerical Recipes in Fortran: the Art of Scientific Computing*. Cambridge: Cambridge University Press, 2nd edn., 1992.

[10] M. R. Spiegel, *Mathematical Handbook of Formulas and Tables*. New York: McGraw-Hill, 1968.

[11] J. Jin, *The Finite Element Method in Electromagnetics*. New York: Wiley, 2nd edn., 2002.
[12] D. M. Pozar and D. H. Schaubert, "Scan blindness in infinite phased arrays of printed dipoles," *IEEE Trans. Antennas Propag.*, **32**, 602–610, June 1984.
[13] J. J. van Tonder and U. Jakobus, "Full-wave analysis of arbitrarily shaped geometries in multilayered media," in *Proceedings of the 14th International Zurich Symposium on Electromagnetic Compatibility*, pp. 459–464. February 2001.
[14] J. R. Mosig, R. C. Hall, and F. E. Gardiol, "Numerical analysis of microstrip patch antennas," in *Handbook of Microstrip Antennas* (J. R. James and P. S. Hall, eds.) London: Peter Peregrinus (on behalf of IEE), 1989.
[15] V. W. Hansen, *Numerical Solution of Antennas in Layered Media*. Taunton: Research Studies Press, 1989.
[16] A. Ishimaru, *Electromagnetic Wave Propagation, Radiation and Scattering*. Engelwood Cliffs, NJ: Prentice-Hall, 1991.
[17] J. A. Kong, *Electromagnetic Wave Theory*. New York: Wiley, 1986.
[18] G. J. Burke and E. K. Miller, "Modeling antennas near to and penetrating a lossy interface," *IEEE Trans. Antennas Propag.*, **32**, 1040–1049, October 1984.
[19] K. A. Michalski and G. Zheng, "Electromagnetic scattering and radiation by surfaces of arbitrary shape in layered media, part I: Theory," *IEEE Trans. Antennas Propag.*, **38**, 335–344, March 1990.
[20] K. A. Michalski and G. Zheng, "Electromagnetic scattering and radiation by surfaces of arbitrary shape in layered media, part II: Implementation and results for continguous half-spaces," *IEEE Trans. Antennas Propag.*, **38**, 345–352, March 1990.
[21] K. A. Michalski, "Extrapolation methods for Sommerfeld integral tails," *IEEE Trans. Antennas Propag.*, **46**, 1405–1418, October 1998.
[22] J. R. Mosig and A. A. Melcón, "Green's functions in lossy layered media: integration along the imaginary axis and asymptotic behaviour," *IEEE Trans. Antennas Propag.*, **51**, 3200–3208, December 2003.
[23] M. Schoeman and P. Meyer, "On the structure and packing of the moment matrix in problems supporting simultaneous electric and magnetic surface currents," *Microwave Opt. Technol. Lett.*, **41**, 500–505, June 2004.

8

The method of moments and stratified media: practical applications of a commercial code

8.1 Printed antenna and microstrip technology: a brief review

Microstrip patch antennas are an example of a large class of modern antennas known as "printed antennas." Microstrip was originally developed in the early 1950s as a transmission line, and the first publication on using this structure as a radiator appears to have been by Deschamp in 1953 [1, Section 1.1]. Almost twenty years then passed until the first patent of the modern microstrip antenna was registered in 1973 by Munson, although the structure was independently discovered in at least one other location.[1]

Microstrip antennas are generally constructed using the same photo lithographic process using to create printed circuit boards. In their simplest form, radiation is due primarily to energy leaking out of the cavity formed by the patch located close to a ground plane; physically, the patch is simply a very wide microstrip line. For the basic rectangular patch, the radiation from two opposite sides reinforces, whereas that from the other two sides cancels. The patch is usually supported on a dielectric substrate of some form, primarily for structural reasons. Typical materials are Teflon and glass-reinforced plastics, as used in printed circuit board technology. Typical material properties for these are ϵ_r in the range from 2–2.5, and $\tan\delta$ from 0.0004–0.002. High-ϵ_r substrates such as alumina ceramics produce physically small patches, but with very limited bandwidth. Typical material properties in this case are: ϵ_r 9.7–10.3, $\tan\delta \approx 0.0004$. For some applications, plastic foam substrates have been used. These materials (sometimes using cheap materials such as expanded polystyrene tiles) have properties close to free space: $\epsilon_r \approx 1.05$, and $\tan\delta \approx 0.0008$.

[1] In 1972, at the National Institute for Defence Research, Council for Scientific and Industrial Research, Pretoria, South Africa. Unfortunately, the only references are internal classified memoranda and reports by C. A. van der Neut and A. Dubbelman.

Popular shapes are the original rectangular shape, which is still the most common, as well as square and circular patches. Patches are usually fed either from the side, typically using a microstrip line, or from below, using either a feed pin (usually the center pin of a coaxial cable) or aperture coupling. It is particularly easy to manufacture arrays using this technology (compared with wire antennas, for instance), since the corporate feed network can share the same substrate as the antenna. High-performance antennas usually split the feed network and the antenna onto two separate layers, to improve bandwidth and minimize unwanted radiation from the feed network. Even these are far easier to manufacture than a waveguide or wire array.

The main advantages of the technology are the following: it can be readily integrated with microwave circuitry; the antennas are flat, and can be conformed to surfaces, since the substrates can be moderately flexible; and it is at least potentially cheap, although high-quality substrates are not. The main drawbacks are limited bandwidth and power-handling capability. The former is the more serious problem in most applications and extensive research has focussed on the use of more complex geometries (doubled-stacked patches, for instance) in an attempt to increase this.

To read more about this class of antennas, the very comprehensive introductory discussion in [2, Chapter 14] can be recommended. Coverage is also available in [3, Section 5.8]. For serious designers, [1] is essential reading.

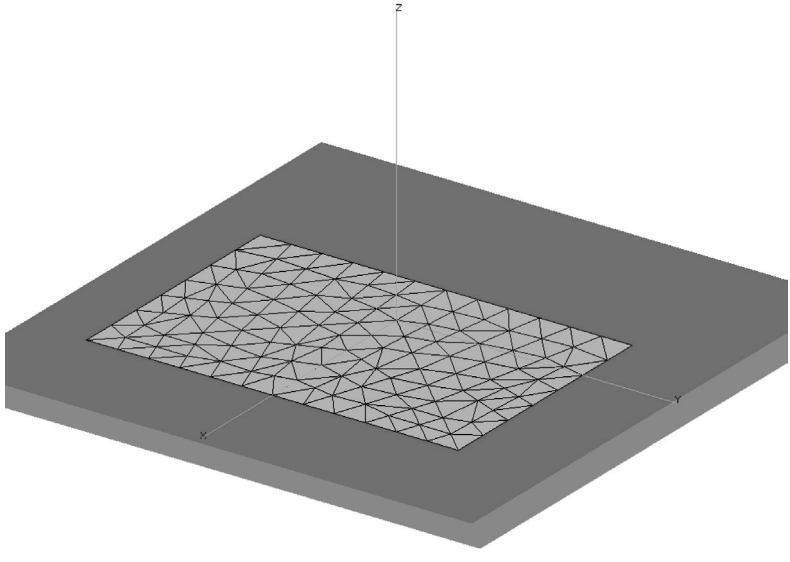


Figure 8.1 FEKO model of a rectangular patch antenna on a grounded substrate at $\lambda_d/15$ discretization.

8.2 A single patch antenna

In this example, a simple patch antenna is analyzed. The antenna is fed from below using an offset "feed pin" – this a quite typical arrangement. The offset is used to obtain matching; the patch has its highest impedance at the edges, and lowest impedance in the middle. A rectangular patch will exhibit two orthogonal resonances, at frequencies where the length or width corresponds to $\approx 0.48\lambda_d$ ($\lambda_d = \sqrt{\epsilon_r}\,\lambda_0$ is the wavelength in the substrate dielectric). In this case, the feed is offset in the x-direction, so the relevant resonance should be expected at about $\lambda_0 \approx \sqrt{\epsilon_r}\,2.08 \cdot 31.18$ mm, i.e. around 3.1 GHz. The geometry is illustrated in Fig. 8.1. It was generated using FEKO and is based on one of the examples shipped with the code.

Modelling hints – microstrip antennas

The PREFEKO model is shown in Fig. 8.2. A few points in this file require comment. Firstly, the feed pin must contact a node on the triangular mesh of the patch. This problem has been encountered before; the solution is explicitly to introduce a node on the patch at this point. (Once again, we comment that this is quite a general issue with MoM codes.) Half the patch is then generated using a triangle and a quadrilateral both of which include this feed pin node; the entire patch is then obtained by imaging in the $y = 0$ plane as usual (the feed pin lies on this plane of symmetry).

The properties of the substrate are defined using the `GF` card. Here, we use the planar multi-layer option 10, for a grounded single dielectric substrate. The other parameters are comprehensively described in the FEKO manual and do not require further comment.

Results for the reflection coefficient of the patch are given in Fig. 8.3, showing computations for both $\lambda_d/15$ and $\lambda_d/25$. (In Chapter 3, the antenna was also analyzed using the FDTD, and it was noted that the resonance was just under 3 GHz for a converged solution.) The antenna is well matched at 2.97 GHz. Compared to our simple estimate above, this is an error of around 4%, but it should be emphasized that that was a very crude approximation. The -10 dB impedance bandwidth is about 100 MHz, or 3%. A simple formula for the bandwidth of microstrip antennas predicts a bandwidth of around h/λ_0, which corresponds well with this result for this $h = 2.87$ mm thick substrate at $\lambda_0 \approx 100$ mm.

8.3 Mutual coupling between microstrip antennas

In a number of practical applications, radiating antennas are located sufficiently close to one another that significant amounts of energy couple between antennas.

```
** Example30a: A rectangular patch antenna on a dielectric substrate with
** a metallic ground plane (wire pin feed)

** Scaling factor since all dimensions below in mm
SF   1                          0.001
** Dimensions of the patch
#len_x = 31.18
#len_y = 46.75
** Feed location and wire diameter
#feed_x = 8.9
#diam = 1.3
** Substrate parameters
#h = 2.87     ** Height
#epsr = 2.2   ** Relative permittivity
** Frequency (for the discretisation)
#freq = 3.0e9
#lam = 1000 * #c0 / #freq / sqrt(#epsr)    ** Wavelength in mm
** Segmentation parameters
IP                              #diam/2   #lam/15   #lam/15
** Generate one quarter of the structure
** Define the points
#x = #len_x - #feed_x
DP   A                         -#feed_x   0.0       0.0
DP   B                          #x        0.0       0.0
DP   C                          #x        #len_y/2  0.0
DP   D                          0.0       0.0       0.0
DP   E                         -#feed_x   #len_y/2  0.0
DP   N                          0.0       0.0       -#h
** Patch
BT   D    B    C
BQ   D    C    E    A
** Symmetry to create the full structure
SY   1    0    3    0
** Feed wire with label 1
LA   1
BL   N    D
** End of geometry
EG   1    0    0    0    0

** Substrate (with groundplane)
GF   10   1              0              1.0       1.0
                               #h        #epsr     1.0

** Voltage source at feed point
A2   1    -1                   1.0       0.0       0.0       0.0       -#h
** Frequency loop in order to compute the impedance
FR   17   0                    2.8e9               3.2e9
** Change the line above as shown below to run with FEKO LITE
** FR   10   0                    2.8e9               3.2e9
** Just compute the impedance, no output of surface currents
OS   0
** Far-field pattern at centre frequency
FR   1    0                    3.0e9
FF   1    73   1    1         0         0         5
FF   1    73   1    1         0         90        5
** End
EN
```

Figure 8.2 PREFEKO file for the rectangular microstrip patch.

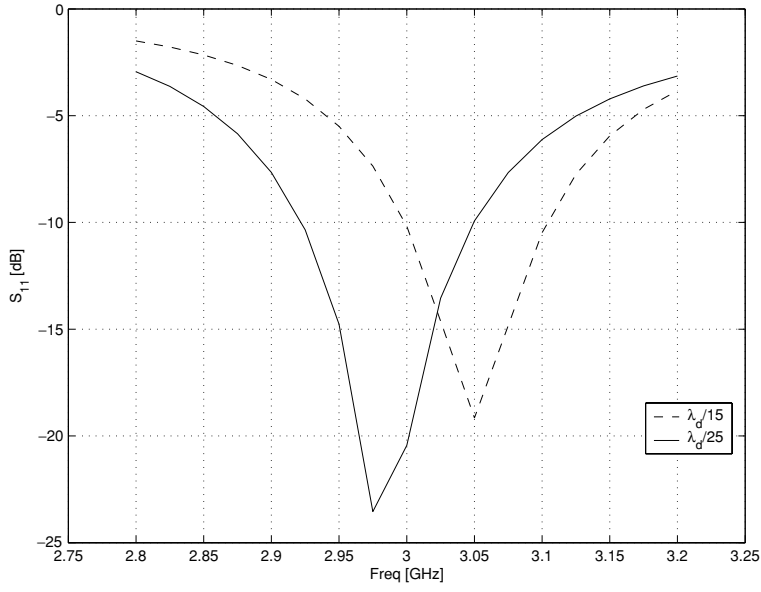


Figure 8.3 Reflection coefficient of the rectangular patch antenna for two discretizations.

This is known as *mutual coupling*.[2] In a typical antenna array, this is an important parameter to establish, since it determines the *active impedance* – also known as the driving point impedance. This is the impedance at each port of the antenna, taking into account mutual coupling from all the other antennas. In a simple two-element array, the formula is:

$$Z_a \equiv \frac{V_1}{I_1} = Z_{11} + Z_{12}\frac{I_2}{I_1} \tag{8.1}$$

If both elements are fed with equal amplitude and phase excitations (i.e. $I_2 = I_1$), the mutual coupling term Z_{12} adds to the self-impedance term Z_{12}. Alternatively, if the antennas are not part of an array, but connected to different RF systems, mutual coupling can result in undesired energy leaking between the systems. This leads into the field of radiated EMC.

Mutual coupling, in terms of voltage (or power) transfer, is complicated by possible mismatches at both transmitter and receiver. The general formula is quite complex, but if both antennas are well matched (in the same Z_0) then S_{12} (or S_{21}, which is identical in reciprocal systems) is the voltage transfer ratio. This can be

[2] Mutual coupling is used for two related, but not identical, physical parameters. In the one case, it refers to the mutual impedance or admittance. In the other, it refers to the energy coupled from one port to another. The specific usage is usually clear from the context.

seen from

$$V_1^- = S_{11}V_1^+ + S_{12}V_2^+ = S_{12}V_2^+|_{S_{11}=0} \tag{8.2}$$

Since microstrip patch antennas are frequently used in an array, it is an interesting exercise to compute the mutual coupling. We are fortunate in that good measured data are available [4]. Jedlicka *et al*. measured the mutual coupling between two patch antennas in both the E-plane (radiating edges adjacent) and the H-plane (non-radiating edges adjacent). The former results in far stronger coupling than the latter, so we will compute E-plane coupling. The elements were $L = 10.57$ cm (radiating edge) $\times$ $W = 6.55$ cm rectangular patch antennas. The substrate thickness was 0.1575 cm, with $\epsilon_R = 2.5$. The loss parameter $\tan\delta$ was not specified, and we will assume it was negligible. The measured resonance frequency was 1.410 GHz. The patches were pin fed. The feed point impedance at the edge of a patch is quite high, and can be reduced by moving the pin a distance x_0 from the edge. This feed-pin offset was not specified in the original article, but can be computed as follows. The maximum resistance is approximated by Munson's value:

$$R_m \approx 60\lambda_0/W \tag{8.3}$$

and the input resistance at feed point position x_0 in from the patch edge is

$$R_\in \approx R_m \cos^2\left(\frac{\pi x_0}{L}\right) \tag{8.4}$$

For this patch, $R_m \approx 195\,\Omega$ and $x_0/L \approx 0.33$ for a $50\,\Omega$ match. Since this is an approximate value, some fine-tuning is necessary with the simulation package to establish the optimal x_0/L as about 0.31. This produces a resonant frequency of $f_r = 1.425$ GHz, around 1% higher than the measured center frequency. Such differences are very common for narrowband structures; the most probable source of error is uncertainty of the exact value of ϵ_R, which is usually easily of this order unless very high quality (and hence expensive) substrates are used. Figure 8.4 shows the computed reflection coefficient. Results are given for both the isolated element case here, as well as the array case, with another patch one wavelength away (terminated in a matched load). As before, the predicted bandwidth of around 0.7% agrees quite well with the computed -10 dB bandwidth of just under 1%.

With the design of the basic patch finalized, the patch is replicated to generate another patch (Fig. 8.5).

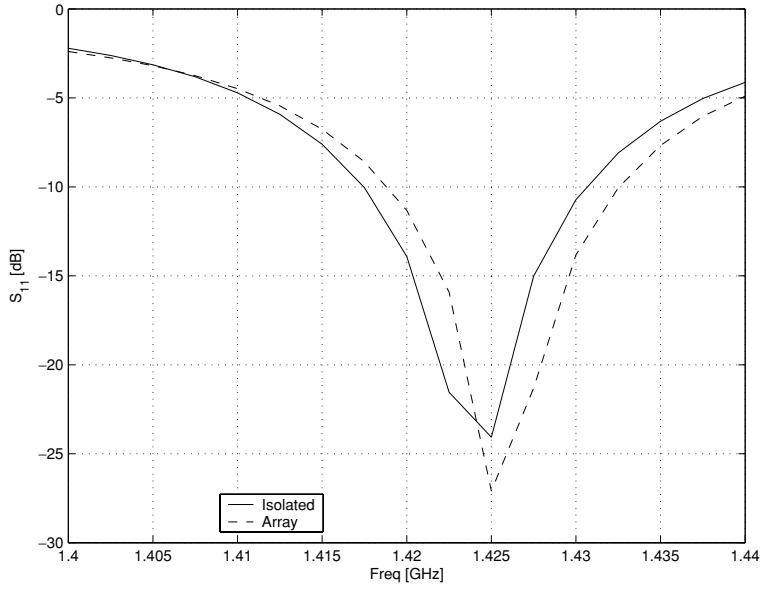


Figure 8.4 Reflection coefficient of the rectangular patch antenna in [4].

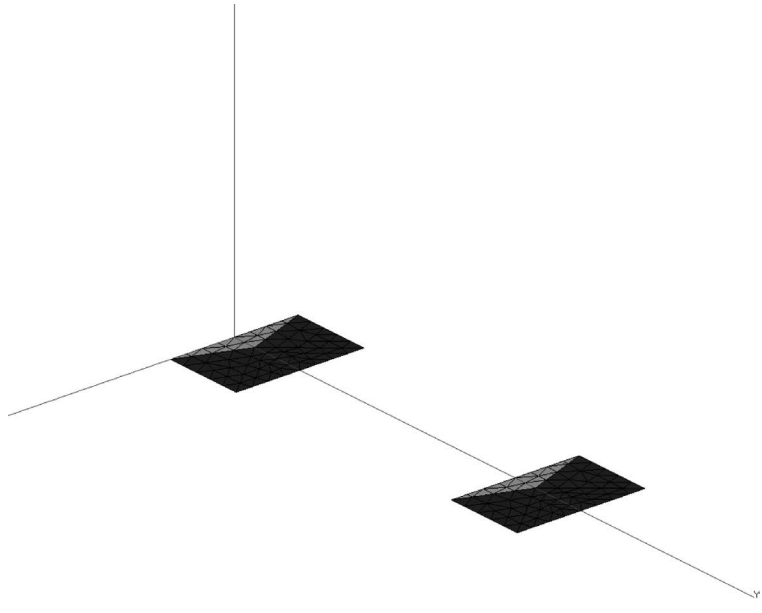


Figure 8.5 FEKO model of the two-element rectangular patch antenna array as in [4], for 1λ spacing. Only the patches are shown.

Modelling hints

As can be appreciated from the preceding material on the Sommerfeld potentials, the computational cost of this formulation is quite high. Symmetry should be exploited as far as possible to reduce this. For this E-plane coupling problem, there is a plane of magnetic symmetry in the plane containing the feed pins. Note that many codes support both geometrical modelling, which is largely a modelling aid, and does not usually significantly reduce computational cost,[a] as well as field (electric or magnetic) symmetry, which does. For this example, using symmetry correctly reduced the number of unknowns and the memory required by a quarter. Unfortunately, the H-plane coupling has *no* field symmetry, since the feed pin is offset in this plane.

Most simulation packages have the ability to copy parts of the geometrical model. In FEKO, this is done using the *translate geometry* facility.

[a] An exception is the present case of the Sommerfeld potentials, where using geometrical symmetry can speed up the matrix fill significantly.

Computing the mutual coupling is a little tedious; one specifies the inter-element spacing, runs the code at f_r, extracts S_{12} and then repeats the process for the next spacing. Results computed using FEKO with a $\lambda_0/15$ discretization are given in Fig. 8.6. (Note that the distance referred to here (and throughout this section) is the distance between adjacent edges, as in [4], rather than the inter-element spacing of array theory.) A convergence check was performed on the $D = 0.2\lambda_0$ case using a $\lambda_0/25$ mesh which confirmed that $\lambda_0/15$ is quite adequate. There are differences between the measured and computed data, at most around 2 dB, but this is to be expected. One reason for this discrepancy is the sensitive nature of this parameter. Figure 8.7 shows S_{21} as a function of frequency; clearly, very small changes in frequency can easily result in the type of discrepancy noted in Fig. 8.6, in either measurement or computation. Another possibility is the experimental setup, whereby dielectric spacers were inserted as the inter-element spacing increased; this is clearly only an approximation of a continuous substrate. Finally, data for the same problem computed by Mosig *et al*. [5, Fig. 8.27] also show differences of a similar type between measured and computed data, although in their case the agreement is better in some places and worse in others compared with our simulation. Their code used entire domain basis functions, so the numerical results cannot be expected to be identical.

For typical narrowband broadside patch array designs, the mutual coupling levels are relatively small, as we have seen, and may be neglected, a result which rather surprised antenna designers – who were used to the much higher levels of

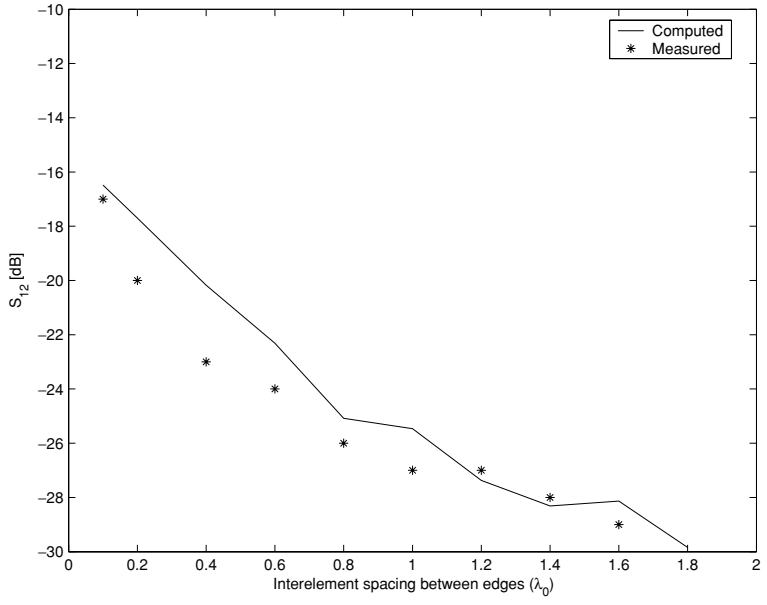


Figure 8.6 S_{12} for the rectangular patch antennas in the text. Measured data from [4].

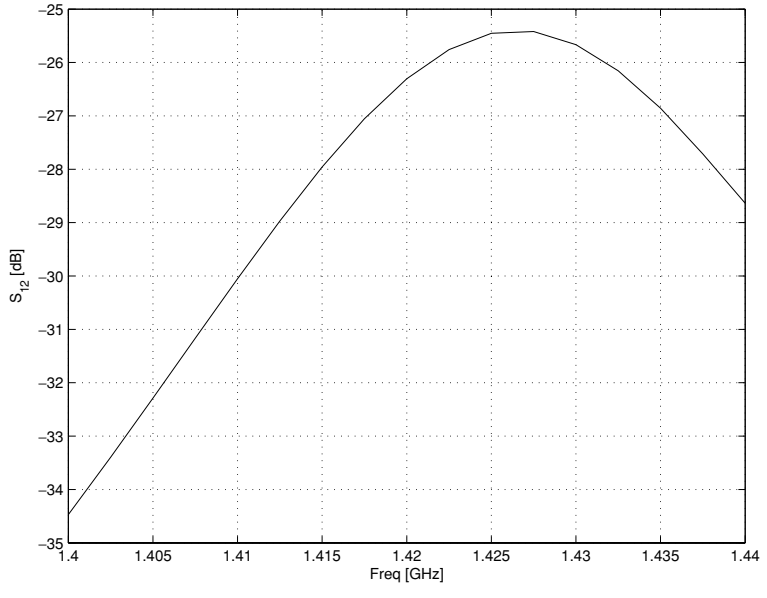


Figure 8.7 E-plane mutual coupling between two patches, one wavelength apart, showing strong frequency dependence.

mutual coupling in wire or slotted waveguide arrays – when microstrip patch arrays were first developed [6, p. 270]. This is *not* however true of arrays using thick substrates and/or high dielectric constants, since surface waves can be strongly excited, resulting in higher levels of coupling. It is also not true of phase scanned arrays, the topic of the next section.

8.4 An array with "scan blindness"

The elementary theory of phased arrays can be found in almost any book on antennas. By adjusting the relative phase between array elements, the position of the main lobe (and of course the side lobes) can be moved; if the phasing can be changed (either manually or electronically) the beam can be "steered." Phased arrays, as such antennas are called, were a crucial defense technology throughout the Cold War, with one of the most dramatic examples of the technology being the DEWS (Distant Early Warning System) radars deployed by the USA to warn of ICBM attack. More recently, "smart" antennas also exploit this effect, although usually to move nulls to cancel undesired signal sources rather than position main beams to detect targets.

In practice, however, arrays can exhibit an effect called "scan blindness," which few textbooks discuss, [3, p. 470] being an exception, since the effect is not predicted by simple antenna theory. Scan blindness occurs at a specific angle (or angles), and at this angle the antenna becomes extremely badly matched, radiating essentially no energy. Different types of arrays can suffer from this, including waveguide and wire arrays, and also printed arrays such as microstrip patch arrays. The common factor in the scan blindness phenomenon is a structure near or on the array face capable of supporting a slow wave; a slow wave is one whose phase velocity is much less than the velocity of light. (Classic examples are helices, corrugated surfaces and grounded dielectric slabs.) TM and TE surface waves have already been discussed, so it is not surprising that microstrip arrays can suffer from this. For printed antennas, two papers by Pozar and Schaubert [7, 8] are the key references, with a comprehensive exposition of the problem supported by results computed using one of the earlier MoM codes able to handle this type of antenna.

Strictly speaking, scan blindness only occurs in infinite arrays, but in sufficiently large finite arrays, the effect in practice is the same: a very poorly matched antenna which hardly radiates. It is possible to formulate the problem in the spectral domain to produce an infinite array [7, 8], but most commercial codes cannot do this. To demonstrate the effect, we will study a large array of thin printed strip dipoles. We use this structure, rather than patches, to permit a larger array to be simulated. Symmetry should also be used as far as possible to increase the effective array size;

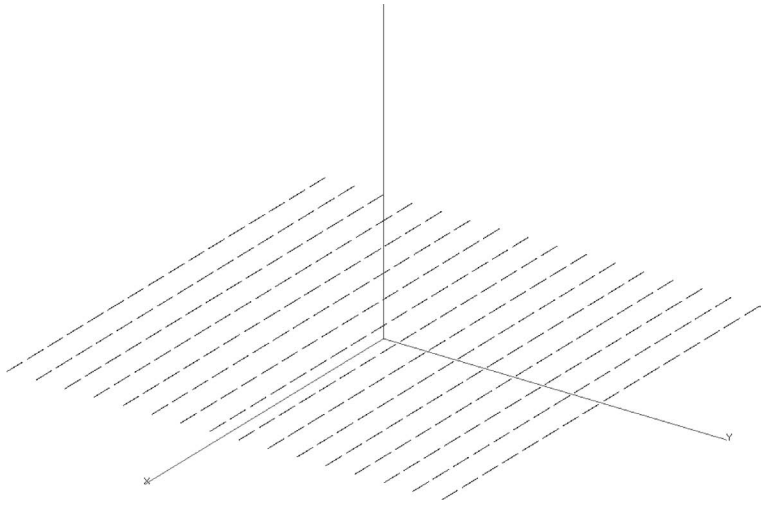


Figure 8.8 256 element printed dipole array.

unfortunately, the phasing of the feeds required to scan the array limits the use of symmetry.

An example of an array produced in FEKO is shown in Fig. 8.8. Each element is a strip dipole, length $L = 0.39\lambda_0$ and width $W = 0.002\lambda_0$, with substrate thickness $h = 0.19\lambda_0$ and relative permittivity $\epsilon_R = 2.55$, as in [7].

Modelling hints – generating a large array

Generating the array can be an exercise in programming; in FEKO, perhaps the simplest approach is to use two nested `FOR` loops, the inner loop generating the dipoles in the E-plane, the outer loop generating "lines" of these dipoles. The key loops are shown in Fig. 8.9; the variable #a is the inter-element spacing, and #N is the square root of the number of elements – the array is square. Similar ideas could also be used in other simulation packages supporting some form of scripting.

It is interesting firstly to study the effect of the array environment on the element. The concept of active impedance has already been introduced in Eq. (8.1) for two-element arrays. For an N element array, the active impedance of element i is

$$Z_{a_i} \equiv \frac{V_i}{I_i} = Z_{ii} + \sum_{j=1}^{N,\ j\neq i} Z_{ij}\frac{I_j}{I_i} \tag{8.5}$$

```
.
.
.
#yc = #a/2
!! FOR #j = 1 to #N/2 ** Outer loop
#xc = (-#N+1)/2*#a
!! FOR #i = 1 to #N   ** Inner loop
#lb = (2*#N)*(#j-1)+2*#i-1
** Generate the strip dipole antenna
DP   A                        #xc       #yc-#w/2  #h
DP   B                        #xc+#L/2  #yc-#w/2  #h
DP   C                        #xc+#L/2  #yc+#w/2  #h
DP   D                        #xc       #yc+#w/2  #h
LA   #lb
BP   A    B    C    D
DP   E                        #xc       #yc-#w/2  #h
DP   F                        #xc-#L/2  #yc-#w/2  #h
DP   G                        #xc-#L/2  #yc+#w/2  #h
DP   H                        #xc       #yc+#w/2  #h
LA   #lb+1
BP   E    F    G    H
#xc = #xc+#a
!!NEXT
#yc = #yc+#a
!!NEXT
SY   1    0    3    0    #N^2
.
.
.
** Set up array feeds.
!! FOR #k = 0 to 10 ** Start of phase angle loop
#thet = RAD(0+#k*5) ** scan angle theta in radians
#delfz = #k_0 * #a * sin(#thet)
#lb1 = 1
#lb2 = 2
** Impose progressive phase shift in voltage in E-plane (phi=zero)
!! FOR #j = 1 to #N ** Outer loop
#phs = 0 ** re-set phase to zero for each constant-y iteration
!! FOR #i = 1 to #N ** Inner loop
!!IF (#j = 1) and (#i=1) THEN
**   This is the first feed point, new feed (to zero all others).
AE   0    #lb1 #lb2 0         1.0       DEG(#phs) 75
!!ELSE
**   Additional feedpoints - add to sources.
AE   1    #lb1 #lb2 0         1.0       DEG(#phs) 75
!!ENDIF
#phs  = #phs + #delfz
#lb1 = #lb1+2
#lb2 = #lb2+2
!! NEXT
!! NEXT
```

Figure 8.9 Key components of the PREFEKO file used to generate the printed dipole array.

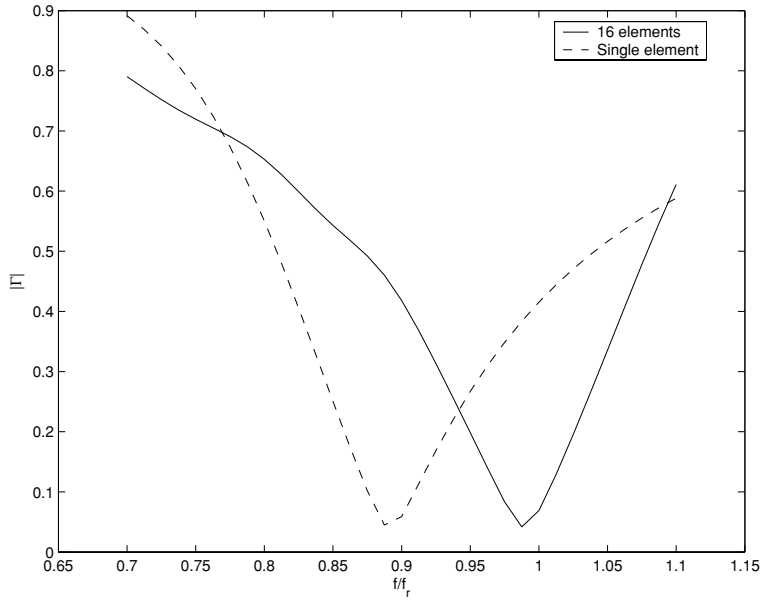


Figure 8.10 Reflection coefficient (in a $Z_0 = 75\ \Omega$ system) versus frequency for both an isolated element and a central element in a 16 element array.

When array feeds are used, this is the impedance automatically computed by codes such as FEKO – although it is not *explicitly* called the active impedance. In Fig. 8.10, the reflection coefficient of an isolated element is compared with that of an element in the center of a 16 element array. (The elements were discretized at around $\lambda_0/50$ for this figure.) Note that the resonance frequency moves upwards by around 10% due to the array enviroment. To compute this result, voltages of the same magnitude and phase were applied to each element. Note that this does *not* guarantee a uniformly illuminated array! The reason is that it is the *currents* which determine the radiation pattern, and since the active impedance differs from element to element, so does the resulting current.

Now, the effect of scan angle can be determined. For an $m \times n$ array of sources, to scan a beam an angle θ_s, ϕ_s off broadside requires that the m, nth source should be phased as

$$e^{jk_0(ma \sin\theta_s \cos\phi_s + nb \sin\theta_s \sin\phi_s)} \tag{8.6}$$

This assumes that the array axes are aligned with the x- and y-axes, as in Fig. 8.8, and that the spacing along these axes is a and b respectively. For reasons discussed in detail in [7], only the E-plane scan (the x–z plane in Fig. 8.8, i.e. $\phi = 0$) exhibits scan blindness, and our simulation will only investigate this plane of scan. We also assume that the inter-element spacings are equal in both planes, that is, $a = b$.

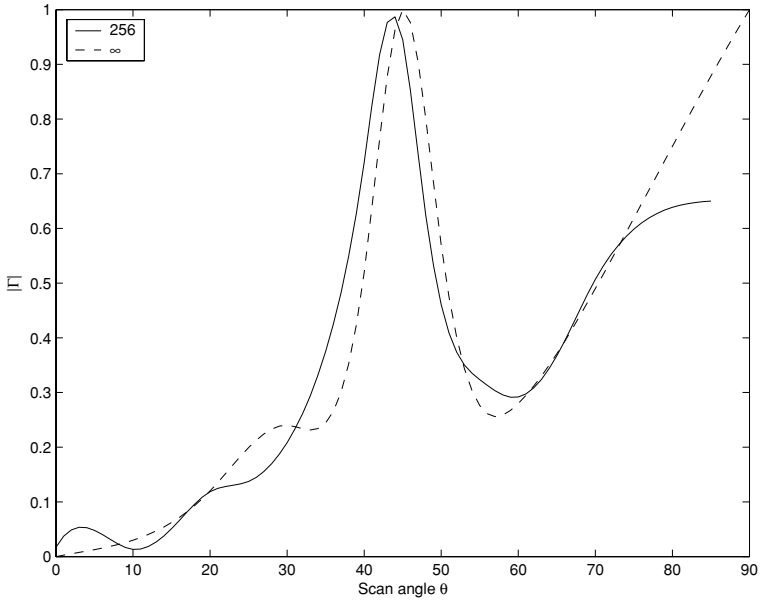


Figure 8.11 Reflection coefficient versus scan angle for a 256 element array and an infinite array; the latter data are from [7].

Hence the *progressive* phase advance (or delay) to add to each element in this plane is

$$\Delta = k_0 a \sin\theta_s \tag{8.7}$$

No progressive phase shift is required in the other plane.

The results of a simulation of a 256 element array are shown in Fig. 8.11. The inter-element spacing is $a = 0.5\lambda_0$ in both planes. Except where mentioned, the substrate was assumed lossless. These results were computed using a $\lambda_0/25$ discretization, which gave acceptable accuracy. For comparison, results computed using an infinite array code [7] are also plotted. The agreement is suprisingly good, and demonstrates how scan blindness can impact on a finite antenna which is of a quite practical size. (Although not shown, an 8×8 array gives a similar result, although the reflection coefficient peak is somewhat lower.) Note that, as in [7], all reflection coefficients for this array are referred to a $Z_0 = 75\ \Omega$ system.

Radiation patterns for scan angles of 40°, 45° and 50° for this 256 element array are shown in Fig. 8.12. (The phasing actually produces a scan angle of $-40°$, $-45°$ and $-50°$; we note this and do not mention it again.) For this computation, a small amount of loss was added to the substrate; $\tan\delta = 0.002$ was used, which is representative of a good low-loss Teflon-fiberglass substrate. Figure 8.12 plots gain, so substrate loss is taken into account; it is clear that the array works very

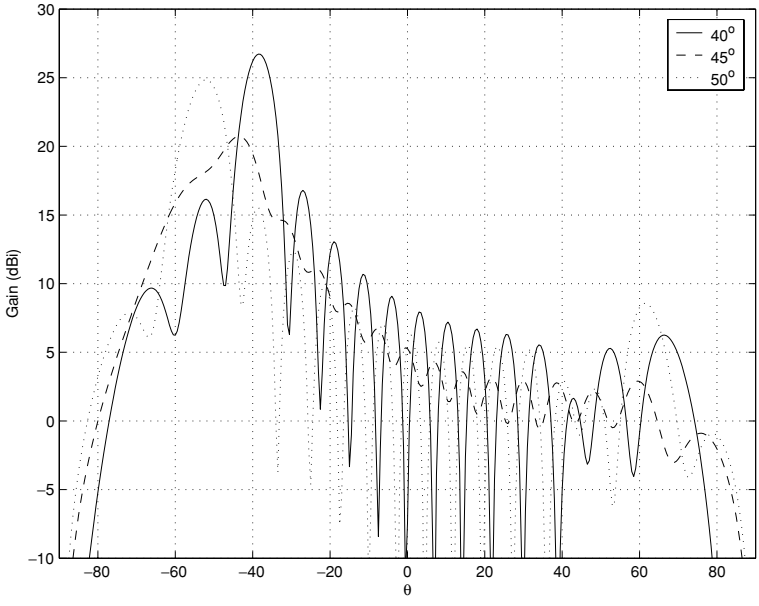


Figure 8.12 E-plane radiation patterns for scan angles of 40°, 45° and 50°, showing scan blindness at 45°.

poorly at $\theta_s = 45°$. In reality, the situation is even worse, since the gain G does not take into account the mismatch loss $(1 - |\Gamma|^2)$ which the antenna presents to the source. The literature on antennas does not seem to adopt a consistent approach to incorporating this effect; some authors [2] incorporate it into antenna efficiency. The product of $G(1 - |\Gamma|^2)$ is also sometimes referred to as realized gain. At the blind scan angle, $\Gamma \approx 1$ so the product $G(1 - |\Gamma|^2)$ is almost zero. At $\theta_s = 50°$, the pattern has improved again, although the peak gain is not quite as large as at 40°. The reason for this is no doubt that the magnitudes and phases of the element currents on the outside of the array differ significantly from those of the central elements due to the different active impedances, and this effect becomes more pronounced as the scan angle increases off broadside.

Modelling hints – array feeds

When modelling an array, the feeds need to imposed. Most MoM codes permit a number of sources to be used. Most sources are essentially *impressed voltage* feeds: a voltage is specified at each feed point – or port in network theory. The code then computes the resulting current, and from this, the impedance at the port. Multiple feeds simply augment the right-hand side (or forcing) vector $\{V\}$ of the generalized MoM impedance matrix $[Z]\{I\} = \{V\}$. Note that active

impedance calculations, unlike mutual power coupling ones, are not affected by the terminating impedance(s) at the other ports.

If some type of loop structure is used to apply feeds – for instance, for different scan angles – *it is very important to ensure that all the previous feeds are zeroed*. How this is done varies from code to code; in FEKO, one tags the first source as a new source and the code zeros all previous ones.

Modelling hints – a useful equivalence for strip dipoles

A strip dipole is often very thin in comparison to its length, and currents are thus essentially constrained to flow along its length in much the same way as with a wire dipole. In this case, it is possible to model a strip as a thin dipole. For a strip width W, the equivalent wire radius a to use is $a = W/4$ [3]. In the case just discussed, we chose to model the strip with surface patches, but using equivalent dipoles would be an option worth pursuing.

8.5 A concluding discussion of stratified media formulations

The printed dipole array example concludes this chapter on the practical application of Sommerfeld potentials. Before leaving this topic, it is worth briefly mentioning the issue of memory requirements and run-times. For the former, there is little overhead when using the Sommerfeld formulation, since the memory requirement is still dominated by the matrix storage, which remains N^2, with N the number of unknowns. For the 256 element array, the FEKO simulation with $h/\lambda_0 = 1/25$ had 4864 basis functions, but the use of symmetry resulted in only 2432 unknowns. This required around 183 MB of RAM to store. The statistics for run-time are interesting: this is a moderately large problem in MoM terms, and in a free-space environment one would expect the matrix solution time to start dominating the run-time. In this case, however, using the Sommerfeld potentials, the time required to compute the impedance matrix elements exceeded the time required to solve the linear system by a factor of around fifteen.[3] By comparison, for a free-space problem with the same number of unknowns (and the same memory requirement), the ratio was around two and a half. Note that we are comparing run-times for problems with the *same number of unknowns*, to get an idea of the cost of the Sommerfeld potentials compared to the free-space Green function. This is *not* equivalent to running the simulation with a grounded substrate with $\epsilon_R = 1$, i.e. vacuum. In this case, one needs to image the patches and the feeds in the ground

[3] Actual "wall clock" run-times are so dependent on computer technology that, in common with much of this book, we prefer to use ratios where possible.

plane, using symmetry, so the equivalent problem will have more unknowns. Hence *for the same physical problem* of a grounded "dielectric" (actually vacuum) slab, the Sommerfeld formulation is actually little more costly.[4] Of course, this is not relevant in practice, since most substrates have dielectric constants significantly larger than unity, and there is no alternative but the Sommerfeld approach.

Summarizing the development in this and the preceding chapter, stratified media MoM formulations are theoretically complex, challenging to implement but potentially very efficient. This is largely due to *only* the metallic regions of the antenna (wire, patch, feed network etc.) being discretized – hence quite large microstrip antennas can be modelled. In the context of RF and microwave engineering, the most important contemporary application of this theory is to printed antenna technology, of which microstrip is the most commercially important type, and our examples have concentrated on this technoloqy. Historically, terrestrial broadcasting, especially LF, MF and HF was another important application – indeed, this prompted Sommerfeld's original work – but with the exception of some specialized military systems, this is hardly a dominant technology at present. Subsurface imaging is another significant contemporary application; however, real grounds are not always well stratified, and even if so, the stratifications may not be parallel with the ground–air interface.

In concluding this chapter, some final points should be noted. Firstly, the Sommerfeld–MoM assumes an *infinitely* large substrate on a similarly infinite ground plane. Hence, such MoM programs do *not* provide any information about the effects of *finite* substrates/grounds. Also, many programs based on the Sommerfeld potentials are not truly general purpose. There are theoretical reasons for this: the near-fields are typically obtained via interpolation tables, the far-fields via asymptotic integrals, which may neglect some terms. Using such a program, especially for fields very close to interfaces, may result in anomalies; see for example [9]. However, for the purpose most commercial codes are designed for, usually microstrip and printed structures, the codes are generally robust and accurate.

References

[1] J. R. James and P. S. Hall, eds., *Handbook of Microstrip Antennas*. London: Peter Peregrinus (on behalf of IEE), 1989.
[2] C. A. Balanis, *Antenna Theory: Analysis and Design*. New York: Wiley, 2nd edn., 1997.
[3] W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design*. New York: Wiley, 2nd edn., 1998.

[4] Note that many codes – FEKO, for instance – will treat this case as an error if using the Sommerfeld formulation, and one is forced to use free-space imaging as above.

[4] R. P. Jedlicka, M. T. Poe, and K. R. Carver, "Measured mutual coupling between microstrip antennas," *IEEE Trans. Antennas Propag.*, **29**, 147–149, January 1981.
[5] J. R. Mosig, R. C. Hall, and F. E. Gardiol, "Numerical analysis of microstrip patch antennas," in *Handbook of Microstrip Antennas* (J. R. James and P. S. Hall, eds.). London: Peter Peregrinus (on behalf of IEE), 1989.
[6] D. M. Pozar and D. H. Schaubert, *Microstrip Antennas: The Analysis and Design of Microstrip Antennas and Arrays*. New York: IEEE Press, 1995.
[7] D. M. Pozar and D. H. Schaubert, "Scan blindness in infinite phased arrays of printed dipoles," *IEEE Trans. Antennas Propag.*, **32**, 602–610, June 1984.
[8] D. M. Pozar and D. H. Schaubert, "Analysis of an infinite array of rectangular microstrip patches with idealized probe feeds," *IEEE Trans. Antennas Propag.*, **32**, 1101—1107, October 1984.
[9] D. B. Davidson and H. d. T. Mouton, "Validation of, and limitations on, the use of NEC-4 for radiation from antennas buried with a homogeneous half-space," *Appl. Comput. Electromagn. Soc. J.*, **13**(3), 302–309, 1998.

9

An introduction to the finite element method

9.1 Introduction

The finite element method (FEM) is one of the best-known methods for the solution of partial differential equations in applied mathematics and computational mechanics. It is a method for solving a differential equation subject to certain boundary values, and in its modern form originated in the field of structural mechanics during the late 1950s; the first specific usage of the term "element" is due to no lesser a person than Courant. In common with the MoM, its historical antecedents are far older than this, in this case dating back to the nineteenth century and the variational methods first described by Lord Rayleigh. It is very widely and routinely used in structural mechanics today, as well as in computational fluid dynamics, computational thermodynamics, the numerical solution of Schrödinger's equation, field problems in general, and of course, in electromagnetics.

An historical aside – Courant and the finite element method

The finite element method as presently accepted can be credited to Courant – whom we have already encountered in the context of the Courant limit for the FDTD method. The published version of his 1942 address to the American Mathematical Society contained an appendix added after the talk, to show by example how variational methods could be put to wider use in potential theory. He used piecewise linear approximations, on a set of triangles which he called "elements" – and thus the method was born [1, p. 5].

With the background we have now acquired with the FDTD and MoM, readers will recognize many features in common with both of these methods in the

treatment to follow; indeed, they will probably not be surprised to learn that both can be formulated within an FEM setting. In common with the MoM, the core idea is to replace some unknown function on a domain by an ensemble of elements, with *known* shape but *unknown* amplitude. Unlike the basic FDTD, where the approximation of the $\vec{E}$ and $\vec{H}$ fields is always done on a rectangular, staggered grid, the FEM permits very general geometrical elements to be used and (usually) only uses one grid. The most widely used elements are known as *simplicial* – this simply means line elements in 1D, triangular in 2D and tetrahedral in 3D. Nonetheless, rectangular, prismatic and even curvilinear elements also find widespread application. Since the improved geometrical modelling made possible especially by triangular or tetrahedral meshes is one of the major features distinguishing the FEM from the FDTD, our study of the FEM will be restricted to these elements. Interested readers may find treatments of other element shapes in the references.

Similar to the FDTD, but unlike the MoM, the FEM is based on a local description of the field quantities, derived from the differential equation description of the Maxwell equations, and does not automatically incorporate the Sommerfeld radiation condition.[1] In practice, this means some form of mesh termination scheme is required. The easiest is usually an absorbing boundary condition of some type. (However, it is also possible to use an "exact" termination scheme using the MoM on the boundary. This is covered in Chapter 10.) In common with the FDTD, and due to the differential equation basis of the two methods, the FEM permits very straightforward treatment of material discontinuities.

The FEM was first applied in electromagnetics during the late 1960s, at much the same time as the initial work using the MoM and FDTD. The two earliest applications were independently published by Silvester, and by Arlett, Bahrani and Zienkiewicz. Some of the history of the FEM in electromagnetics may be found in [1, p. 5]. However, this promising start was arrested during the 1970s and early 1980s because of a problem called "spurious modes" which, combined with substantial computational cost and complex coding, held back widespread adoption of the FEM in electromagnetics. Fortunately, there was a major theoretical breakthrough with *edge elements* in 1980s, which led to a far greater understanding of the spurious mode problem, and the introduction of largely effective solutions. This improved theoretical understanding, combined with the widespread availability of very powerful computers, and increasing interest in wave interaction with non-metallic structures, has made the FEM a major analysis tool of contemporary CEM.

[1] This should not be confused with the Sommerfeld *potentials* for stratified media.

9.2 Variational and Galerkin weighted residual formulations: the Laplace equation

9.2.1 The weighted residual approach

The FEM can be derived via two different, but equivalent, procedures. On the one hand, there is the Galerkin weighted residual formulation, already encountered in Chapter 4.[2] On the other hand, there is the variational approach (or more fully, the variational boundary value problem). The latter is used by most textbooks. The former is more direct at the formulation level, but incorporating the boundary conditions is somewhat less obvious. We will discuss both approaches in this chapter.

As usual, we will first illustrate the ideas with a simple example. Consider the following partial differential equation (PDE) in two dimensions:

$$\nabla \cdot \epsilon \nabla \phi = 0 \tag{9.1}$$

For linear, isotropic media, we have $\epsilon = \epsilon_r \epsilon_0$, and this is equivalently

$$\nabla \cdot \epsilon_r \nabla \phi = 0 \tag{9.2}$$

In a materially homogeneous region, this reduces to the Laplace equation:

$$\nabla^2 \phi = 0 \tag{9.3}$$

With a PDE, boundary conditions must of course be specified. For a second-order PDE such as this, the following on the *closure* (boundary) are necessary and sufficient for a unique solution.[3]

- A value of function ϕ is specified – this is a *Dirichlet* boundary condition. If $\phi = 0$, this is called a *homogeneous* boundary condition.
- A value of the normal derivation, $\frac{\partial \phi}{\partial n}$, is specified – this is a *Neumann* boundary condition. Again, if $\frac{\partial \phi}{\partial n} = 0$, this is called a *homogeneous* Neumann boundary condition.
- A linear combination of the above is specified – $\epsilon \frac{\partial \phi}{\partial n} + \gamma \phi = q$. This is known as a *mixed* boundary condition (also sometimes as a Cauchy boundary condition); the Neumann boundary condition is a special case of this with $\gamma = 0$.

Note that these may be mixed in any ratio along the boundary: the boundary may be entirely Dirichlet, or entirely Neumann,[4] or entirely mixed, or some combination of these along different sections of the boundary. However, they must be *disjoint* – that is, more than one may not be simultaneously specified along the same part of the boundary.

[2] Readers who are not working through this book sequentially might wish to read Chapter 4 at this stage.
[3] Note that for higher-order PDEs, additional boundary conditions are required.
[4] In which case, the PDE can be solved only to within an unknown constant.

In terms of our previous work on the method of weighted residuals in Chapter 4, the linear operator for the Laplace equation is $\mathcal{L} = \nabla^2 \cdot$; the unknown function $f = \phi$ and the forcing function $g = 0$. (Again, in this case, the mathematical term homogeneous is sometimes used, this time in the context of the PDE.) We proceed as with the MoM, by introducing basis functions, weighting functions W and an inner product. The unknown function (the potential ϕ in this case) is expanded as

$$\phi \approx \sum_{i=1}^{N} a_n h_n \tag{9.4}$$

Suitable weighting functions w_m are introduced:

$$W = \sum_{m=1}^{M} w_m \tag{9.5}$$

and an inner product is defined for this two-dimensional problem as

$$\langle a, b \rangle = \iint_S ab \, dS \tag{9.6}$$

Hence, as before, a linear system is obtained, with entries of the following form for the m, nth system matrix element:

$$\langle w_m, \mathcal{L} a_n h_n \rangle \tag{9.7}$$

At this stage, this looks so similar to the MoM that one might wonder why the FEM is regarded as a different method. (Indeed, a number of workers in the 1980s tried to unify the methods thus.) Although in general terms there are indeed similarities at this very fundamental functional analysis level, in practice there are great differences which lead to different algorithms being required. The most important is that the operator $\mathcal{L}$ is now a *differential* as opposed to an integral operator; this means that only elements in close geometrical proximity have non-zero system matrix entries, and hence a very large number of the matrix entries are zero. Mathematically, this is a *sparse* matrix; the MoM with integral equations generates full matrices. Another important difference is that with the MoM integral equation formulation, the boundary conditions are built into the formulation; with the FEM, these must be explicitly imposed (and we have not discussed at all how to do this).

In short, the devil is in the detail, which we must now address, and to do this it is convenient to use the variational approach, the topic of the next section. But first, some finite element terminology will be introduced. With finite elements, we usually employ basis functions which span only a small part of the domain – subsectional as opposed to entire domain, in MoM parlance. This region is generally known as the *element*, and the basis function is also frequently called the

shape function. The term *elemental function* is also sometimes encountered. (It is frequently thought that the term "finite element" comes from this geometrical decomposition into finite regions – as opposed to infinite elements, which are also sometimes used – but it has also been attributed to the finite energy in an element.) As with the MoM, a variety of shape functions have been used. Generally, the most useful are polynomial interpolation functions – although shortly we will see another type of incomplete polynomial function, which is not interpolatory, but is very widely used, namely the edge-based element.

9.2.2 The variational approach

The equivalent variational functional

At this stage, we are going to look at the finite element method from a different viewpoint, namely that of the variational functional approach. Instead of directly solving Eq. (9.1), we are going to work with an equivalent problem, namely an energy *functional*, whose minimum corresponds to the solution of the PDE. For Eq. (9.1), a suitable functional is:

$$W(\phi) = \frac{1}{2}\iint \epsilon(\nabla\phi)^2\, dS \tag{9.8}$$

We state this without proof for the present – subsequently we will return to this, since the proof yields important information about the boundary conditions. We note that this is the energy $\frac{1}{2}\iint \vec{D}\cdot\vec{E}\, dS$. We also note that the function ϕ in the original equation had to be at least twice differentiable; in the above, it need only be once differentiable. Due to this "weakening" of the requirements on the function, this is sometimes called the *weak formulation*. For a linear, isotropic medium, we have $\epsilon = \epsilon_r\epsilon_0$ and since we are eventually going to set the derivative of W to zero, we can just as well divide out by ϵ_0 at this stage, leaving only the ϵ_r term:

$$W(\phi) = \frac{1}{2}\iint \epsilon_r(\nabla\phi)^2\, dS \tag{9.9}$$

The shape functions

In one dimension, the only choice to make is the shape of the *basis* function, but in two and three dimensions, we can choose both the shape function and the geometrical shape of the element. The most popular choices in two dimensions are *triangular* and *quadrilateral* elements; for reasons already discussed, we will focus on triangular elements, although we will use rectangular elements to introduce some ideas regarding vector elements. Assuming that the geometrical region (the domain) has been decomposed into elements – later, we will discuss ways of doing

this – we note that Eq. (9.8) is valid within each element, and in the following we will initially focus on this energy functional on an element-by-element basis.

Zero-order elements (the equivalent of the pulse basis functions we used for the first MoM example back in Chapter 4) cannot be differentiated even once, so are not admissable in this problem. Hence, we will start with first-order elements. In this case, the approximating function can be written as

$$\phi \approx a + bx + cy \tag{9.10}$$

The constants a, b and c are, of course, what we require the FEM eventually to compute for us. However, it is more convenient to write this in a form where the unknowns are the potentials at the three triangle nodes, or in other words:

$$\phi \approx \alpha_1(x, y)\phi_1 + \alpha_2(x, y)\phi_3 + \alpha_3(x, y)\phi_3 \tag{9.11}$$

This assumes the existence of suitable functions $\alpha_1(x, y)$, $\alpha_2(x, y)$ and $\alpha_3(x, y)$; their properties will emerge shortly.

Noting $\phi_1 = a + bx_1 + cy_1$ and similarly for the other two nodes, we have

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{9.12}$$

Inverting the nodal coordinate matrix, we find:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} \tag{9.13}$$

Now we have:

$$\phi = [\,1\ x\ y\,] \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} \tag{9.14}$$

which may be rewritten as

$$\phi = \sum_{i=1}^{3} \phi_i \alpha_i(x, y) \tag{9.15}$$

with

$$\alpha_1 = \frac{1}{2A}[(x_2y_3 - x_3y_2) + (y_2 - y_3)x + (x_3 - x_2)y] \tag{9.16}$$

and A the triangle area (which is conveniently half the determinant of the nodal coordinate matrix).[5] The other functions α_2 and α_3 are obtained by cyclic interchange of the indices, modulus three.

Note that the functions α_i are interpolatory on the three vertexes (nodes): i.e. unity at node i, and zero at the other nodes. (Once again, we caution that not all the finite elements we will study have this property.)

Manipulating the energy term

Substituting Eq. (9.15) into Eq. (9.8), the following is obtained for the energy in an element e:

$$W^e = \frac{1}{2}\iint_{S^e} \epsilon_r^e \nabla\phi \cdot \nabla\phi \, dS$$
$$= \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3} \epsilon_r^e \phi_i \left[\iint_{S^e} \nabla\alpha_i \cdot \nabla\alpha_j \, dS\right] \phi_j \tag{9.17}$$

where we have now assumed that the permittivity is constant within element e. This is very compactly written in matrix notation as:

$$W^e = \frac{1}{2}\{\phi\}^T \epsilon_r^e [S^e]\{\phi\} \tag{9.18}$$

with $\{\phi\}$ the vector of nodal potentials and

$$S_{ij}^e = \iint_{S^e} \nabla\alpha_i \cdot \nabla\alpha_j \, dS \tag{9.19}$$

This matrix is often called the *stiffness matrix*, from the structural mechanics origin of the method, but this has no physical meaning in electromagnetics and we will not use this term frequently. (In this chapter, we will use $[S]$ for this matrix, and $[T]$ for another frequently encountered matrix. This notation is due to Silvester and Ferrari [2]. Unfortunately, there is no standard notation in this regard in the literature. Peterson, for instance, uses $[E]$ and $[F]$ respectively [3], as does Jin [4].) The expressions are simple to evaluate, for example,

$$S_{12}^e = \frac{1}{4A}[(y_2 - y_3)(y_3 - y_1) + (x_3 - x_2)(x_1 - x_3)] \tag{9.20}$$

Connecting the elements

At this stage, we have worked in isolation, considering the element on its own. Each element has nodes numbered *locally* from one to three. In practice of course, there will be a (perhaps very large) number of elements, with nodes numbered

[5] Note here that A is a *signed* quantity, whose sign depends on whether the nodes are numbered clockwise or anticlockwise. See Section 9.7.2 for further discussion.

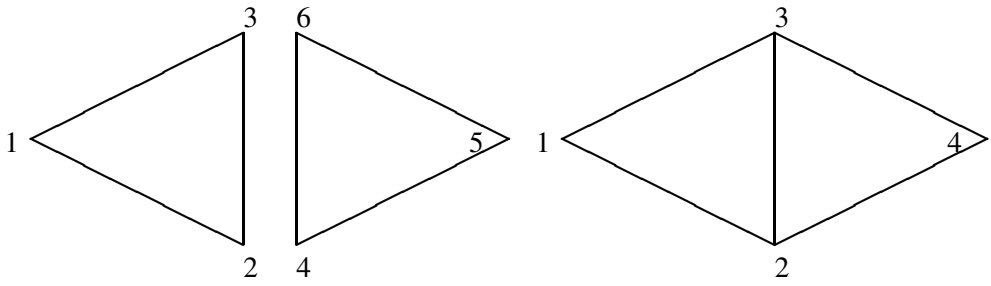


Figure 9.1 Two triangular elements, disconnected (left) and connected (right).

according to some *global* numbering scheme. (It is worth commenting that mapping local to global information, and vice versa, usually requires a significant amount of book-keeping in the average FEM code.) We need some method to *connect* the elements; various approaches are available. At present, we will assume the existence of a *connection matrix* which tells us how to map the unconnected nodes to the connected mesh. As a simple example, see Fig. 9.1, which shows two such triangles. The connection matrix for this system is

$$[C] = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{bmatrix} \tag{9.21}$$

and thus

$$\{\phi_{\text{dis}}\} = [C]\{\phi_{\text{con}}\} \tag{9.22}$$

with

$$\begin{aligned} \{\phi_{\text{dis}}\} &= \{\phi_1\, \phi_2\, \phi_3\, \phi_4\, \phi_5\, \phi_6\}^T_{\text{dis}} \\ \{\phi_{\text{con}}\} &= \{\phi_1\, \phi_2\, \phi_3\, \phi_4\}^T_{\text{con}} \end{aligned} \tag{9.23}$$

Although this may be belaboring the obvious, this connection matrix ensures that the potential at each node is the same on all elements sharing that node. (This seems simple and obvious, but we will see that in the context of vector fields, this may not always be desirable.)

Using this, the resulting equation for the energy in the whole system is:

$$\begin{aligned} W &= \frac{1}{2}\{\phi_{\text{con}}\}^T[S]\{\phi_{\text{con}}\} \\ [S] &= [C]^T[\epsilon_r^e S_{\text{dis}}][C] \end{aligned}$$

The term $[\epsilon_r^e S_{\text{dis}}]$ requires that the stiffness matrix of each element be multiplied by the relative permittivity associated with the element before one proceeds further; it is included in the overall $[S]$ matrix for the problem.

However, the formulation is not completed yet. It will be recalled that it is the solution which *minimizes* the variational functional which corresponds to the solution of the PDE, and all we have here is an expression for the energy in the connected elements. We must now establish this minimum. In doing this, we need to distinguish formally between *free* (f) and *prescribed* (p) potentials here. The latter are those prescribed by the Dirichlet boundary conditions. The former are the *degrees of freedom* (the unknowns) in the problem. It is convenient if we choose to number first the free and then the prescribed potentials; this can be done relatively easily, even for moderately complex geometries.

Differentiating with respect to the free potentials, and setting the resultant expression to zero, one obtains

$$\frac{\partial W}{\partial\{\phi_f\}} = \frac{\partial}{\partial\{\phi_f\}}\left(\{\phi_f \phi_p\}^T \begin{bmatrix} S_{ff} & S_{fp} \\ S_{pf} & S_{pp} \end{bmatrix} \begin{Bmatrix} \phi_f \\ \phi_p \end{Bmatrix}\right) = 0 \tag{9.24}$$

Expanding the quadratic, differentiating, and then using $[S_{fp}] = [S_{pf}]$, yields:

$$\begin{bmatrix} S_{ff} & S_{fp} \end{bmatrix} \begin{Bmatrix} \phi_f \\ \phi_p \end{Bmatrix} = 0 \tag{9.25}$$

or, more conveniently,

$$[S_{ff}]\{\phi_f\} = -[S_{fp}]\{\phi_p\} \tag{9.26}$$

Once again, this is a system of linear equations which can be solved using standard techniques. Here we should note that the matrices $[S_{ff}]$ and $[S_{fp}]$ are sparse, containing only entries where nodes are shared by elements; for initial implementation work we need not exploit this, but FEM codes for practical applications must, or much of the benefit of the FEM is lost. Note also that these terms include ϵ_r^e in the S matrix elements as above.

A mathematical aside – partial differentiation of matrices and vectors

Since the free potentials are most conveniently written as a vector, it is useful to note that vectors can be differentiated much as scalars, viz.

$$\frac{\partial C\{x\}}{\partial\{x\}} = C$$

$$\frac{\partial\{x\}^T[A]\{x\}}{\partial\{x\}} = 2[A]\{x\}$$

etc. In the above, C is a scalar constant and $[A]$ a constant matrix. This result greatly simplifies the analytical work required in minimizing the functional. Such identities can be proven by expanding the vector expression into its components, and then differentiating with respect to each of them in turn. A good reference to read more on this topic is [5, Appendix B].

Coding hints – FEM data structures

Note that in practice one rarely numbers all the nodes in an unconnected fashion first; instead, node 4 on the right-hand triangle would probably be referenced using some data structure of the form `element(m)%nodeone`, with `m` the element number, in a language such as FORTRAN 90 which supports *derived data types* – i.e. objects of a type defined by the user. The % in FORTRAN 90 is a *component selector*, and returns the component called `nodeone` from the `m`th entry in derived data type `element`. In MATLAB (which does not support this type of derived data structure), one might have a variable named `element_nodeone(m)`, or use a two-dimensional array of the form `element_nodes(m,local_node_num)`; there are a variety of possibilities.

Furthermore, even if used, the connection matrix is also not stored as explained here; the reason is that it is highly sparse and could be stored far more efficiently in some type of *compressed storage* scheme.

9.2.3 Some practical issues: assembling the system

In FEM parlance, the process of filling the finite element system matrix is frequently known as *matrix assembly*. For practical codes, it is generally convenient to loop over the elements rather than the nodes (recall that the degrees of freedom are the nodal potentials for this first-order scheme). This is known as *assembly by elements*. For a particular global degree of freedom i, any element which contains this node will contribute to the matrix. For triangles, this number depends on the mesh. We will now discuss two practical methods which simplify this matrix assembly process.

Connecting the system

The connection matrix is useful for explaining the method, but inconvenient. Practical programs do this essentially by inspection. A global numbering system is adopted from the start. As each element's $[S]$ matrix is computed, it is entered into the global matrix. A formal method for doing this has been described in

[2, pp. 51–53], but essentially one simply adds the contributions of each elemental matrix at the appropriate global row and column entry. Once again, note that *sparsity* has not yet been exploited.

Handling the boundary conditions

Repeating the matrix equation to solve,

$$[S_{ff}]\{\phi_f\} = -[S_{fp}]\{\phi_p\}$$

we see that the prescribed boundaries form the right-hand side of the matrix equation. The easiest approach is to number free unknowns first, then prescribed unknowns, as already briefly mentioned. Entries of the form S_{ff} (i.e. both nodes free) are entered into the system matrix; entries of the form S_{fp} (i.e. one node free) are multiplied by the prescribed potential and entered into the right-hand side vector. Entries of the form S_{pf} and S_{pp} play no role. (Actually, $[S_{pf}]^T = [S_{fp}]$ and this is implicitly included during the minimization process, when this is exploited.)

Another method has been described in the literature [2, pp. 49–50] which is useful when it is *not* possible, or very inconvenient, to number first free then prescribed elements; it uses dummy entries, and increases the matrix size slightly.

9.2.4 More on variational functionals

Earlier, we mentioned that the equivalence between the PDE and the variational functional lies at the heart of the variational FEM approach. Having now seen a basic FEM formulation developed, we need to return to the theoretical underpinnings of the method. We will work with the more general Poisson PDE, which includes a source term, for a homogeneous region:

$$\nabla^2\phi = -\frac{\rho}{\epsilon} \tag{9.27}$$

where ρ is the source, and the boundary conditions on $S \equiv S_1 + S_2$, as before, are Dirichlet on S_1 and Neumann on S_2. There does not appear to be a systematic process to construct variational functionals from PDEs (the reverse process is called Euler's method), and usually one instead shows that the proposed variational functional has the required properties. Thus we propose that the following variational functional has an extremal point, which corresponds to the solution of the Poisson equation above, with the required boundary conditions:

$$W(\phi) = \frac{1}{2}\iint \nabla\phi \cdot \nabla\phi \, dS - \iint \phi\frac{\rho}{\epsilon} \, dS \tag{9.28}$$

and we will then show that it indeed has these properties.

Proving the equivalence of the functional and PDE

We will now apply what is known as a variational analysis. We postulate the following:

$$\phi' = \phi + \theta h \tag{9.29}$$

where ϕ' is the trial solution; ϕ is the solution of the PDE; h is some (differentiable) function (which, rather importantly, must be zero at prescribed boundaries since by definition ϕ is known there) and θ is a (real valued) perturbation parameter. This is then substituted into the variational functional, Eq. (9.28):

$$W(\phi + \theta h) = W(\phi) + \theta \iint \nabla\phi \cdot \nabla h \, dS - \theta \iint \frac{\rho}{\epsilon} h \, dS + \frac{1}{2}\theta^2 \iint \nabla h \cdot \nabla h \, dS$$

The first and last terms are always greater than or equal to zero (*positive semi-definite* in mathematical terms). The term in θ is the *first variation*; what we must now show is that this is zero. To do this, we will use *Green's theorem*, which is essentially multi-dimensional integration by parts:

$$\iint_S u\nabla^2 v \, dS = \oint_C u(\nabla v) \cdot d\vec{C} - \iint_S \nabla u \nabla v \, dS \tag{9.30}$$

Using this, one finds:

$$\iint \nabla\phi \cdot \nabla h \, dS = \oint h \frac{\partial \phi}{\partial n} \, dC - \iint h\nabla^2\phi \, dS \tag{9.31}$$

Now, a subtle argument is introduced. The contour integral must be zero to eliminate the first variation. Clearly, on S_1, $h = 0$ by definition, since the value of ϕ is known. If $\frac{\partial \phi}{\partial n} = 0$ on S_2, then we have achieved our aim. This, of course, is just the homogeneous Neumann boundary condition.

The other surface integral term yields

$$-\iint h\nabla^2\phi \, dS = \iint \frac{\rho}{\epsilon} h \, dS \tag{9.32}$$

since ϕ is the solution of the PDE. This cancels with the other term in θ. Hence, the first variation is zero, subject to either Dirichlet boundary conditions on S_1 or homogeneous Neumann boundary conditions on S_2, and we have shown what we set out to achieve.

In the finite element procedure, we actually perform the operation in the inverse order. Minimizing[6] the energy functional, by differentiating with respect to the

[6] In general, one should rather speak of rendering the functional stationary, or finding the extremal point, but for this problem the functional is indeed minimized.

free potentials, is equivalent to forcing the first variation to zero; given prescribed boundary conditions on S_1, we then *naturally* enforce homogeneous Neumann boundary conditions on S_2. (It is worth noting that the latter boundary condition is enforced in an *average* sense on S_2; that is, it is not exactly enforced at each point on S_2.)

Summary of boundary conditions

The issue of boundary conditions is so important with the FEM that it deserves to be highlighted. There are two types of boundary conditions in elementary FEM analysis.

- Dirichlet boundary conditions: these are *essential* and must be explicitly set.
- Homogeneous Neumann boundary conditions: these are *natural* and are implicitly enforced. An homogeneous Neumann boundary condition corresponds to a symmetry plane; it is often used to reduce the computational domain.

The reason that it is so important to be aware of this is that even if one is only using an FEM code and has no intention of ever writing one, code developers assume that users know this. In particular, it is very important to realize that an unset boundary condition is not an error in the FEM process: it is a natural homogeneous Neumann boundary condition.

As mentioned earlier, more complex boundary conditions may also be encountered, including inhomogeneous Neumann boundary conditions and mixed boundary conditions.

Boundary conditions at material interfaces

One of the great strengths of the FEM is that handling inhomogeneous regions is very simple. There are, however, one or two subtleties worth highlighting. The boundary conditions on the electrostatic potential at the interface between regions 1 and 2, with appropriate dielectric constants, are:

$$\phi_1 = \phi_2 \tag{9.33}$$

$$\epsilon_{r1}\frac{\partial \phi_1}{\partial n} = \epsilon_{r2}\frac{\partial \phi_2}{\partial n} \tag{9.34}$$

The former comes from the requirement of tangential electric field continuity, the latter from normal electric flux continuity.

With the connection matrix approach, we force potentials to be continuous at a material interface. It turns out that the latter is a natural boundary condition of the variational approach. This is an important point. To show this, one starts with

$$\nabla \cdot \epsilon \nabla \phi = -\frac{\rho}{\epsilon} \tag{9.35}$$

and the variational functional

$$W(\phi) = \frac{1}{2}\iint \epsilon \nabla\phi \cdot \nabla\phi \, dS - \iint \phi\rho \, dS \tag{9.36}$$

and proceeds with an analysis along exactly the same lines as before, but with the domain split in two.[7] Two additional terms then appear in the first variation, representing the flux continuity condition at the interface. From the stationarity requirement, flux continuity follows (for details, refer to [4, Section 3.2]).

Within a code, the above is usually entirely invisible to the user.

9.2.5 The Poisson equation: incorporating a source term

Including the term $-\frac{1}{\epsilon}\int \phi\rho \, dS$ representing the source in the functional results in a new matrix, $[T]$. (Again, for historical reasons, this is sometimes called the *mass matrix*.) The (known) source term ρ is discretized using the same interpolation scheme as ϕ, i.e. first-order triangular finite elements in this case, *but with known coefficients*. The entries in $[T]$ are computed from

$$T_{ij}^e = \iint_{S_e} \alpha_i \alpha_j \, dS \tag{9.37}$$

with α the nodal interpolation functions as before.

The result is a matrix equation of the following form:

$$[S_{ff}]\{\phi_f\} = \frac{1}{\epsilon}[T]\{\rho\} - [S_{fp}]\{\phi_p\} \tag{9.38}$$

It is interesting to note that the inhomogeneous part of the PDE ($\{\rho\}$) plays the same role in the finite element system matrices as the inhomogeneous part of the boundary conditions ($\{\phi_p\}$).

9.2.6 Discussion

This completes our introductory discussion of the method. An obvious extension for the Laplace (and Poisson) equations is to introduce higher-order elements, using quadratic, cubic, quartic or even higher. This has been very comprehensively addressed in [2], and for static problems works very well. However, for dynamic problems, our main interest, we need to introduce a different type of element, called variously the edge element, vector element or Whitney element, so we will not pursue scalar elements any further. However, before we can address vector

[7] The extension to an arbitrary number of different materials is obvious.

elements, we need to introduce a concept widely used in FEM analysis, namely simplex coordinates, the topic of the next section.

9.3 Simplex coordinates

Simplex coordinates – also known as homogeneous or barycentric (or in 2D, area) coordinates – provide an entirely local geometrical description within a triangle (in 2D) or tetrahedron (in 3D). This is very convenient, since it allows much of the work required to be done once (on what is often called the "parent" triangle) and then with some simple geometrical scaling, it can be applied to any triangle or tetrahedron. They are intimately linked to *simplicial elements* – the simplest possible geometrical shape in the space, that is line elements in one dimension, triangles in two dimensions and tetrahedra in three dimensions. (The concept can be extended to higher dimensions, but loses any geometrical interpretation.)

In general, simplex coordinates are defined as the ratios of lengths (1D), areas (2D) or volumes (3D) that a point in the interior (or on the boundary) splits the line/triangle/tetrahedron into. The size $\sigma(S)$ of a simplex S is defined as:

$$\sigma(S) = \frac{1}{N!}\begin{vmatrix} 1 & x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(N)} \\ 1 & x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(N)} \\ \vdots & & & & \\ 1 & x_{N+1}^{(1)} & x_{N+1}^{(2)} & \dots & x_{N+1}^{(N)} \end{vmatrix} \tag{9.39}$$

where superscripts denote space directions and subscripts denote vertices.

9.3.1 Simplex coordinates in one, two and three dimensions

In one dimension, we have

$$\lambda_1 = \frac{\sigma(S_1)}{\sigma(S)} = \frac{\begin{vmatrix} 1 & x \\ 1 & x_2 \end{vmatrix}}{L} = \frac{x_2 - x}{L}$$

$$\lambda_2 = \frac{\sigma(S_2)}{\sigma(S)} = \frac{\begin{vmatrix} 1 & x_1 \\ 1 & x \end{vmatrix}}{L} = \frac{x - x_1}{L}$$

These express the ratios of length from the right and left nodes respectively to point x, to the total length of the element. These are frequently encountered in MoM analysis as local coordinates.

In two dimensions, we have

$$\begin{aligned}\lambda_1 &= \frac{\sigma(S_1)}{\sigma(S)} \\ &= \frac{\begin{vmatrix} 1 & x & y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}}{2A} \\ &= \frac{(x_2 y_3 - x_3 y_2) + (y_2 - y_3)x + (x_3 - x_2)y}{2A}\end{aligned} \tag{9.40}$$

This represents the ratio of the area of the triangle P23 to 123 – see Fig. 9.3, in Section 9.6.3. It will be noted that $\lambda_1 = \alpha_1$, the first-order interpolatory function used in our earlier analysis, indicated how convenient the simplex coordinates are for functions defined over a triangle. There are three simplex coordinates in 2D: λ_1, λ_2 and λ_3, describing the three area ratios.

In three dimensions, we have

$$\begin{aligned}\lambda_1 &= \frac{\sigma(S_1)}{\sigma(S)} \\ &= \frac{\begin{vmatrix} 1 & x & y & z \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix}}{6V}\end{aligned}$$

This represents the ratio of the volume of the tetrahedron P234 to the volume of the element.

There are four simplex coordinates in 3D: λ_1, λ_2, λ_3 and λ_4, describing the four volumetric ratios.

9.3.2 Some properties of simplex coordinates

Aside from the interpretation as the ratio of sizes, simplex coordinates have other important properties. Some of these are as follows.

- The coordinates are normalized, thus $\sum_{i=1}^{N+1} \lambda_i = 1$.
- In two and three dimensions, the *gradient* of each simplex coordinate is a constant, and normal to the relevant edge (2D) or face (3D). In 2D, for example:

$$\nabla\lambda_i = \frac{l_i}{2A}\hat{n}_i \tag{9.41}$$

with A the area of the triangle, l_i the length of edge i, and $\hat{n}_i$ the normal to edge i. This property is extensively exploited in vector elements, of which more later.

- Because of the normalization, $0 \leq \lambda_i \leq 1 \, \forall i$. This can be a useful and quick test to see whether a point lies inside or outside an element.

9.4 The high-frequency variational functional

For electrodynamic problems, subject to the deterministic vector wave equation,

$$\nabla \times \frac{1}{\mu_r} \nabla \times \vec{E} - k_0^2 \epsilon_r \vec{E} = -jk_0 Z_0 \vec{J} \tag{9.42}$$

with $\vec{J}$ a source internal to domain Ω and k_0 the free-space wavenumber, the equivalent variational functional which must be rendered stationary is:[8]

$$F(\vec{E}) = \int_\Omega \left[\frac{1}{\mu_r} |\nabla \times \vec{E}|^2 - k_0^2 \epsilon_r |\vec{E}|^2 \right] d\Omega + jk_0 Z_0 \int_\Omega \vec{E} \cdot \vec{J} \, d\Omega \tag{9.43}$$

This assumes either homogeneous Dirichlet or Neumann boundary conditions or a mixture of the two on the boundary of domain Ω.

A closely related functional for the source-free vector wave equation

$$\nabla \times \frac{1}{\mu_r} \nabla \times \vec{E} - k_i^2 \epsilon_r \vec{E} = 0 \tag{9.44}$$

is the following:

$$F(\vec{E}) = \int_\Omega \left[\frac{1}{\mu_r} |\nabla \times \vec{E}|^2 - k_i^2 \epsilon_r |\vec{E}|^2 \right] d\Omega \tag{9.45}$$

subject to the same boundary conditions. In this case, the solution is the set of eigenvalues k_i and associated eigenvectors $\vec{E}_i$.

In order to show the above properties, one proceeds in a fashion similar to the Poisson equation, using a vector Green's theorem for the double-curl operator. The details are available in [2, 4] and although more complex than the Poisson case, the method is the same, so we will not repeat them here.

This form (often called the curl-curl form) has been used for high-frequency FEM analysis for many years. However, although it appears fairly straightforward to discretize, it turned out to have a number of problems which occupied analysts for some years. One of the most important advances was the introduction of vector (edge) elements in the late 1980s, and this is the topic of the next section.

[8] This is actually the functional for lossless materials; see [4, Chapter 6] for further discussion of this.

9.5 Spurious modes

One of the supposed strengths of the FEM was its accuracy, in particular when compared to a method such as the FDTD, until serious problems with "spurious modes" were found using standard (node-based) FEM for electromagnetic eigenvalue problems (we will define these later). The traditional, nodal FEM approach, typical of structural mechanics, deals with a vector field by approximating each component separately:

$$E_x \approx \sum E_{x_i} f(x, y, z) \tag{9.46}$$

with $f(x, y, z)$ a standard basis function such as those we have already seen (although extended to three dimensions). This was then repeated for E_y and E_z and substituted into Eq. (9.43) or (9.45). As Silvester and Pelosi comment [1, p. 8]:

> The first approach (nodal elements) may be called the structural mechanics approach. . . . at least some theory and much practical experience should be transferable to electromagnetics. Further, it has the appeal of simplicity and familiarity. The same approximating functions can be called upon to serve for both scalar and vector cases, and the vectorial coefficients have clear meaning as component representations of $\vec{E}$ or $\vec{H}$. . . . the structural mechanics approach has one major flaw for electromagnetic field analysis: *it doesn't work very well*. The reason is simply that the fields that occur in structural mechanics and those encountered in EM are fundamentally different. The electromagnetic field vectors not only obey the Maxwell curl equations, but they are also constrained by the divergence equations.

Before discussing some of the more intricate details of spurious modes, we note an immediate and practical problem with the nodal approach: since the field is approximated by its values at the nodes, if we use the method we used for the static problems for connecting the elements (that is, all values at a node are set equal on all the elements which share the node), then the result is that we force *all* components of the field to be continuous. At an interface between two different types of material, only the *tangential* components of $\vec{E}$ or $\vec{H}$ should be continuous. If the material boundary happens to coincide with a plane parallel to one of the coordinate axes, then it would not be too difficult to arrange that we do this with only the tangential field components, leaving additional degrees of freedom to permit the normal field to be discontinuous. But in general, we are unlikely to be so fortunate, and the material interface will create a very tricky problem indeed.

By comparison, the vector FEM approach approximates the full vector field

$$\vec{E} \approx \sum e_{ij} \vec{w}_{ij} \tag{9.47}$$

with the edge-based *vector* function:

$$\vec{w}_{ij} = \lambda_i \nabla \lambda_j - \lambda_j \nabla \lambda_i \tag{9.48}$$

As before, λ_i is the simplex coordinate with respect to node i. This is then used to discretize Eq. (9.43) or (9.45).

It is far from immediately apparent why what appears to be a minor change in approach should yield significantly better solutions – after all, the vector basis functions are simply another way of representing the vector nature of the problem. In order to understand this, we need first to look a little more carefully at the high-frequency functionals. We will start with the eigenvalue problem, where the problems originate.

Following the standard discretization and substitution of the basis functions, the stationary points of the functional, Eq. (9.45), correspond to solutions of the following generalized eigenmatrix equation:

$$[S]\{e_i\} = k_i^2[T]\{e_i\} \tag{9.49}$$

where $[S]$ and $[T]$ represent the discretized versions of the first and second terms in Eq. (9.45). The eigenvalues k_i represent the resonance frequencies of the cavity, and the vectors $\{e_i\}$ the eigenvalues, i.e. the various resonant modes (or eigenmodes).

Various approaches are now possible. A particularly revealing one is to note that the divergence constraint,

$$\nabla \cdot \epsilon \vec{E} = 0 \tag{9.50}$$

is implied within the functional, but in *frequency dependent* form. We can see this by taking the divergence of both sides of Eq. (9.44); noting the vector identity $\nabla \cdot \nabla \times \vec{a} \equiv 0 \ \forall \vec{a}$, it is clear that

$$k_i^2 \nabla \cdot \epsilon_r \vec{E} = 0 \tag{9.51}$$

For the dynamic case (that is, $k_i \neq 0$) the divergence equation is indeed satisfied.

The problem, however, enters via the *other* possibility for satisfying this equation, namely $k_i = 0$. In this case, the divergence equation *is no longer necessarily satisfied*. This corresponds of course to the static case, where $\vec{E} = -\nabla V$, and we note (since $\nabla \times \nabla V = 0 \ \forall V$) the theoretically infinite number of solutions of the form of the field as the gradient of a potential and zero eigenvalue $\{(\nabla V, 0)\}$ also satisfies the vector wave equation, constituting its *null-space* (also known as kernel, abbreviated *ker*, in some of the literature).

A particularly elegant example of such a null-space eigenmode for the well-known rectangular waveguide problem was given by [6], and it is so illuminating that it is worth repeating here. Peterson considered the classic eigenvalue problem of a rectangular waveguide with PEC walls, dimensions a by b, with the

solutions

$$\vec{E}_{mn} = -\hat{x}\frac{n\pi}{b}\ \cos\frac{m\pi x}{a}\ \sin\frac{n\pi y}{b} + \hat{y}\frac{m\pi}{a}\ \sin\frac{m\pi x}{a}\ \cos\frac{n\pi y}{b} \tag{9.52}$$

with eigenvalues:

$$k_{mn}^2 = \left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2 \tag{9.53}$$

This is very well known and features prominently in almost any undergraduate electromagnetic text. However, these texts never mention that there is *another* valid solution of the vector wave equation, viz. the static solution:

$$\vec{E}_{mn}^{\text{spur}} = \hat{x}\frac{m\pi}{a}\ \cos\frac{m\pi x}{a}\ \sin\frac{n\pi y}{b} + \hat{y}\frac{n\pi}{b}\ \sin\frac{m\pi x}{a}\ \cos\frac{n\pi y}{b} \tag{9.54}$$

with eigenvalues $k_{mn} = 0$.

The spurious solution(s) look almost identical to the waveguide solutions, but are critically different – note that they can be written in the form

$$\vec{E}_{mn}^{\text{spur}} = \nabla\left(\sin\frac{m\pi x}{a}\ \sin\frac{n\pi y}{b}\right) \tag{9.55}$$

Also very importantly, unlike Eq. (9.52), these static solutions do *not* have zero divergence, as can quickly be established by inspection.

Because the eigenvalues of these "spurious modes" are zero, these are simply rejected as unwanted solutions when one does an analytical solution of the problem (using separation of variables, for instance). However – and this is a critical point! – the variational functional admits these solutions, and *the finite element procedure will also compute them*. (Unless, that is, one can modify the functional to exclude these solutions – there has been success with such approaches and we will mention this again later, but the formulation is somewhat more involved.)

So, to summarize, due to the properties of the high-frequency variational functional, the finite element procedure will produce not only the wanted, dynamic eigenvalues and eigenvectors, but also a number of "zero" eigenvalues and associated static eigenvectors. Since the finite element solution is of course approximate, the "zero" eigenvalues will not be exactly zero, but may shift up in frequency. If their values become sufficiently large, they may creep into the range of the dynamic eigenvalues and we will no longer be able to distinguish between the dynamic eigenvalues and these (very poor) approximations of zero. In this case, we have a "spurious mode" – an eigenvalue and associated eigenvector in the

high-frequency range, but not satisfying the divergence criteria and hence entirely unphysical.

There was significant confusion in some of the earlier papers on edge elements, as they were then known, and when reading some of these, one may find claims that edge elements entirely eliminate spurious modes. This is not correct – edge elements still compute these modes, but with better fidelity, so that they do not corrupt the desired range of eigenmodes. There have been other approaches which aim to eliminate the spurious modes entirely, but edge elements do *not* accomplish this.

Regarding deterministic problems, since $k_i \neq 0$ will have been set in a deterministic problem, the numerical process, now being capable of reproducing an irrotational mode spectrum, instead ensures that such a modal content is absent [2, p. 313].[9]

It is interesting that spurious modes were not encountered in the FDTD community. The reason is that the Yee grid implicitly satisfies Gauss' laws (the divergence criteria).

9.6 Vector (edge) elements

9.6.1 An historical perspective

What are now called vector elements, but were originally known as "edge-based" elements, date back to the 1980s in CEM, although the underlying ideas of the structure of the electromagnetic field date back to 1957 and what are known as Whitney forms. In 1980, the French mathematician J. C. Nedelec published a paper which has since become the canonical reference in this field [8] although, ironically, he did not define the edge-based element itself; instead, the paper investigates the structure of the polynomial spaces which the basis functions should span in a highly mathematical format, which is not readily accessible to electronic engineers. (He was clearly influenced by earlier ideas of Raviart and Thomas [9] and it is useful to read their paper before attempting to read Nedelec's.) Some of the earliest work in electrical engineering is due to Bossavit [10]; Barton and Cendes [11] were among the first to address high-frequency electromagnetics with edge elements and their derivation is the one now generally given. Another type of related element, also a vector element, was the hexahedral element, originally introduced by Welij in its lowest order straight-sided form in 1985 [12], and in

[9] There is another school of thought on this topic. It has been argued that the driven solution can be viewed as a sum of eigenvectors, and hence incorrect eigenvectors may also corrupt a deterministic problem [7, p. 408]. In any case, by either argument, edge elements also lead to better solutions for deterministic problems.

generalized form by Crowley, Silvester and Hurwitz in 1988 [13]. Cendes' subsequent work produced one of the first higher-order tetrahedral elements [14]. Webb and Forghani's work on hierarchal tetrahedral elements was the standard reference for many years [15], until succeeded by Webb's later work [16].

During the 1990s, many researchers made excellent use of these elements and also advanced the theory underlying them. The following is only a selection of the work: Lee and Mittra worked on cavity eigenvalue problems [17] (and this paper remains useful today, since it contains analytical expressions for the elemental matrices); Dibben and Metaxas used edge elements for time domain analysis [18]; Savage and Peterson introduced alternative higher-order tetrahedral elements in [3]; Jin, Volakis, Kempel and their students made significant contributions to applications, especially cavity backed patches (this work is well summarized in [4, 19]), and also in new hierarchal elements [20]; Dyczij-Edlinger, Peng and Lee made advances in understanding the impact of the low-frequency ill-conditioning of the curl-curl formulation [21]; Graglia, Wilton and Peterson made progress with interpolatory as opposed to hierarchal elements [22]; and the present author extended work on waveguide analysis using higher-order mixed and complete elements [23], and with Botha, worked on error estimation [24].

9.6.2 Theory of vector elements

With this historical background, we now return to the elements. Before we study them in detail, we will first look at the impact they had on CEM. Although much of the early literature concentrates on the "spurious mode" problem, there are practical reasons which make these elements very useful in analysis. Firstly, for the lowest order elements, the degrees of freedom are proportional to the tangential electric field along an edge (and hence the widely used name, edge elements); we will show this shortly. Thus tangential continuity is very simple to enforce. Secondly, flux continuity is a natural boundary condition. Thirdly, it is easier to model corners, or other regions where the field becomes singular, since there is no nodal value at the singularity. Finally, they greatly ameliorated the problems with spurious modes: we will return to this subsequently.

Vector elements are most easily introduced using a 2D vector element for the *rectangular* element, shown in Fig. 9.2. The field is approximated as:

$$\vec{E}_e \approx \sum_{i=1}^{4} \vec{N}_i^e E_i^e \tag{9.56}$$

Here, $\vec{N}_i^e$ is the *vector* basis function and E_i^e is a scalar *degree of freedom*, the tangential field along the ith edge in this case. The vector functions $\vec{N}_i^e$ are

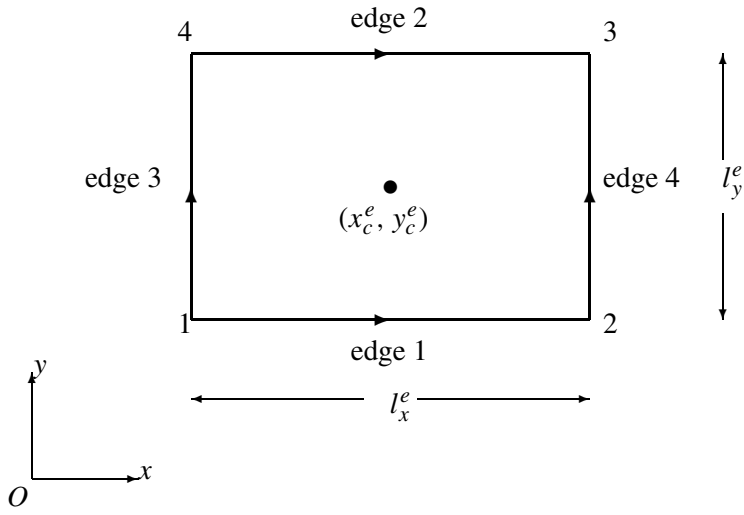


Figure 9.2 The rectangular edge element. Based on [4, Fig. 8.1].

given by:

$$\vec{N}_1^e = \frac{1}{l_y^e}\left(y_c^e - y + \frac{l_y^e}{2}\right)\hat{x}$$

$$\vec{N}_2^e = \frac{1}{l_y^e}\left(y - y_c^e + \frac{l_y^e}{2}\right)\hat{x}$$

$$\vec{N}_3^e = \frac{1}{l_x^e}\left(x_c^e - x + \frac{l_x^e}{2}\right)\hat{y}$$

$$\vec{N}_4^e = \frac{1}{l_x^e}\left(x - x_c^e + \frac{l_x^e}{2}\right)\hat{y}$$

with $(x_c^e; y_c^e)$ the coordinates of the center of the element, and l_x^e and l_y^e the element lengths in x- and y-directions respectively.

Now, note the following: $\vec{N}_1^e$ is zero on edge 2 (since $y = y_c^e + l_y^e/2$ everywhere on edge 2) and it is unity on edge 1; also, it is purely tangential ($\hat{x}$-directed) along this edge. On edges 3 and 4 it increases linearly from the top to the bottom, and it is purely normal ($\hat{x}$-directed) along these edges. One quickly establishes that $\vec{N}_2^e$ has the same properties, but with edges 1 and 2 interchanged, and that $\vec{N}_3^e$ and $\vec{N}_4^e$ also have similar properties, but obviously with x and y interchanged. In short, these basis functions provide a *mixed-order* approximation of the field – on the edges, the approximation is constant tangentially, and linear normally. (Indeed, these elements are frequently called CT/LN elements, constant tangential/linear normal.) Note also that due to these properties, E_1^e is the tangential field along

edge 1, and similarly E_2^e, E_3^e and E_4^e are the tangential fields along edges 2, 3 and 4 respectively. These are the degrees of freedom for this element. *Very importantly*, these properties permit enforcing tangential continuity *without* affecting the normal components, and this is precisely the boundary condition required by $\vec{E}$ or $\vec{H}$ fields, or indeed any 1-forms in the language of differential forms.

A mathematical aside – differential forms

Some of the work on vector elements uses the mathematics of differential forms – Bossavit is one of the main proponents of this [25]. Although the ideas can be readily understood without any knowledge of this field, it is useful to know a smattering of the terminology.

- 0-forms: this is a scalar function with functional but not derivative continuity, an example being the electric static potential ϕ.
- 1-forms: these are vector functions with tangential but not normal continuity, such as $\vec{E}$. These are also known as polar, or true, vectors, and are time-even under time reversal.
- 2-forms: these are vector functions with normal but not tangential continuity, such as $\vec{B}$. These are also known as axial vectors, or pseudo-vectors, and are time-odd under time reversal.
- 3-forms: discontinuous scalar functions, such as $\nabla \cdot \vec{D}$.

For an elegant discussion of polar versus axial vectors, and time symmetry, Feynman's chapter on this is a classic [26, Chapter 52].

Note that this element is *not* by design interpolatory, although for this lowest order element it can be made thus.[10] The degrees of freedom (E_1^e, E_2^e, E_3^e and E_4^e) represent field quantities along an edge; indeed, in Nedelec's original work, they are defined as integrals of the tangential field component along the edge, i.e. the average tangential field value. This is quite different to the nodal elements discussed earlier.

We should also comment that there are a variety of names for this element, including mixed order; "first" order; "half-th order", $H_0(\text{curl})$; and as already mentioned, constant tangential/linear normal (CT/LN). This last is especially insightful and is the present author's preference.

These elements have other additional significant properties. Interestingly, by taking $\hat{z} \times \vec{N}_i^e$, another class of elements is derived with the complementary property of providing *normal* continuity; these are useful for problems involving

[10] The degrees of freedom have been interpreted as the tangential field value at the center of the relevant edge by some researchers who have worked with interpolatory vector elements.

current, or 2-forms. Furthermore, we have already seen that the full-wave functional has a term of the form $\int_{S^e} \nabla \times \vec{E} \cdot \nabla \times \vec{E}\, dS$. It is important to note that $\hat{x}$-directed terms linear in x do *not* contribute to this term; i.e. these would be "wasted" degrees of freedom, which have been removed from these elements. This observation, at heart, was the core of Nedelec's contribution. Finally, *within* the element, the approximated $\vec{E}$ field has *zero divergence*. (Recall that this is not explicitly enforced in the curl-curl functional.) Because the spurious modes are associated with solutions with non-zero divergence, many early papers on vector elements concentrated on this property. Whilst low-order vector elements are indeed divergence free *within* the elements, the divergence is discontinuous at element boundaries, and furthermore, a number of successful vector elements are *not* divergence free. (Indeed, an argument has be made that since one is *not* removing the spurious modes, but computing them more accurately, the element should not be divergence free!) The superior suppression of spurious modes is now understood to be due to a better approximation of the null-space of the vector wave equation, that is, the zero frequency solutions we discussed above. The vector elements do a better job of representing these static $\nabla\phi$ eigenmodes; the reason is that the tangential-continuity-only of the vector elements admits a larger number of functions in the null-space. We noted earlier that ϕ should be continuous, implying that $\nabla\phi$ must be tangentially continuous (which is all that is imposed by edge elements), but the natural boundary condition permits the normal derivative to exhibit the correct jump discontinuity at material interfaces. Webb's 1993 paper remains one of best discussions of edge-based elements [27].

9.6.3 Vector elements on triangles – the Whitney element

Our preceding discussion considered rectangular elements. As mentioned on several occasions, one of the main advantages of the FEM over the FDTD is the geometrical modelling flexibility afforded by triangular and tetrahedral elements in two and three dimensions respectively, so it is important to understand how the same properties can be obtained for these types of elements.

Vector elements on simplicial elements are defined in terms of simplex coordinates. Again, these have acquired a variety of names during their development, including Whitney, Nedelec, Bossavit or simply edge-based elements. In its lowest order form, the element has the following definition:

$$\vec{w}_{ij} = \lambda_i \nabla\lambda_j - \lambda_j \nabla\lambda_i \tag{9.57}$$

There are three such elements per triangle, or six per tetrahedron, each associated with the edge from node i to node j, as will now be demonstrated.

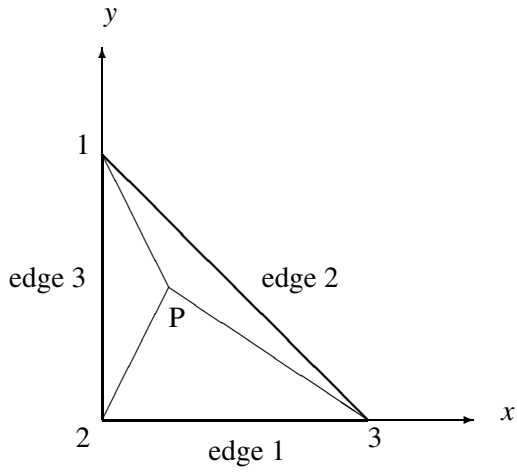


Figure 9.3 The right-angled parent triangle.

The Whitney element is the basis for *all* vector simplicial elements, both interpolatory and hierarchal, so its properties are of great importance. Firstly, an obvious question is, why does it have this specific form? To answer this, it is useful to study the right-angled triangle shown in Fig. 9.3, of unit length along the x- and y-axes. (It is also a useful exercise in understanding simplex coordinates.) The simplex coordinates are the ratios as follows:

$$\begin{aligned}\lambda_1 &= \frac{\text{area}_{\triangle \text{P23}}}{\text{area}_{\triangle 123}} \\ &= \frac{1/2 \text{ base} \times \text{height}}{1/2} \\ &= y \end{aligned} \tag{9.58}$$

since the area of triangle 123 is 1/2, and the base of triangle P23 is unity and its height is y.

Similarly,

$$\begin{aligned}\lambda_2 &= 1 - (x + y) \\ \lambda_3 &= x \end{aligned} \tag{9.59}$$

The expression for λ_2 is easily derived from the property $\sum_{i=1}^{3} \lambda_i = 1$. Now that we have explicit expressions for the simplex coordinates, their gradients follow trivially:

$$\nabla\lambda_1 = \hat{y} \tag{9.60}$$

$$\nabla\lambda_2 = -\hat{x} - \hat{y} \tag{9.61}$$

$$\nabla\lambda_3 = \hat{x} \tag{9.62}$$

We note that $\nabla\lambda_1$ is normal to edge 1 (that is, the edge opposite node 1), and similarly $\nabla\lambda_2$ and $\nabla\lambda_3$ are normal to edges 2 and 3 respectively.

Now, the Whitney functions can be written in explicit Cartesian form as follows:

$$\begin{aligned}\vec{N}_1 = \vec{w}_{23} &= \lambda_2\nabla\lambda_3 - \lambda_3\nabla\lambda_2 \\ &= (1-x-y)\hat{x} - x(-\hat{x}-\hat{y}) \\ &= (1-y)\hat{x} + x\hat{y} \\ \vec{N}_2 = -\vec{w}_{13} &= -y\hat{x} + x\hat{y} \\ \vec{N}_3 = \vec{w}_{12} &= -y\hat{x} + (-1+x)\hat{y}\end{aligned} \tag{9.63}$$

These are illustrated in Fig. 9.4.

Due to the simple form of these functions on this right-angled parent element, we can immediately establish some of the crucial features of these functions. Let us focus on $\vec{N}_1 = \vec{w}_{23}$. Along edges 2 and 3, this function is purely normal, and increases linearly from node 1 to node 2 along edge 3, and similarly from node 1 to node 3 along edge 2. Along edge 1, it has both tangential and normal components. These are easily separated on this right-angled parent element; on edge 1, they are the $\hat{x}$ and $\hat{y}$ components respectively, that is, $(1-y)|_{y=0} = 1$ and x respectively. Thus, on this edge, the tangential component is constant, and the normal component is linear. In short, $\vec{N}_1 = \vec{w}_{23}$ is a basis function with a constant tangential component on edge 1, and linear normal components along all the edges. The same is easily shown for the other two basis functions. Hence, this Whitney element has the same mixed-order CT/LN behavior as the rectangular element studied earlier. Furthermore, suitable degrees of freedom are again the average tangential fields along each edge. It is also immediately obvious from Eq. (9.63) that the divergence of the Whitney functions is zero.

An important note: although we have established these properties on a right-angled parent element, they are *generally* true for Whitney elements on *any* triangle; we will not however show this now. (Some further discussion on the Whitney element may be found in Appendix A.)

Another important point: what of the normal field components? The boundary condition in this case is normal *flux* continuity; it turns out that this is a natural boundary condition of the variational process, and hence is automatically satisfied at material interfaces [4, Section 5.8.3].

It is an interesting question to ask *why* this function might originally have been proposed. Firstly, as already noted, the gradient of a simplex coordinate is *constant*, and is directed perpendicular to the edge opposite the relevant node. Hence, using the gradient of the simplex coordinates promises a method to separate normal and tangential components, which it will be recalled is highly desirable, due to the

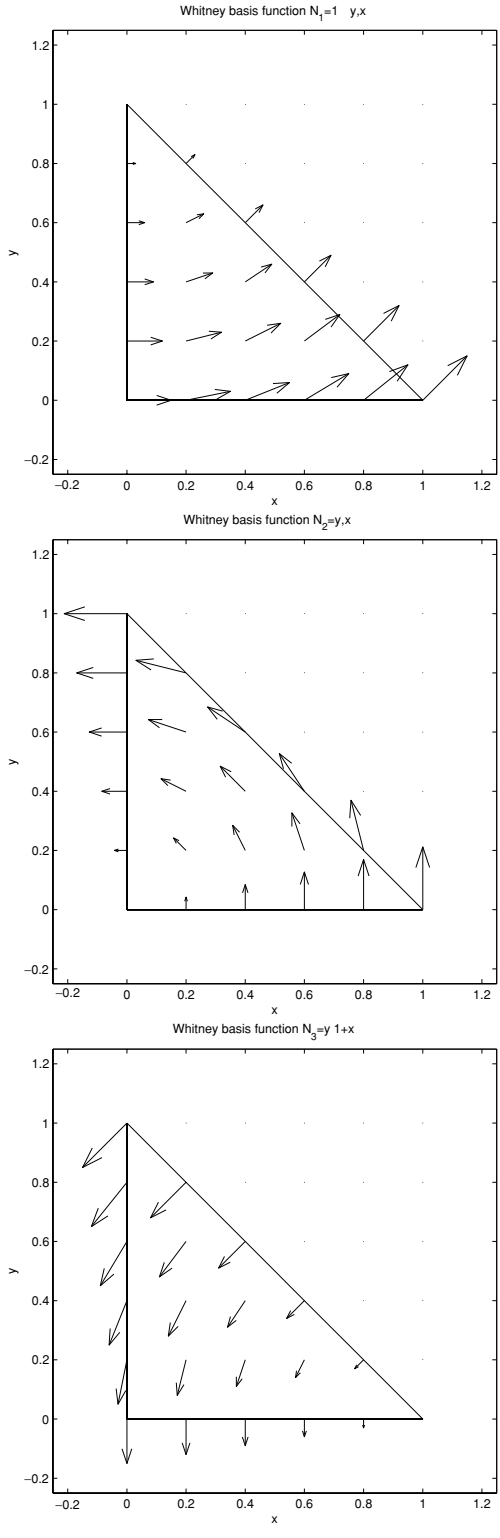


Figure 9.4 The three Whitney basis functions for triangles.

continuity requirements of vector fields. Now, using $\nabla\lambda_i$ alone can only given a constant approximation; multiply by λ_j and a linear form is obtained. To make this linear *along* an edge, *both* non-zero simplex coordinates are needed. Thus $\lambda_i\nabla\lambda_j \pm \lambda_j\nabla\lambda_i$ is a reasonable guess. The $+$ form can be rewritten as $\nabla(\lambda_i\lambda_j)$, which is in the null-space of the curl operator (the first term in the functional), hence the $-$ form is a good guess. Our detailed analysis above on the right-angled parent element confirms this.

In closing this introductory discussion on Whitney elements, it is very important to note that that the vector field can *only* be recovered by the vector sum of the three vector basis functions and the appropriate amplitudes (the degrees of freedom which the finite element procedure yields); the degrees of freedom lose the convenient interpretation of nodal elements as a field component value at a node.

Whitney elements revolutionized HF FEM analysis from the mid 1980s on; Ansoft's Eminence package (now HFSS) was one of the first commercial codes to exploit these elements for the three-dimensional finite element analysis of high-frequency devices. Extending the elements to higher order has been a controversial topic; many different forms have been published. The most comprehensive publication in the electrical engineering literature is Webb's relatively recent work [16]. A comparison of a number of these elements has been published by the present author [23]. This is discussed in Chapter 10.

9.7 Application to waveguide eigenvalue analysis

9.7.1 The two-dimensional variational functional for homogeneous waveguide

Waveguide eigenanalysis is one of the classic applications of the FEM. It is useful in its own right, but also serves as an excellent tool to illustrate the application of vector elements. We will analyze a rectangular waveguide, homogeneously filled, since then the eigenmodes split into pure TE and TM modes; an inhomogeneously loaded waveguide requires a more complex approach since the propagation modes are then hybrid in nature. (A discussion of this and suitable formulation may be found in [4, Section 8.2].) The functional for the transverse field components, *subject to the prescribed boundary condition* $\hat{n} \times \vec{E}_t = 0$, is:

$$F(\vec{E}_t) = \frac{1}{2}\int\int_S \left[\frac{1}{\mu_r}(\nabla_t \times \vec{E}_t)\cdot(\nabla_t \times \vec{E}_t) - k_i^2\epsilon_r \vec{E}_t \cdot \vec{E}_t\right] dS \qquad (9.64)$$

This is Eq. (9.45), with $E_z = 0$, i.e. no *longitudinal* field components, and again assuming lossless materials. The eigenvectors of this eigenvalue problem are the

TE modes, and the eigenvalues k_i are the corresponding cut-off wavenumbers,[11] with $k_z = 0$. ∇_t is the transverse del operator.

It is important to note that the prescribed boundary conditions, which amount to the edges lying on the PEC, *must be explicitly enforced*. This implies that the vector of unknowns, $\{e\}$, in the generalized eigenvalue problem:

$$[S]\{e\} = k^2[T]\{e\} \tag{9.65}$$

would appear to include *prescribed, i.e. zero*, values. This is incorrect. It may be shown that this equation includes *only* contributions from the free edges, i.e.

$$[S_{ff}]\{e_f\} = k^2[T_{ff}]\{e_f\} \tag{9.66}$$

To derive this, write the discretized functional *before it is rendered stationary* as:

$$F\{e\} = \{e_f e_p\}^T \begin{bmatrix} S_{ff} S_{fp} \\ S_{pf} S_{pp} \end{bmatrix} \{e_f e_p\} \tag{9.67}$$

Now, differentiating with respect to the free edges, and then applying the prescribed boundary condition $\{e_p\} \equiv 0$, one obtains Eq. (9.66).

This does of course require (globally) numbering the free edges first, and then the prescribed edges. If using a connection matrix approach, another renumbering matrix could be used afterwards to implement this. Alternatively, during matrix assembly, any entries corresponding to prescribed edges can simply be removed from the system.

If the TM modes are sought, then Eq. (9.64) must be solved with $\vec{H}_t$ as the working variable. In this case, homogeneous Neumann boundary conditions are appropriate – i.e. *no* explicit boundary conditions need be set at all.

This problem is especially easy to solve using rectangular elements, but since we would like to illustrate the application of the Whitney elements, we will use triangular elements. Firstly, we will need a mesh of such elements, but we will defer consideration of this until later, and concentrate initially on the theoretical analysis. For each element, we need the elemental (that is, stiffness $[S]$ and mass $[T]$) matrix elements. Using simplex coordinates, we can evaluate these quite easily.

9.7.2 Explicit formula for the elemental matrix entries

Before deriving expressions for the elemental matrices, it is worth briefly reviewing the two approaches which have been used. The approach we will use is essentially a direct approach, where we evaluate the simplex coordinates in terms of

[11] This is a special case of the more general functional [4, Eq. 8.36], which includes non-zero values of k_z.

the Cartesian coordinates of the actual element. The other approach uses the right-angled parent element of Fig. 9.3, and computes the matrices for this element; a coordinate transformation is then performed to the actual element, and the inverse of the Jacobian of this transformation is used to scale the matrix elements. The former approach is that of Lee and Mittra, who published some of the first explicit formulas in [17] for tetrahedral CT/LN elements (these formulas were extended by the present author to diagonally anisotropic materials in [28]), Savage and Peterson, who presented a very useful alternative formulation in [3], and Jin [4]. The latter approach is best exemplified by [7]. Savage and Peterson's approach leads to particularly compact expressions, and is the one we will use here. The following is based on their work, but simplified to triangles, using notation consistent with that of this chapter, and using the standard Whitney elements. (Savage and Peterson further scale the elements by the edge lengths.)

Recall that the variational formulation requires the evaluation of two matrices:

$$S_{ij} = \iint_S \nabla_t \times \vec{N}_i \cdot \nabla_t \times \vec{N}_j \, dS \tag{9.68}$$

and

$$T_{ij} = \iint_S \vec{N}_i \cdot \vec{N}_j \, dS \tag{9.69}$$

With z the direction of propagation, the $\nabla_t \times$ and ∇_t operators reduce to the two-dimensional operators in the (x, y) plane, which we will imply in the following.

The CT/LN elements are given by $\vec{N}_i = \vec{w}_{i1,i2} = \lambda_{i1}\nabla\lambda_{i2} - \lambda_{i2}\nabla\lambda_{i1}$ per edge. Here, $i1$ and $i2$ are the endpoints of edge i. The local triangular numbering scheme is as already discussed.

Now, the three simplex coordinates λ_i are given by

$$\lambda_i = a_i + b_i x + c_i y \tag{9.70}$$

and the gradient thereof by

$$\nabla\lambda_i = b_i \hat{x} + c_i \hat{y} \tag{9.71}$$

The actual coefficients $\{a_i; b_i; c_i\}$ may be computed by inverting the coordinate matrix

$$\begin{bmatrix} b_1 & c_1 & a_1 \\ b_2 & c_2 & a_2 \\ b_3 & c_3 & a_3 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix}^{-1} \tag{9.72}$$

This equation may be obtained by writing Eq. (9.70) for each node i and noting that $\lambda_i = 1$ at node i. Now the following two vectors are defined for nodes i

and j:

$$\begin{aligned}\vec{v}_{ij} &= \nabla\lambda_i \times \nabla\lambda_j \\ &= \hat{z}(b_i c_j - b_j c_i) \\ &= -\vec{v}_{ji}\end{aligned} \tag{9.73}$$

This vector is easily computed once $\{b_i; c_i\}$ are known. Similarly we define

$$\begin{aligned}\phi_{ij} &= \nabla\lambda_i \cdot \nabla\lambda_j \\ &= b_i b_j + c_i c_j\end{aligned} \tag{9.74}$$

Note that both $\vec{v}_{ij}$ and ϕ_{ij} are constant within a triangle, and hence may be taken outside integrals in which they appear.

Consider the evaluation of the curl-curl term, Eq. (9.68):

$$\begin{aligned}\nabla \times \vec{N}_i &= \nabla \times (\lambda_{i1}\nabla\lambda_{i2} - \lambda_{i1}\nabla\lambda_{i2}) \\ &= \nabla \times (\lambda_{i1}\nabla\lambda_{i2}) - \nabla \times (\lambda_{i2}\nabla\lambda_{i1}) \\ &= 2\nabla\lambda_{i1} \times \nabla\lambda_{i2} \\ &= 2\vec{v}_{i1,i2}\end{aligned} \tag{9.75}$$

From the second to third line in the above, the vector identities $\nabla \times (\phi\vec{A}) = \phi\nabla \times \vec{A} + \nabla\phi \times \vec{A}$ and $\nabla \times \nabla\phi \equiv 0$ have been used.

Using this, Eq. (9.68) becomes:

$$\begin{aligned}S_{ij} &= 4\iint_S \vec{v}_{i1,12} \cdot \vec{v}_{j1,j2}\, dS \\ &= 4A\vec{v}_{i1,12} \cdot \vec{v}_{j1,j2}\end{aligned} \tag{9.76}$$

Note that the widely used expression for element area in terms of the determinant of the coordinate matrix,

$$2A' = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \tag{9.77}$$

actually yields a potentially signed area A', whose sign depends on the sense (clockwise or anticlockwise) of the coordinate numbering. A in the above is the *unsigned* area of the element, that is $A = |A'|$.

The second term that appears in Eq. (9.69) requires the computation of dot products:

$$\begin{aligned}\vec{N}_i \cdot \vec{N}_j &= (\lambda_{i1}\nabla\lambda_{i2} - \lambda_{i2}\nabla\lambda_{i1}) \cdot (\lambda_{j1}\nabla\lambda_{j2} - \lambda_{j2}\nabla\lambda_{j1}) \\ &= [\lambda_{i1}\lambda_{j1}(\nabla\lambda_{i2} \cdot \nabla\lambda_{j2}) - \lambda_{i1}\lambda_{j2}(\nabla\lambda_{i2} \cdot \nabla\lambda_{j1}) \\ &\quad -\lambda_{i2}\lambda_{j1}(\nabla\lambda_{i1} \cdot \nabla\lambda_{j2}) + \lambda_{i2}\lambda_{j2}(\nabla\lambda_{i1} \cdot \nabla\lambda_{j1})]\end{aligned} \tag{9.78}$$

Using the notation of Eq. (9.74), this can be written as:

$$\vec{N}_i \cdot \vec{N}_j = [\lambda_{i1}\lambda_{j1}\phi_{i2,j2} - \lambda_{i1}\lambda_{j2}\phi_{i2,j1} - \lambda_{i2}\lambda_{j1}\phi_{i1,j2} + \lambda_{i2}\lambda_{j2}\phi_{i1,j1}] \quad (9.79)$$

Thus the associated matrix elements become:

$$\begin{aligned} T_{ij} = \phi_{i2,j2} \iint_S \lambda_{i1}\lambda_{j1}\, dS - \phi_{i2,j1} \iint_S \lambda_{i1}\lambda_{j2}\, dS \\ -\phi_{i1,j2} \iint_S \lambda_{i2}\lambda_{j1}\, dS + \phi_{i1,j1} \iint_S \lambda_{i2}\lambda_{j2}\, dS \end{aligned} \quad (9.80)$$

Using the general integration formula for integrals in simplex coordinates [2, p. 458]:

$$\iint_S \lambda_1^i \lambda_2^j \lambda_3^k\, dS = \frac{2!\; i!\; j!\; k!}{(2+i+j+k)!} A \quad (9.81)$$

the expression for T_{ij} may be simplified (note that $0! \equiv 1$). In Eq. (9.80), each integral involves integration over two simplex coordinates, possibly identical. These can be expressed in matrix form as

$$M_{ij} = \iint_S \lambda_i \lambda_j\, dS = \frac{1}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad (9.82)$$

Using this, Eq. (9.80) reduces to

$$\begin{aligned} T_{ij} = A[\phi_{i2,j2} M_{i1,j1} - \phi_{i2,j1} M_{i1,j2} \\ -\phi_{i1,j2} M_{i2,j1} + \phi_{i1,j1} M_{i2,j2}] \end{aligned} \quad (9.83)$$

9.7.3 Coding

We now have all the theory we need. However, finite element codes require a lot of "house-keeping" – the unstructured nature of finite element meshes is both their strong point (permitting very accurate local geometrical modelling) and a significant complication (since a lot of lists need to be generated and the maintained). We will now discuss a number of these issues.

Edge and node numbering schemes

With an FEM code, adopting sensible local and global numbering conventions *and then using these consistently* is absolutely essential. The local edge numbering scheme we discussed earlier (whereby the edge number corresponds to the node opposite) is not widely used in practice. The following is the most widely used in the literature:

Edge	Local edge number
e_{12}	1
e_{13}	2
e_{23}	3

In the above, e_{ij} is the edge directed from node i to node j. It is important to note that although the degree of freedom associated with the edge is a scalar, it is nonetheless *signed*.

A convention that can be recommended is *first* to sort the nodes in each element into ascending global order. This ensures that when edges are assigned, they are always directed from lower to higher node numbers, and thus the edges shared by two or more elements always have the same sign. All the local edge numbering schemes in use in the literature are consistent [3, 17] (taking into account that some number from 0 and some from 1). (Note that the *sign* of the edges is not, however, consistent: for example, edge 3 in [4, Fig. 8.2] has the opposite sense to that above.)

Global edge numbers are assigned from 1 upwards; within an element, global edges are incremented in the same pattern as the local edges. To illustrate this by example, element $e1$ will always contain edges 1, 2 and 3 (although not necessarily global nodes 1, 2, 3 and 4, of course, since these are assigned by the mesher); if element $e2$ shares its first edge with element $e1$, then its remaining edges will be globally numbered 4 and 5, local edges e_{13}^{e2} and e_{23}^{e2} respectively.

The above sounds more complex than it is, as is often the case with finite element data structures, and becomes clear when coding.

Data structures

Before programming starts, it is useful to establish the major data structures that will be needed. For a mesh with N_n nodes, N_e elements, and E edges, the major data structures required will include at least the following.

vertices Dimensioned as $(N_n, 2)$. This stores the (x, y) coordinates of each vertex (node).

nodes Dimensioned as $(N_e, 3)$. This stores the three nodes associated with each element.

edge_nodes Dimensioned as $(E, 2)$. This stores the global nodes that each edge connects.

materials Dimensioned as (N_e). This stores the material number. Another (usually very much smaller) data structure will be required to store the actual constitutive parameters for each material.

dof Dimensioned as E for Whitney elements. These are the degrees of freedom.

Two major data structures omitted here (deliberately) are the $[S]$ and $[T]$ matrices for the system. For initial work, these can simply be stored as full matrices, but to exploit fully the power of the FEM, sparse storage schemes must be used. This is discussed in Chapter 10.

The above data structures are accessed so frequently that they should be globally accessible. In MATLAB, this is done using the `global` statement. In FORTRAN 90, one uses modules.

Meshing

For the beginner, this often seems the most challenging task. For 2D problems however, one can build quite satisfactory meshes by hand. The easiest way of generating triangular meshes for a rectangular domain is first to divide the domain into smaller rectangles, and then to split each of these further into two triangles. An example of such a mesh is shown in Fig. 9.5. (Also shown on this plot are global node and element numbers; the manner in which these are assigned is essentially

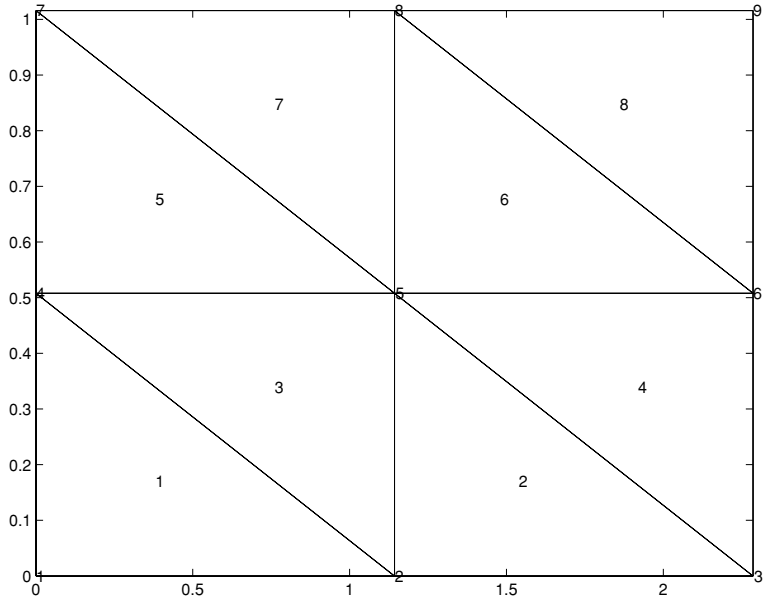


Figure 9.5 An eight-element triangular mesh.

arbitrary, and the finite element code should be able to handle this.) It is also easy to automate this type of meshing procedure.

Book-keeping

The issue of making the edges has already been discussed. The book-keeping required does not end here, however. One also needs connection information (the equivalent of the connection matrix discussed earlier). For a "regular" triangular mesh such as that of Fig. 9.5, it is clear that an edge can be connected to at most two triangles, but in general, no such assumption can be made.

Building the interconnectivity data is primarily a problem in list-searching. The simplest method of doing this is for *each* edge, to search through all elements and see whether the edge nodes coincide. This is *not* a good idea for large meshes, since this is an $\mathcal{O}(EN) \sim \mathcal{O}(N^2)$ operation, but for small meshes it works. Real codes use additional node-element lists to accelerate the search.

One also needs some type of renumbering scheme, so that the free edges may be numbered first. An approach which works is first to flag each edge as free or prescribed. In the present case, simply checking whether the nodal coordinates of the edge coincide with $x = 0$, $x = a$, $y = 0$ or $y = b$ is sufficient, but in general this can also be quite a complex search. Once this has been done, an index list is then built which gives the original global edge number for each degree of freedom. Again, this sounds more complex than it actually is. With these data, and with the convention that shared edges have the same sign, matrix assembly proceeds very quickly.

Solving the eigenvalue problem

From a mathematical viewpoint, the most complex part of the finite element analysis (and certainly the most computationally expensive) is actually the solution of the generalized eigenvalue problem represented by Eq. (9.49), repeated here:

$$[S]\{e_i\} = k_i^2[T]\{e_i\}$$

Fortunately, modern scientific programming environments such as MATLAB make this very simple; for instance, in MATLAB, the function `eig` solves this with one command! (Similar routines are available in LAPACK, if using FORTRAN 90 or C, although calling them requires a little more work.) What emerges from the analysis is a set of eigenvalues, each with its associated eigenvectors.

As should be anticipated from our earlier discussion, this vector element FEA includes static modes. (This very important point is often not mentioned explicitly, and causes novices no end of problems.) Interestingly, it is possible to *predict* the number of such modes. The idea is the following. For the Whitney element, the curl of the field is represented by a constant. For the null-space of the eigenvalue

problem, where the field can be represented by a potential, this potential function must thus be linear. The obvious approximation of a linear potential using nodal elements would require one degree of freedom per unconstrained (free) node. One of the solutions is actually the trivial solution $\vec{E} = 0$ (corresponding to a constant potential) and must be discounted (since it is also a valid, albeit trivial, solution of the dynamic problem) and thus the dimension of the null-space, K, is the number of unconstrained nodes minus one for Whitney elements. Hence K can be very large. In the 2D case, the ratio of edges to nodes tends to around three, so almost one-third of computed eigenvalues are actually null-space ones.

In practice, the trivial solution is also irrelevant. So, once the eigenvalue problem has been solved, we must first sort the computed eigenvalues into ascending order, then count the number of free *nodes*, i.e. $K + 1$, and then finally, eigenvalue $K + 2$ is the first eigenvalue of interest. (Again, this type of operation is very easily implemented in MATLAB, using the `sort` function.)

Post-processing

Once the finite element analysis is complete, the vector degrees of freedom need to be post processed to yield meaningful field data. As has been commented previously, unlike interpolatory nodal-based elements, where a degree of freedom typically represents a field component at a particular node, hierarchal vector elements only reconstruct a physically meaningful field when summed together. In this case, the eigenvector corresponding to a particular eigenvalue does not in itself directly represent a field. Given the degrees of freedom and the corresponding basis functions, the field $\vec{E}(x, y)$ can be computed at any point within the element.

For this, one needs to compute directly the sum of the Whitney elements within each element, that is:

$$\vec{E}^e(x, y) = E^e_{12}\vec{w}_{12} + E^e_{13}\vec{w}_{13} + E^e_{23}\vec{w}_{23} \tag{9.84}$$

with E^e_{ij} the degrees of freedom and w_{ij} the basis functions. (Here, it is worthwhile pointing out that some authors include the appropriate edge lengths in the basis function, e.g. $w_{ij} = \ell_{ij}(\lambda_i \nabla\lambda_j - \lambda_j \nabla\lambda_i$.) The reason this is sometimes done is that the degree of freedom is then the tangential field at each edge. In this case, the $[S]$ and $[T]$ matrix entries are scaled appropriately [3], and the basis functions in Eq. (9.84) must of course *also* include the edge length. This is obvious, but easy to overlook, since the lengths are often implied but not consistently retained in some of the literature.)

All the theory needed for this has already been presented. The simplex coordinates for point (x, y) are computed from its basic definition as in Eq. (9.40),

expanded here for all three coordinates:

$$\lambda_1 = \frac{\begin{vmatrix} 1 & x & y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}}{2A}$$

$$\lambda_2 = \frac{\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x & y \\ 1 & x_3 & y_3 \end{vmatrix}}{2A}$$

$$\lambda_3 = \frac{\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x & y \end{vmatrix}}{2A} \tag{9.85}$$

The gradients are computed as in Section 9.7.2, using specifically Eq. (9.71).

9.7.4 Results

The eigenvalues can be put into one-to-one correspondence with the analytically known eigenvalues. For a standard X-band guide, with $a = 22.86$ mm and $b = 10.16$ mm, the first eight TE eigenmodes are listed in Table 9.1.

The relative error of the eigenvalues computed with the FEM compared to the analytical results is shown in Fig. 9.6. Clearly, refining the mesh has the desired result of decreasing the error. Individually, the eigenmodes display different convergence with, for instance, the seventh eigenmode (TE_{31}) being accurately computed

Table 9.1 *First eight transverse electric modes in a standard X-band waveguide, giving cut-off wavenumber and frequency*

Mode	k_c (rad/m)	f_c (GHz)
TE_{10}	137.43	6.5573
TE_{20}	274.86	13.1146
TE_{01}	309.21	14.7539
TE_{11}	338.38	16.1455
TE_{30}	412.28	19.6719
TE_{21}	413.71	19.7401
TE_{31}	515.35	24.5899
TE_{40}	549.71	26.2292

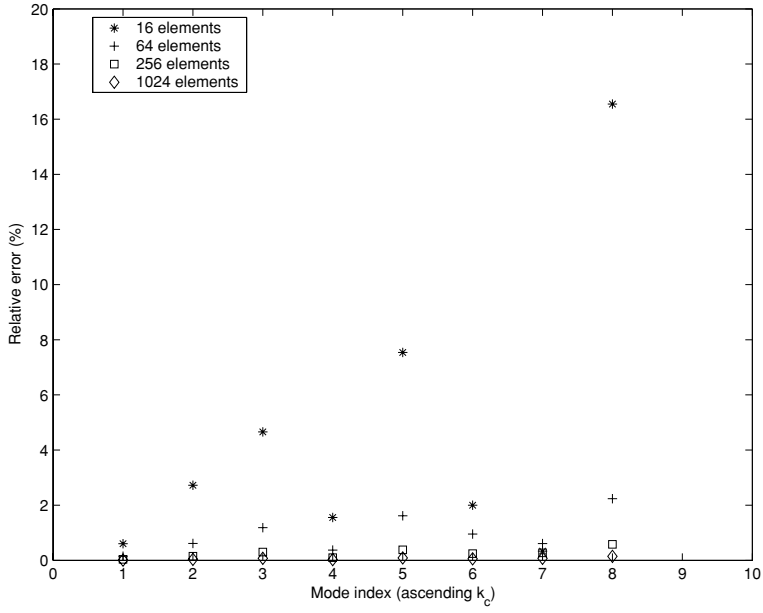


Figure 9.6 The relative error in the first eight eigenmodes.

by even the very coarse 16 element mesh. This behavior has been observed in many implementations, and what is usually studied is an average error. In Fig. 9.7, the result for the RMS error of the first eight eigenmodes is plotted versus average triangle length h. Theoretically, the Whitney element is complete to zeroth order, so the error term should be of $\mathcal{O}(h)$. Since the functional depends on the square of the field, and is stationary at the true solution, the resulting error is $\mathcal{O}(h^2)$. We can confirm this on the log-log plot; this is (approximately) a straight line, with slope 2.14 (this can be conveniently obtained using the MATLAB function `polyfit`). Hence the error E is:

$$E = Kh^2 \tag{9.86}$$

where K is an unknown coefficient. This is a well-known result in finite element analysis [2, p. 148] (note that the exponent has the incorrect sign in this reference). It is also confirmed by the interpolation error bound of ch^k, with c a constant and $k = 1$ in this case, originally given by Nedelec [8, Eq. 22] (although this is not exactly the same as the overall error, which is what we are evaluating) if one recalls that the eigenvalue, as a stationary property, is the square of this estimate. (Morishita and Kumagai showed that with the curl-curl functional, the eigenvalue is stationary [29, Section IV]; this is also discussed by Chen and Lien [30].)

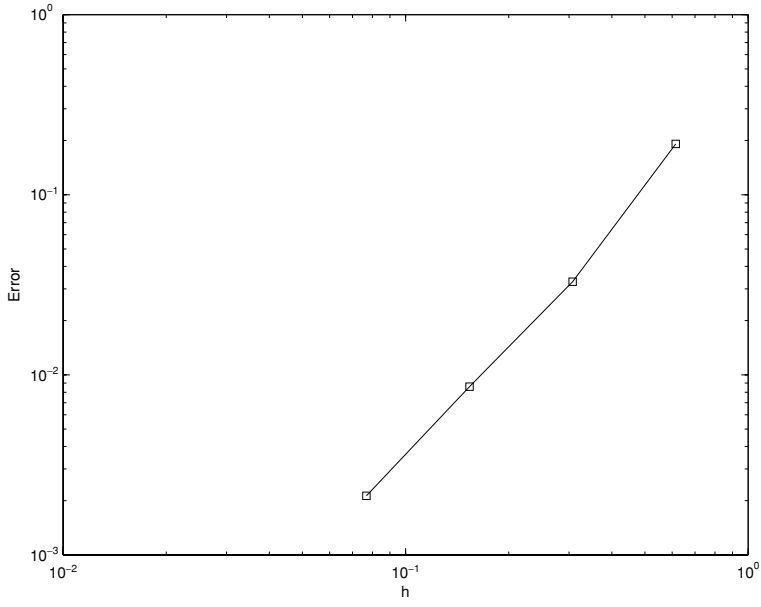


Figure 9.7 RMS error in first eight eigenmodes versus average mesh size h.

The eigenmodes are conveniently compared visually. Figures 9.8 and 9.9 show the first six eigenmodes, computed analytically and with a 256 element FEM solution respectively. These results were plotted with the MATLAB `quiver` function.[12] Note that the sign of the eigenmode is essentially arbitrary; for instance, the TE_{01} eigenmode has been computed with opposite sign by the analytical and finite element methods. Also, for interest, the first six "spurious" eigenmodes are shown in Fig. 9.10. The wavenumbers appear to be complex; this is simply due to taking the square root of numbers approximating zero, but slightly negative. There are 105 such eigenvalues and associated eigenmodes, in a problem with 360 degrees of freedom. One notes that, in general, these modes satisfy the boundary condition of zero tangential field, but cannot of course be recognized as traditional TE modes.

9.8 The three-dimensional Whitney element

The FEM using vector elements in 3D is in a sense just a straightforward extension of the 2D analysis; however, the mesh generation and book-keeping problems become formidable and we will not discuss the actual implementation of such a code in detail; developing a truly general-purpose 3D FEM code is a challenging

[12] Readers should note that this is a rather tricky function to use.

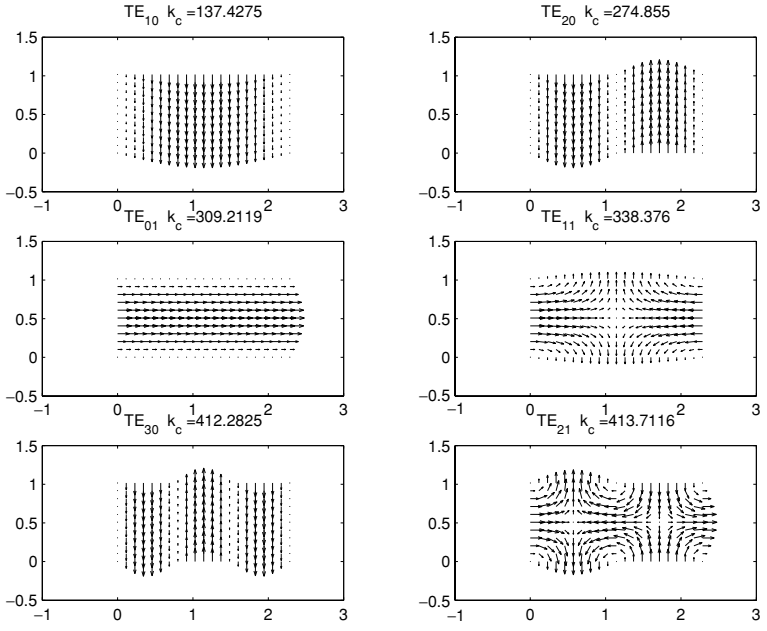


Figure 9.8 Quiver plot of the first six eigenmodes, computed analytically.

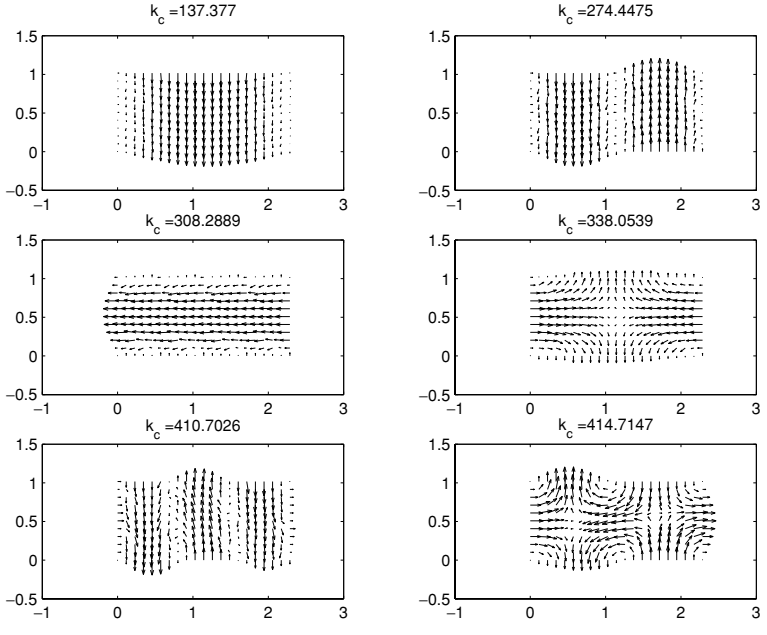


Figure 9.9 Quiver plot of the first six eigenmodes, computed with a 256 element FEM solution.

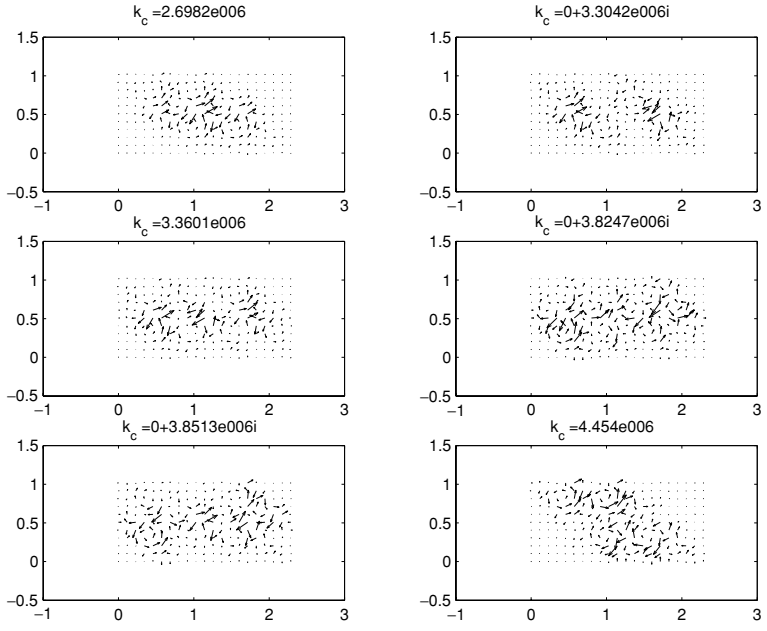


Figure 9.10 Quiver plot of the first six "spurious" eigenmodes, computed with a 256 element FEM solution.

task, although developing a special purpose 3D finite element analysis code is not an entirely unreasonable undertaking. References that can assist in this regard may be found in Section 9.9.

The three-dimensional Whitney element is exactly the same as in two dimensions

$$\vec{w}_{ij} = \lambda_i \nabla \lambda_j - \lambda_j \nabla \lambda_i \tag{9.87}$$

with the obvious difference that there are now six degrees of freedom per tetrahedron, rather than three per triangle, since a tetrahedron has six edges. This element has exactly the same well-known properties of constant tangential/linear normal field (CT/LN) approximation along edges (hence, of mixed order) as its two-dimensional counterpart and needs no further discussion. Once again, conventions should be adopted right from the start; since we are going to address higher-order elements later, which have degrees of freedom linked to faces as well as edges, we need also to number faces. See Table 9.2 for one such convention. (The face numbering conventions in the literature are generally not consistent. This one follows [3, Table II], but differs from [17], for example.)

All the comments made in the context of two-dimensional elements are equally germane here; however, the coding effort is at least an order of magnitude more,

Table 9.2 *Local edge and face numbering convention for 3D tetrahedrons*

Local edge numbering			Local face numbering			
Edge	Local nodes		Face	Local nodes		
1	1	2	1	1	2	3
2	1	3	2	1	2	4
3	1	4	3	1	3	4
4	2	3	4	2	3	4
5	2	4				
6	3	4				

due to the complexity of three-dimensional tetrahedral meshes and the much larger problem size required by realistic problems, and hence we conclude our introductory coverage at this point.

9.9 Further reading

This chapter has focussed heavily on vector finite elements; the explanations of the properties of the elements reflect what might be called the current orthodoxy. It should be mentioned that there has been criticism of these elements from some quarters, most stridently from Mur [31]. One should note that his criticism is heavily influenced by his work on magnetostatic problems, where the permeability can vary enormously from element to element and vector elements may indeed exhibit serious problems due to this. Recall also our earlier discussion about material interfaces and field continuity, and the problems with node-based elements, in Section 9.5; de Lager and Mur were able to introduce a node-based element which can indeed handle material discontinuities [32]. However, at the time of writing, this element had not been applied to 3D high-frequency analysis, and it seems likely that the current vector elements will continue to dominate finite element analysis in the forseeable future. On the topic of spurious modes, work by Vardapetyan and Demkowicz has addressed the problem at a quite fundamental level, introducing Lagrange multipliers in the functional; [33] is representative of their work.

More generally, the reader is fortunate that there are a number of excellent and current texts on the FEM available. Silvester and Ferrari's book [2] (first published in 1983, approximately doubling in length with the 1990 second edition, and increasing again in length significantly with the 1996 third edition) was for years the only reference in the field, and the current edition contains good coverage of high-frequency topics, in addition to extensive coverage of statics and magnetostatics. (Incidentally, the second edition contains some useful material which was

not in the third, and is also worth acquiring if the opportunity presents itself.) Jin's text has recently been revised [4] and is probably the book of first choice for high-frequency electromagnetics, which it concentrates on exclusively. Volakis, Chatterjee and Kempel's text also focusses on high-frequency applications, and contains much useful information on various elements [19]. Pelosi, Coccioli and Selleri's book lives up to its name, and is a good starter text [34]. Peterson, Ray and Mittra's book is somewhat more general in scope than just the FEM, but provides particularly deep coverage of coupled FEM/MoM formulations [35]. The text by Salazar-Palma *et al.* [7] is more of a research monograph; it concentrates primarily on interpolatory elements. The coverage is more theoretical than the other texts discussed here, and it is especially useful as preparatory reading if one intends working through mathematical papers such as Nedelec's.

Two other very useful sources are the 1996 anthology edited by Silvester and Pelosi [1]; the extensive annotations are especially useful for putting the work in perspective, and the anthology contains a number of earlier papers which are otherwise hard to come by. Some important papers have appeared since the anthology was published (and have been referenced in this chapter) but these are generally easily accessible. The collection edited by Itoh, Pelosi and Silvester [36] (also in 1996) contains a number of important contributions; in the context of vector elements, [6] deserves particular mention. For readers who would like to embark on their own three-dimensional implementation, there are two papers which will be of considerable interest, since they provide an eminently practical viewpoint on finite element coding. The first is by the present author [37]; the second reflects experience by Kempel's group [38], and was written specifically to complement the former. In [37], a number of practical issues are discussed, but mesh generation and linear algebra are only very briefly considered. In [38], an excellent overview of the many meshing packages available is provided, as well as a discussion of some sparse matrix solution routines. Sparse matrix schemes are on the one hand essentially an entirely practical problem, but on the other their efficient use is essential for commercial codes – we will briefly discuss this in the next chapter. The book by Duff *et al.* [39] is the standard reference on this. It has to be commented that specifically the topic of sparse matrices is not well treated in the CEM literature on finite elements.

9.10 Conclusions

This chapter has introduced the finite element method for high-frequency electromagnetic field solutions. Using primarily the variational formulation, the basic method was introduced for the scalar Laplace equation, following which we immediately addressed vector (edge) elements for the vector wave equation. An

eigenvalue problem was solved, and used to illustrate ideas both about the theory of finite element solutions of the vector wave equation, as well as a plethora of practical issues which one must address when writing an actual finite element code. Two-dimensional finite element codes require only moderate coding complexity, and it is quite realistic to attempt development of such a code oneself. The extension to three dimensions has been discussed. Without wishing to dissuade readers from attempting a full higher-order 3D finite element implementation using tetrahedral elements, we should caution that getting all the aspects required working together efficiently, and reliably, is no small undertaking. One way of "easing" into this would be to start with a "brick" mesh; the bricks can be subdivided into tetrahedra. (One way to do this was shown in [40, Fig. 9.5], unfortunately not repeated in the third edition.) Of course, this does not truly exploit the power of the FEM for modelling complex geometries. Another approach is to use prismatic meshes; Volakis, Kempel and their colleagues have been very successful with such meshes for a variety of antenna problems [19, 38]. One might term this $2\frac{1}{2}$D-modelling, although of course the full 3D field solution is obtained.

In the following chapter, a variety of more advanced topics on the FEM are introduced. Starting with the extension of vector elements to higher orders, the application of these will be illustrated by way of a deterministic problem (an obstacle in a rectangular waveguide, analyzed using both commercial and research codes). The FEM/MoM hybrid formulation will be introduced, and some results shown. Then a time domain formulation of the finite element method for the vector wave equation is outlined. The issue of sparse matrix storage schemes and solution methods is considered, before finishing the coverage with an introduction to the field of error estimation and mesh adaptation.

References

[1] P. P. Silvester and G. Pelosi, *Finite Elements for Wave Electromagnetics*. New York: IEEE Press, 1996.

[2] P. P. Silvester and R. L. Ferrari, *Finite Elements for Electrical Engineers*. Cambridge: Cambridge University Press, 3rd edn., 1996.

[3] J. S. Savage and A. F. Peterson, "Higher-order vector finite elements for tetrahedral cells," *IEEE Trans. Microwave Theory Tech.*, **44**, 874–879, June 1996.

[4] J. Jin, *The Finite Element Method in Electromagnetics*. New York: Wiley, 2nd edn., 2002.

[5] D. H. Norrie and G. de Vries, *An Introduction to Finite Element Analysis*. New York: Academic Press, 1978.

[6] A. F. Peterson and D. R. Wilton, "Curl-conforming mixed-order edge elements for discretizing the 2D and 3D vector Helmholtz equation," in *Finite Element Software for Microwave Engineering* (T. Itoh, G. Pelosi, and P. P. Silvester, eds.), Chapter 5. New York: Wiley, 1996.

[7] M. Salazar-Palma, T. K. Sarkar, L. E. García-Castillo, T. Roy, and Djordjević, *Iterative and Self-Adaptive Finite-Elements in Electromagnetic Modelling*. Boston, MA: Artech House, 1998.
[8] J. C. Nedelec, "Mixed finite elements in $\Re^3$," *Numerische Mathematik*, **35**, 315–341, 1980.
[9] P. A. Riavart and J. M. Thomas, "A mixed finite element method for 2nd order elliptic problems," in *Mathematical Aspects of the Finite Element Method* (I. Galligani and E. Mayera, eds.), vol. 606 of *Lecture Notes on Mathematics*, pp. 293–315. New York: Springer-Verlag, 1977.
[10] A. Bossavit, "Finite elements for the electricity equation," in *The Mathematics of Finite Elements and Applications* (J. R. Whiteman, ed.), pp. 85–91. London: Academic Press, 1982.
[11] M. L. Barton and Z. J. Cendes, "New vector finite elements for three-dimensional magnetic field computation," *J. Appl. Phys.*, **61**, 3919–2921, April 1987.
[12] J. S. van Welij, "Calculation of eddy currents in terms of H on hexahedra," *IEEE Trans. Magn.*, **21**, 2239–2241, 1985.
[13] C. W. Crowley, P. P. Silvester, and H. Hurwitz, "Covariant projection elements for 3D vector field problems," *IEEE Trans. Magn.*, **24**, 397–400, 1988.
[14] Z. J. Cendes, "Vector finite elements for electromagnetic field computation," *IEEE Trans. Magn.*, **27**, 3958–3966, September 1991.
[15] J. P. Webb and B. Forghani, "Hierarchal scalar and vector tetrahedra," *IEEE Trans. Magn.*, **29**, 1495–1498, March 1993.
[16] J. P. Webb, "Hierarchal vector basis functions of arbitrary order for triangular and tetrahedral finite elements," *IEEE Antennas Propag.*, **47**, 1244–1253, August 1999.
[17] J.-F. Lee and R. Mittra, "A note on the application of edge-elements for modeling three-dimensional inhomogeneously-filled cavities," *IEEE Trans. Microwave Theory Tech.*, **40**, 1767–1773, September 1992.
[18] D. C. Dibben and R. Metaxas, "Time domain finite element analyis of multimode microwave applicators," *IEEE Trans. Magn.*, **32**, pp. 942–945, May 1996.
[19] J. Volakis, A. Chatterjee, and L. Kempel, *Finite Element Method for Electromagnetics: Antennas, Microwave Circuits and Scattering Applications*. Oxford and New York: Oxford University Press and IEEE Press, 1998.
[20] L. S. Andersen and J. L. Volakis, "Development and application of a novel class of hierarchical tangential vector finite elements for electromagnetics," *IEEE Trans. Antennas Propag.*, **47**, 112–120, January 1999.
[21] R. Dyczij-Edlinger, G. Peng, and J. Lee, "A fast vector-potential method using tangentially continuous vector finite elements," *IEEE Trans. Microwave Theory Tech.*, **46**, 863–868, June 1998.
[22] R. D. Graglia, D. R. Wilton, and A. F. Peterson, "Higher order interpolatory vector bases for computational electromagnetics," *IEEE Trans. Antennas Propag.*, **45**, 329–342, March 1997.
[23] D. B. Davidson, "An evaluation of mixed-order versus full-order vector finite elements," *IEEE Trans. Antennas Propag.*, **51**, 2430–2441, September 2003.
[24] M. M. Botha, Efficient finite element electromagnetic analysis of antennas and microwave devices: the FE-BI-FMM formulation and a posteriori error estimation for *p* adaptive analysis. Ph.D. Thesis, Department of Electrical and Electronic Engineering, University of Stellenbosch, 2002.
[25] A. Bossavit, *Computational Electromagnetism: Variational Formulations, Complementarity, Edge Elements*. San Diego, CA: Academic Press, 1998.

[26] R. P. Feynman, R. B. Leighton, and P. Sands, *The Feynman Lectures on Physics*, vol. 1. Reading, MA: Addison-Wesley, 1963.
[27] J. P. Webb, "Edge elements and what they can do for you," *IEEE Trans. Magn.*, **29**, 1460–1465, March 1993.
[28] D. B. Davidson, "Comments on and extensions of "A note on the application of edge-elements for modeling three-dimensional inhomogeneously-filled cavities"," *IEEE Trans. Microwave Theory Tech.*, **46**, 1344–1346, September 1998.
[29] K. Morishita and N. Kumagai, "Unified approach to the derivation of variational expression for electromagnetic fields," *IEEE Trans. Microwave Theory Tech.*, **25**, 34–40, January 1977.
[30] C. H. Chen and C. Lien, "The variational principle for non-self-adjoint electromagnetic problems," *IEEE Trans. Microwave Theory Tech.*, **28**, 878–886, August 1980.
[31] G. Mur, "The fallacy of edge elements," *IEEE Trans. Magn.*, **34**, 3244–3247, September 1998.
[32] I. E. de Lager and G. Mur, "Generalized Cartesian finite elements," *IEEE Trans. Magn.*, **34**, 2220–2227, July 1998.
[33] L. Vardapetyan and L. Demkowicz, "hp-adaptive finite elements in electromagnetics," *Comput. Methods Appl. Mech. Eng.*, **169**, 331–344, 1999.
[34] G. Pelosi, R. Coccioli, and S. Selleri, eds., *Quick Finite Element Method for Electromagnetic Waves*. Boston, MA: Artech House, 1998.
[35] A. F. Peterson, S. L. Ray, and R. Mittra, *Computational Methods for Electromagnetics*. Oxford and New York: Oxford University Press and IEEE Press, 1998.
[36] T. Itoh, G. Pelosi, and P. P. Silvester, eds., *Finite Element Software for Microwave Engineering*. New York: Wiley, 1996.
[37] D. B. Davidson, "Implementation issues for three-dimensional vector FEM programs," *IEEE Antennas Propag. Soc. Mag.*, **42**, 100–107, December 2000.
[38] A. Awadhiya, P. Barba, and L. Kempel, "Finite element method programming made easy???," *IEEE Antennas Propag. Soc. Mag.*, **45**, 73–79, August 2003.
[39] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*. Oxford: Oxford University Press, 1986.
[40] P. P. Silvester and R. L. Ferrari, *Finite Elements for Electrical Engineers*. Cambridge: Cambridge University Press, 2nd edn., 1990.

10

A selection of more advanced topics on the finite element method

In this final chapter, we discuss a selection of more advanced topics, primarily relating to the finite element method. However, as will be seen, a linkage to the method of moments will be established, and perhaps rather less expectedly, the finite difference time domain method will also emerge as a special case of a finite element time domain treatment, so amongst other purposes, the chapter serves to draw together these three apparently quite different methods.

We will start by considering a very important extension of the vector elements, namely higher-order elements. Following this, the stationary functional formulation for deterministic (driven) problems will be outlined. In the preceding chapter, an eigenvalue problem was used to illustrate the FEM in two dimensions; in this chapter, a deterministic three-dimensional problem will be discussed, namely the analysis of waveguide obstacles. Finite element analysis is ideal for this problem, and good results have been obtained by a number of workers. Results for two waveguide problems computed using FEM codes incorporating higher-order elements will be shown. Then, a hybrid FEM/MoM formulation, which has proven very powerful for specialized applications, will be introduced, and an application to radiation exposure assessment near a base-station antenna will be presented. Following this, time domain finite element analysis is briefly discussed.

We conclude the chapter with a discussion on two issues which impact on efficiency. Firstly, sparse matrix storage schemes are briefly outlined, and secondly, error estimation and the use of mesh adaptation based on this is discussed.

The coverage in this chapter is at a higher level than in much of the rest of this book. Generally, the topics discussed are too complex to permit a simple implementation, and the intention of this chapter is rather to sensitize the reader to current topics of interest in the field. Nonetheless, with the exception of time domain FEM, aspects of all the topics discussed are either already incorporated in commercial codes, or can be expected to be available shortly.

10.1 Higher-order elements

Although extending "edge" elements to higher orders became a topic of interest as soon as the CT/LN elements achieved widespread acceptance, it remains a topic of active research at present, a decade or more later. Development of such elements raises a number of issues, including: hierarchal versus interpolatory behavior; methods for the construction of the element shape functions; the interpretation of the degrees of freedom; the construction of prototype elemental matrices (analytical versus quadrature); and the efficient iterative solution of the poorly conditioned linear algebra systems which unfortunately often result. Various names are in use: the two-part field description as used in the preceding chapter (e.g. linear tangential/quadratic normal, LT/QN) is particularly insightful and is used here. However, before introducing higher-order elements, it is worthwhile briefly discussing the question of completeness and vector elements.

10.1.1 Complete versus mixed-order elements

A family of polynomials is *complete* to order N if a linear combination of its members can exactly express any polynomial of degree not exceeding N, but no higher [1, p. 272]. For a complete first-order approximation of a function in x and y, three terms are needed; one constant and two terms linear in x and y respectively. Clearly, for a first-order complete expansion of a two-dimensional vector *field*, each component will require three terms, hence six degrees of freedom will be required. For a tetrahedral element, approximating a three-dimensional field, twelve are needed (there is an additional linear term in z for each component, and of course, three components). By comparison, the Whitney triangular element has three degrees of freedom, and the tetrahedral element six; as we have seen in Section 9.6, this results in certain field components being approximated by a constant, and clearly these elements are of mixed order.

So many of the early papers on Whitney elements emphasized the mixed-order nature of the element that it is not always appreciated that being of mixed order is not an essential property of vector elements per se. Complete sets of vector elements have also been described [2], *with degrees of freedom proportional to tangential field components*, as for mixed-order elements. This permits enforcement of only tangential field continuity, as for mixed-order elements, with normal (dis)continuity following as a natural boundary condition, as discussed in Chapter 9. (It is easy to produce complete *scalar* or nodal elements, but then of course we are back with the inconvenient problem of having degrees of freedom representing a Cartesian component at a point, rather than a tangential field component, which was one of the major motivations for the development of vector elements, as we saw in in Chapter 9.) For wave *eigenvalue* problems, such complete sets

Table 10.1 *Webb's hierarchal elements (to second order complete) [3]*

CT/LN ($6 \times 1 = 6$ edge-based degrees of freedom)		
Edge-based	1 per edge	$\zeta_i \nabla \zeta_j - \zeta_j \nabla \zeta_i$
Additional LT/LN functions ($6 \times 1 = 6$ extra edge-based degrees of freedom)		
Edge-based	1 per edge	$\nabla \left(\zeta_i \zeta_j \right)$
Additional LT/QN functions ($4 \times 2 = 8$ extra face-based degrees of freedom)		
Face-based	2 per face	$\zeta_j \zeta_k \nabla \zeta_i + \zeta_i \zeta_k \nabla \zeta_j - 2\zeta_i \zeta_j \nabla \zeta_k$ for $\{i;\ j;\ k;\ \} = \{1;\ 2;\ 3\}$ and $\{2;\ 3;\ 1\}$
Additional QT/QN functions (6×1 edge-based $+ 4 \times 1$ face-based $= 10$ extra degrees of freedom)		
Edge-based	1 per edge	$\nabla \left(\zeta_i \zeta_j [\zeta_i - \zeta_j] \right)$
Face-based	1 per face	$\nabla \left(\zeta_i \zeta_j \zeta_k \right)$

After [4], ©2003 IEEE, reprinted with permission.

of vector elements produce "wasted" degrees of freedom, as we have already discussed. In essence, Nedelec's constraints provide mixed-order elements that model the curl-space as efficiently as possible, for a given number of degrees of freedom. However, not all problems, in particular deterministic ones, share these characteristics. Recent work by Webb [3] and the present author [4] has indicated that some vector electromagnetic problems are more efficiently analyzed using complete-order vector elements, typically when the solution is dominated by electric fields strongly "gradient" in nature.

10.1.2 Hierarchal vector basis functions

There are presently two competing approaches to higher-order vector elements. One approach is interpolatory; in this case, a degree of freedom is typically associated with a tangential field at a specific point. The other approach is hierarchal, in which case a specific higher-order set contains all the lower-order basis functions.[1] For mesh refinement/enrichment purposes, hierarchal elements are very useful, and here we consider only the use of such elements, in particular those presented in [3]. (For a comprehensive discussion of interpolatory elements, see [5]. These elements can be used for h-adaptation, but are inconvenient at least for p-adaptation. We will discuss these topics in Section 10.9.) These elemental basis functions are summarized in Table 10.1, along with the number of degrees of freedom per tetrahedron and their respective associations with edges or faces. Webb presented the information slightly differently in his paper [3, Tables III, IV

[1] Nodal elements can be both interpolatory and hierarchal; there does not appear to be a proof prohibiting a set of vector elements from having both properties. However, no such vector elements have yet been proposed.

Table 10.2 *Comparison of various hierarchal element schemes (to LT/QN)*

	CT/LN, all	
Edge-based	1 per edge	$\zeta_i \nabla \zeta_j - \zeta_j \nabla \zeta_i$
	LT/QN, Savage [10]	
Edge-based	1 per edge	$\nabla(\zeta_i \zeta_j)$
Face-based	2 per face (and $\{j; i; k\}$ but not $\{k; i; j\}$)	$\zeta_i(\zeta_j \nabla \zeta_k - \zeta_k \nabla \zeta_j)$
	LT/QN, Webb and Forghani [7]	
Edge-based	1 per edge	$\nabla(\zeta_i \zeta_j)$
Face-based	2 per face (and $\{j; k; i\}$ but not $\{i; j; k\}$)	$\zeta_i \zeta_k \nabla \zeta_j$
	LT/QN, Andersen and Volakis [11]	
Edge-based	1 per edge	$(\zeta_i - \zeta_j) \times (\zeta_i \nabla \zeta_j - \zeta_j \nabla \zeta_i)$
Face-based	2 per face (as for Savage's elements)	$\zeta_i(\zeta_j \nabla \zeta_k - \zeta_k \nabla \zeta_j)$
	LT/QN, Webb [3]	
Edge-based	1 per edge	$\nabla(\zeta_i \zeta_j)$
Face-based	2 per face	$\zeta_j \zeta_k \nabla \zeta_i + \zeta_i \zeta_k \nabla \zeta_j - 2\zeta_i \zeta_j \nabla \zeta_k$ for $\{i; j; k; \} = \{1; 2; 3\}$ and $\{2; 3; 1\}$

After [4], ©2003 IEEE, reprinted with permission.

and V]; here, the additional gradient-space functions required for the LT/LN and QT/QN elements have been explicitly written as gradients of products of simplex coordinates to highlight this functional dependence. Note that only the *additional* basis functions required are tabulated, to avoid repetition; i.e. the full second-order QT/QN set of basis functions will include *all* thirty listed.

Webb's approach is elegant in that one progressively enriches the curl space, and then the gradient space.[2] (Earlier proposals did not follow this approach.) For example, moving from CT/LN to LT/LN, one adds elements of the form $\nabla\left(\zeta_i \zeta_j\right)$, one per edge, which is clearly in the gradient space. (The curl of this is identically zero.) This then gives a complete first-order approximation function. Moving from LT/LN to LT/QN, an additional eight face-based degrees of freedom are added, giving twenty vector-based functions and degrees of freedom per tetrahedron. "Face-based" means that the degree of freedom is associated with the integral of the tangential field over the face.

Many other hierarchal elements have been published, in particular of LT/QN order. Some of these are summarized in Table 10.2. This table should serve as a useful summary of some of the various elements in current use. Another recent

[2] The Whitney element is actually a special case; it includes elements of both.

contribution on hierarchal elements is the work of Sun and Lee [6]; they use a slightly different approach to construct the elements, but the resulting basis functions are very similar, although not identical, to [3]. Most of these (including those of Savage described above) can be seen as variants of the elements originally proposed by Webb and Forghani [7]. (Indeed, not only are these variants on a theme, they are also linear transforms.) A number are summarized in Table 10.2. Note that all the face elements exclude (arbitrarily) one possible combination of $\{i; j; k\}$; this asymmetry has long been noted, and is required to avoid linearly dependent basis functions.

These elements are generally constructed by "inspection," using the properties of simplex coordinates, and the gradients thereof. Webb's recent work is the most comprehensive and theoretically motivated development along these lines to appear in the electrical engineering literature. It is worth investigating the properties of these elements a little further, since some of these are far from trivially obvious. For instance, it is not immediately apparent *why* the higher-order hierarchal elements have degrees of freedom associated with edges, faces or in some cases, with neither of these (the "volume-centered" degrees of freedom).

10.1.3 Properties of hierarchal basis functions

For this, it is useful to return to some basic properties of these elements, as originally laid down by Nedelec [8]. (It should be commented that not all vector elements which have been proposed satisfy his criteria, but those presently under discussion do.) Nedelec focussed on degrees of freedom, rather than basis functions; indeed, his original work simply states the necessary properties, rather than proposing actual basis functions. The degrees of freedom as he defined them are not unique,[3] even for the lowest order (Whitney) element, although in practice the non-uniqueness is only a matter of a constant for the lowest order case and does not impact on the code at all. However, as seen in the previous section, a variety of different basis functions have been proposed for higher-order elements.

This is rather cryptically implied in Definition 4 of Nedelec's original work [8]. For "kth" mixed-order elements, the $6k$ edge-based degrees of freedom for 3D elements ($3k$ in 2D) should be given by

$$\int_a \vec{u} \cdot \hat{t} q \, dC, \qquad \forall q \in P_{k-1} \tag{10.1}$$

$\vec{u}$ is a basis function and $\hat{t}$ is the unit vector along edge a. P_k is the linear space of polynomials of degree $\leq k$. For the Whitney element, with $k = 1$, we see that

[3] The polynomial space described is, however.

q may only be a constant. In the case of this element, with $(\zeta_i \nabla\zeta_j - \zeta_j \nabla\zeta_i)$ form, this constant is often implicitly unity, and the associated Nedelec degree of freedom (which may be viewed as located at the middle of the relevant edge, although this is not essential) is the tangential field on this edge. We commented earlier in Section 9.6 that it may be shown that the integral of the tangential component of the Whitney element along an edge is constant; we will now do this.

This proof is rather simple. Integrating the Whitney element along an edge yields two integrals. The first is of the form

$$\int_a \zeta_i (\nabla\zeta_j \cdot \hat{t})\, dC \tag{10.2}$$

and the other has i and j interchanged, and is of opposite sign. Throughout the element, $\nabla\zeta_j$ is constant, and it is perpendicular to the edge opposite node j (this was discussed in Section 9.6). Clearly, $\nabla\zeta_j \cdot \hat{t}$ is thus also a constant along any particular edge. Along the edge directed from nodes i to j, what remains is an integral of a simplex coordinate, varying linearly from 0 to 1, along the edge. The result is $\pm 1/2\ell$, with ℓ the edge length, and the sign depending on the direction of integration. Clearly, this is a constant, as is the other integral. Obviously, incorporating additional constants, such as Nedelec's q, changes only the final constant, which is irrelevant in practice. The result is as in Appendix A,

$$\vec{E}_{\tan}|_{\text{edge}_i} = \frac{E_i}{\ell_i} \tag{10.3}$$

When $\vec{E}_{\tan}|_{\text{edge}_i}$ is integrated along edge i, the result is the well-known identity that the appropriate degree of freedom is the tangential field along edge i.

Importantly, on the other two sides, one or the other simplex coordinate will be zero, and the other *entirely normal to the edge*. Thus Eq. (10.1) will yield zero for this term on the other two edges (due to the $\vec{u} \cdot \hat{t}$ term). The argument is precisely the same for tetrahedra.

Additional edge-based degrees of freedom, as required for Webb's scheme for LT/LN order elements and higher, of the form

$$\nabla(\zeta_i \zeta_j) = \zeta_i \nabla\zeta_j + \zeta_j \nabla\zeta_i$$

yield exactly the same result – they contribute additional degrees of freedom on edge $\{i; j\}$ and nothing to the other edges. (Note that a different choice of q may be required in this case, otherwise the degree of freedom is zero. A linear function is an obvious possibility, such as a suitable Legendre polynomial.)

Now, the face-based elements. Nedelec's original definition of the $4k(k-1)$ face-based degrees of freedom for higher-order elements of maximum (but not

complete) order k was

$$\iint_f \vec{u} \times \hat{n} \cdot \vec{q}\, dS, \qquad \forall \vec{q} \in (P_{k-2})^2 \tag{10.4}$$

Here, $\hat{n}$ is the unit vector normal to edge f, and the polynomial q is now two dimensional (for $k = 2$, this must be a constant). Let us now see why these additional degrees of freedom, which enrich the curl space for the LT/QN element, are associated *only* with faces. We will consider vector elements of the form $\zeta_i(\zeta_j \nabla\zeta_k - \zeta_k \nabla\zeta_j)$; the Webb LT/QN enrichment in Table 10.1 is a linear combination of two such forms, so the argument includes these. On face i, j, k, one of the simplex coordinates will always be zero on each edge; for example, ζ_i is zero on edge $\{j, k\}$, so these do not contribute to the edge-based degrees of freedom. (This extends to faces, e.g. ζ_i is zero everywhere on face $\{j, k, l\}$. Hence this basis function will have no tangential projection on any other face.)

Over face $\{i, j, k\}$, the degrees of freedom are thus

$$\iint_f \zeta_i \zeta_j (\nabla\zeta_k \times \hat{n}) \cdot \vec{q}\, dS - \iint_f \zeta_i \zeta_k (\nabla\zeta_j \times \hat{n}) \cdot \vec{q}\, dS \tag{10.5}$$

The $(\nabla\zeta_k \times \hat{n})$ and $(\nabla\zeta_j \times \hat{n})$ terms are constant over this face, as is q, and what remains are two standard integrals in simplex form, proportional to the triangle area and thus constant.

The higher-order elements (quadratic tangential/cubic normal, QT/CuN, etc.) involve additional "volume-centered" degrees of freedom. These each involve products of *all* the simplex coordinates, so are clearly zero on *all* faces and edges. For these basis functions, the associated degree of freedom as defined by Nedelec is a weighted integral over the volume.

10.1.4 Practical impact of higher-order basis functions in an FEM code

The discussion in the preceding section may appear highly theoretical, so it is worthwhile summarizing the practical impact hereof. Finite element codes do not usually actually compute the degrees of freedom as defined by Eqs. (10.1) and (10.4), since this usually serves no particular purpose. The "degrees of freedom" for which an FEM code solves are usually simply the unknowns associated with each basis function; as we have seen, for the Webb elements (and most other properly defined vector elements) these degrees of freedom can be correctly associated with edges, faces or the volume, and for the first two, the degrees of freedom are tangential field projections onto the edge or face, as required by Nedelec's original work. To enforce field continuity correctly, a degree of freedom associated with an

edge or face must simply be shared between all connected elements; we discussed this in the context of edges in the preceding chapter. Note that edges have directions; the numbering scheme used there ensured that the directions were consistent between elements, and one must do the same with faces. Volume-centered degrees of freedom have no projection on the edges or faces and hence are *not* shared by adjoining elements.

A tricky problem which was surprisingly neglected in the literature until recently was the question of how to match fields when using hierarchal elements to an actual specified field, as required by a Dirichlet boundary condition, for instance. With interpolatory elements, this would be trivial, since each degree of freedom would, by design, correspond to a tangential field component at a point on each element. With hierarchal elements however, this is only uniquely defined for Whitney elements. Webb [9] has very elegantly addressed this issue for higher-order hierarchal elements using the elements in [3]. As we have seen, starting with the conventional Whitney elements, Webb's elements enrich alternately the gradient and curl spaces. Webb exploits this in [9] to match alternately the tangential components of the electric field, and then the normal component of the electric field (the curl space). Since any such matching using hierarchal elements is approximate, he uses a projective approach to improve the accuracy of the matching.

10.2 The FEM from the variational boundary value problem viewpoint

It is useful at this stage to introduce some further ideas from functional analysis, extending the introduction in Section 4.5. This approach is strongly influenced by the methods used in applied mechanics, and is based on a development presented by Botha [12]. It will be especially useful when error estimation methods are discussed later in this chapter.

Firstly, we define a *bilinear form*. If X and Y are vector spaces, a bilinear form $\mathcal{B} : X \times Y \to \mathbb{C}$ is an operator with the properties

$$\begin{aligned} \mathcal{B}(\alpha u + \beta w, v) &= \alpha\mathcal{B}(u, v) + \beta\mathcal{B}(w, v), \qquad u, w \in X,\ v \in Y \\ \mathcal{B}(u, \alpha v + \beta w) &= \alpha\mathcal{B}(u, v) + \beta\mathcal{B}(w, v), \qquad u \in X,\ v, w \in Y \end{aligned} \tag{10.6}$$

with α and β complex numbers. In short, the operator $\mathcal{B}$ is linear in each of its "slots."

In the context of the high-frequency functional, the boundary value problem to be solved on domain Ω, in terms of the electric field, is the vector wave equation with appropriate boundary conditions, either Dirichlet on Γ_D or Neumann on Γ_N,

with of course the boundary (also called *closure*) $\Gamma = \Gamma_D + \Gamma_N$:

$$\nabla \times \frac{1}{\mu_r} \nabla \times \vec{E} - k_0^2 \varepsilon_r \vec{E} = -jk_0 Z_o \vec{J} \text{ on } \Omega$$
$$\hat{n} \times \vec{E} = 0 \text{ on } \Gamma_D$$
$$\hat{n} \times \nabla \times \vec{E} = \vec{N} \text{ on } \Gamma_N \tag{10.7}$$

This is a "strong" version of the problem; a vector field $\vec{E}$ which satisfies the vector wave equation must be twice differentiable.

Note that from one of Maxwell's curl equations, the Neumann boundary condition can also be written as

$$\hat{n} \times \vec{H} = \frac{j}{\mu\omega} \vec{N} \text{ on } \Gamma_N \tag{10.8}$$

Thus, the Neumann boundary condition can be seen equivalently as a constraint on tangential $\vec{H}$.

Using a method of weighted residuals approach, with an arbitrary testing function $\vec{W}$, and otherwise proceeding in a very similar manner to that of Section 9.2.4, it may be shown that the following is the "weak" representation of the boundary value problem represented by Eq. (10.7):

$$\begin{aligned} &\iiint_V \left[\frac{1}{\mu_r} (\nabla \times \vec{E}) \cdot (\nabla \times \vec{W}) - k_0^2 \epsilon_r \vec{E} \cdot \vec{W} \right] dV \\ &- \iint_{\Gamma_D} \frac{1}{\mu_r} (\nabla \times \vec{E}) \cdot (\hat{n} \times \vec{W}) \, dS \\ &= - \iint_{\Gamma_N} \frac{1}{\mu_r} \vec{N} \cdot \vec{W} dS - jk_0 Z_0 \iiint_V \vec{J} \cdot \vec{W} dS \\ &\text{with } \hat{n} \times \vec{E} = 0 \text{ on } \Gamma_D \end{aligned} \tag{10.9}$$

A symmetry argument is used to establish that $\vec{W}$ must also satisfy the homogeneous boundary condition on Γ_D, so that the surface integral over Γ_D on the left-hand side falls away. The final form of the variational boundary value problem is

$$\mathcal{B}(\vec{E}, \vec{W}) = \mathcal{L}(\vec{W}) \qquad \forall \ \vec{W} \in W, \ \vec{E} \in W \tag{10.10}$$

The bilinear and linear forms are defined as

$$\mathcal{B}(\vec{E}, \vec{W}) = \iiint_V \left[\frac{1}{\mu_r} (\nabla \times \vec{E}) \cdot (\nabla \times \vec{W}) - k_0^2 \epsilon_r \vec{E} \cdot \vec{W} \right] dV \tag{10.11}$$

$$\mathcal{L}(\vec{W}) = - \iint_{\Gamma_N} \frac{1}{\mu_r} \vec{N} \cdot \vec{W} \, dS - jk_0 Z_0 \iiint_V \vec{J} \cdot \vec{W} \, dS \tag{10.12}$$

The space in which the solution and testing vector functions lie is defined as

$$W = \{\vec{a} \in H(\text{curl}, \Omega) | \hat{n} \times \vec{a} = 0 \text{ on } \Gamma_D\} \tag{10.13}$$

This is the space of curl-conforming vector basis functions which we have already discussed, with the additional constraint of the homogeneous Dirichlet boundary condition.

In this development, the Neumann boundary condition has been "absorbed" into the variational boundary value problem – via the first term on the right-hand side of Eq. (10.12). (It will be recalled that in Section 9.2.4, a similar result was obtained in the context of a homogeneous Neumann boundary condition.) The Dirichlet boundary condition must however be explicitly enforced via a restriction on the space W. (This sounds more complex than it is; as was seen in Section 9.7, this is implemented in practice by zeroing the prescribed edges.)

With the variational boundary value problem established, one can then proceed to demonstrate that the stationary functional representation of the problem is the following:

$$F(\vec{E}) = \frac{1}{2}\mathcal{B}(\vec{E}, \vec{E}) - \mathcal{L}(\vec{E}), \qquad \vec{E} \in W \tag{10.14}$$

This is the familiar curl-curl functional, which we used in the preceding chapter (although the linear term was zero for the eigenvalue problem). Note that this (and indeed the variational boundary value problem from which the stationary functional form is obtained) is known as a "weak" form; the differentiability requirements on the solution space have been reduced (the function $\vec{E}$ need only be once differentiable now).

10.3 A deterministic 3D application: waveguide obstacle analysis

10.3.1 Introduction

The analysis of waveguide discontinuities has been a canonical problem for analytical, approximate, and now numerical approaches since the pioneering work of Marcuvitz and colleagues during the Second World War, now some sixty years back. Using variational formulations, and quasi-static approximations of the fields, Marcuvitz *et al.* were able to analyze an extraordinary variety of problems, documented in the classic text originally published in 1951 and now fortunately available again [13]. Subsequently, mode-matching methods were introduced for the analysis of "stepped" discontinuities, – i.e. structures where the waveguide modes could be computed in a stepwise fashion, and matched at two-dimensional planes. However, for general, arbitrary discontinuities, and of course those involving non-metallic discontinuities such as dielectrics, differential equation based methods

such as the finite element method (FEM) and finite difference time domain (FDTD) method are now the methods of choice. In this section, we will first present the formulation for this, which also affords the opportunity to deal with the more general version of the curl-curl functional as discussed in Section 10.2, and then analyze a waveguide device using both a code developed by the present author, as well as a commercial FEM package.

10.3.2 The waveguide formulation

The formulation to be discussed is a straightforward extension of Jin's approach [14], published by the present author in [15]. His formulation addressed two-port, single-mode analysis, with the waveguide oriented in the $\hat{z}$-direction. Here, general waveguide orientation(s) will be considered. The formulation assumes hollow, rectangular guide at the ports (although the extension to homogeneously filled guide is straightforward). The TE_{10} mode is assumed in the following. In between the ports, in the region to be discretized using finite elements, the waveguide may contain linear, inhomogeneous, lossy, dielectric and/or magnetic material(s); and/or conductors (for instance, posts or irises); and may change orientation (e.g. *E*-plane bends) or dimension (e.g. *E*- and/or *H*-plane steps). The formulation to be used does, however, assume isotropic media. The generalization of the analysis to multiple ports, the inclusion of higher-order modes, and the extension to more general waveguide, will be outlined subsequently.

Formulation overview

The key part of the formulation is to write the electric field at port 1 (S_1) as the sum of the known incident and unknown reflected fields in terms of the (ξ, η, ζ) coordinate system local to the port, with ζ in the local direction of propagation, and set to zero at each port, as follows:

$$\begin{aligned}\vec{E}(\xi,\eta,\zeta) &= \vec{E}^{\text{inc}}(\xi,\eta,\zeta) + \vec{E}^{\text{ref}}(\xi,\eta,\zeta) \\ &= (E_0\vec{e}_{10}(\xi,\eta)e^{-jk_{\zeta_{10}}\zeta} + RE_0\vec{e}_{10}(\xi,\eta)e^{+jk_{\zeta_{10}}\zeta})|_{\zeta=0} \qquad (10.15)\end{aligned}$$

$\vec{e}_{10}(\xi, \eta)$ is the relevant waveguide eigenmode (the TE_{10} eigenmode here) and $k_{\zeta_{10}}$ is the modal propagation constant. Note that it is necessary to retain the $e^{-jk_{\zeta_{10}}\zeta}$ term, even though the field is evaluated at $\zeta = 0$, since the boundary condition to be discussed involves the derivative of the field, which must be evaluated *before* setting $\zeta = 0$.

The next key element of the formulation is to convert Eq. (10.15) to a boundary condition of the third type involving both the field and its normal derivative. Such boundary conditions can be incorporated in the bilinear functional, as will be seen

shortly. The detail is given in [14, Section 8.5], briefly, the result is:

$$\hat{n} \times (\nabla \times \vec{E}) + \gamma \hat{n} \times (\hat{n} \times \vec{E}) = \vec{U}^{\text{inc}} \tag{10.16}$$

with

$$\gamma = jk_{\zeta_{10}}, \quad \vec{U}^{\text{inc}} = -2jk_{\zeta_{10}}\vec{E}^{\text{inc}} \tag{10.17}$$

It should be noted that, in obtaining Eq. (10.16), the transverse only nature of the TE field is exploited. TM modes contain axial $\vec{E}$ field components, and the boundary condition cannot thus be written for an $\vec{E}$ field solver. TM mode analysis could be undertaken by using an $\vec{H}$ field solver.

The same is repeated at port 2, but at that port, there is only an unknown transmitted field:

$$\begin{aligned} \vec{E}(\xi, \eta, \zeta) &= \vec{E}^{\text{trans}}(\xi, \eta, \zeta) \\ &= T E_0 \vec{e}_{10}(\xi, \eta) e^{-jk_{\zeta_{10}}\zeta}|_{\zeta=0} \end{aligned} \tag{10.18}$$

Similar comments apply as regards the $e^{-jk_{\zeta_{10}}\zeta}$ term. The boundary condition at port 2 is

$$\hat{n} \times (\nabla \times \vec{E}) + \gamma \hat{n} \times (\hat{n} \times \vec{E}) = 0 \tag{10.19}$$

In Jin's original formulation, the phase was referenced to each port. In the present formulation, the transmission coefficient T incorporates the "insertion" phase, i.e. for a section of empty guide length ℓ, T will have phase angle $-k_{z_{10}}\ell$. This produces the same phase that would be measured using a vector network analyzer, with reference planes calibrated at the ports.

The equivalent variational functional (assuming isotropic but possibly lossy materials), subject to these boundary conditions on the ports and $\vec{E}_{\text{tan}} = 0$ on the perfectly conducting walls, is:

$$\begin{aligned} F(\vec{E}) = &\frac{1}{2}\iiint_V \left[\frac{1}{\mu_r}(\nabla \times \vec{E}) \cdot (\nabla \times \vec{E}) - k_0^2 \epsilon_r \vec{E} \cdot \vec{E}\right] dV \\ &+ \iint_{S_1} \left[\frac{\gamma}{2}(\hat{n} \times \vec{E}) \cdot (\hat{n} \times \vec{E}) + \vec{E} \cdot \vec{U}^{\text{inc}}\right] dS \\ &+ \iint_{S_2} \left[\frac{\gamma}{2}(\hat{n} \times \vec{E}) \cdot (\hat{n} \times \vec{E})\right] dS \end{aligned} \tag{10.20}$$

This can be obtained from the development in Section 10.2. In this case, in the Neumann boundary condition of Eq. (10.7), repeated here,

$$\hat{n} \times \nabla \times \vec{E} = \vec{N} \quad \text{on } \Gamma_N$$

the vector function $\vec{N}$ is $\vec{U}^{\text{inc}} - \gamma \hat{n} \times (\hat{n} \times \vec{E})$; this is substituted into the linear operator $\mathcal{L}$ of Eq. (10.12), and a vector identity is used to shift one of the $\hat{n} \times \hat{n} \times$ operators to the weighting function.

For readers interested in the details of the finite element discretization of this functional, [14] and [15] are recommended.

Computation of the S*-parameters*

The above formulation produces R and T for port 1 (S_{11} and S_{21}). It must be repeated with an incident field at port 2 to obtain S_{12} and S_{22}. Only the excitation vector changes, so this is simply a question of repeating the matrix solve. For multiple ports, the extension is obvious: T is computed at *each* port, producing one column of the S matrix. The excitation is then repeated at each port to produce other columns.

The S-parameters may be computed directly from the fields on the ports. A more accurate approach uses the orthogonality of the modes to integrate the fields computed over each port [14, Section 8.5]; as an example, for two ports the transmission coefficient is given by:

$$T = \frac{2}{abE_0} \iint_{S_2} \vec{E}(\xi, \eta, \zeta) \cdot \vec{e}_{10}(\xi, \eta)\, dS \tag{10.21}$$

As before, $\vec{e}_{10}(\xi, \eta)$ is the relevant waveguide eigenmode; a and b are the waveguide dimensions.

The waveguide formulation: another perspective

The formulation can be viewed as a finite element/boundary integral (FE/BI) formulation, using the waveguide Green function for "exact" mesh termination. (For radiation or scattering problems, FE/BI formulations use the free-space, or sometimes the half-space, Green function, e.g. [14, Section 10.4]; this is discussed later in this chapter.) The current dominant-mode-only analysis uses only the first in the infinite series of modes comprising the waveguide Green function. It is accurate provided that the ports are sufficiently far removed from the discontinuities (assuming, of course, that only the dominant mode is above cut-off). Higher-order modes are easily included in the formulation; this does require re-computing both the left-hand side matrix and right-hand side vector, since the former has one term dependent on the propagation constant, and the latter is obviously dependent on the incident mode shape. The formulation presently assumes hollow waveguide at the ports, i.e. only TE (and TM modes, if an $\vec{H}$ field solver is also implemented) are included. More exotic modes, or numerically determined ones, could also be incorporated into the formulation.

10.4 Application to two waveguide discontinuity problems

With the formulation in hand, we will now proceed to analyze two waveguide discontinuity problems. The first demonstrates multi-port analysis; the second demonstrates the use of complete vector basis functions. The latter is based on [4].

10.4.1 Application to a Magic-T

Introduction

The "Magic-T" (see Fig. 10.1) is a 180° hybrid. Such devices are four-port structures, with the following interesting properties. A signal applied to port 1 is evenly split into two in-phase components at ports 2 and 3, and port 4 is isolated. Conversely, a signal applied to port 4 is evenly split, but with 180° phase difference, between ports 2 and 3, and port 1 is isolated. It can also be operated as a combiner, in which case when input signals are applied to ports 2 and 3, the sum appears at port 1 and the difference at port 4. Ideally, the S-parameters of the device are [16, p. 402]:

$$[S] = \frac{-j}{\sqrt{2}} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & -1 \\ 1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \tag{10.22}$$

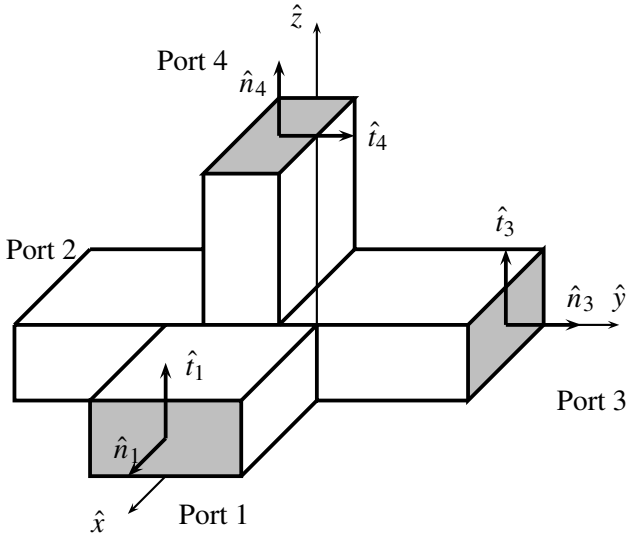


Figure 10.1 The Magic-T hybrid waveguide junction.

The 180° hybrids can be made in various ways, e.g. microstrip or stripline, the "rat race" being a very popular implementation in planar technologies. The Magic-T is a waveguide implementation of a 180° hybrid [16, p. 411].

This is an example of a structure where the approximate analytical techniques of Marcuvitz *et al.* are unable to provide useful data and a full-wave solution becomes imperative. (A complex equivalent circuit is presented for the Magic-T [13, p. 386], but only measured data at one frequency point are provided.) The behavior of the waveguide Magic-T departs very significantly from the ideal of Eq. (10.22), as will be seen.

Setting up the problem

The setup procedure for the junction as shown in Fig. 10.1 illustrates a number of features one would expect in any RF FEM code. The specific code described in detail is FEMFEKO, which is an experimental FEM code using FEKO-like input and output files, but not available for general use at the time of writing. However, the meshing is done using a commercial FEM mesher, FEMAP. In most packages, this type of structure is straightforward to model; in FEMAP, for instance, the `Solid` modelling options are the easiest. First, one 40 mm long section of X-band (22.86 mm $\times$ 10.16 mm) guide is generated (as a solid); then the other 40 mm section (at right angles to the first) is added. The structure is then meshed, using the meshing commands within the package. Following this, the mesh is then `export`-ed as a neutral file, from which it can be used by various analysis packages. (This last step is of course unnecessary in integrated FEM packages incorporating mesher and solver.)

Boundary conditions must then be applied to the structure. "Port" boundaries are required at the four ports of the device, a port corresponding to the region where the modal boundary condition of the preceding theoretical discussion is applied. In FEMFEKO, a port requires two vectors to define it. The first defines the *outward* directed normal on each port. The second defines the relative "sense" of each port; there is an ambiguity regarding the "sense" of the ports, which this helps resolve. The problem is that for a straight section of waveguide, it is obvious that the sense of each port should be the same, either up or down, but for a bent section of guide, the sense is essentially arbitrary. For instance, the tangent vector defining the positive modal sense on port 4 could equally well be chosen as $+\hat{y}$ or $-\hat{y}$. (For the results presented, the former was chosen. On ports 1, 2 and 3, $+\hat{z}$ was chosen as the tangent vector.) Various packages deal with this issue in different ways.

This problem was also solved using a commercial package, ANSOFT's HFSS code. Constructing the finite element model is very similar to the procedure described above, but since the mesher is integrated within the package, it is

appreciably more user-friendly. Nonetheless, the requirement of correctly specifying boundaries in particular rests with the user. HFSS meshes the structure automatically, and then refines the mesh until a user-specified level of accuracy is reached (usually, a negligible change in *S*-parameters from one iterative pass to the next).

For the results to be presented, the geometrical primitive cubes which defined port 1 had a length of 40 mm (i.e. approximately 20 mm of guide from the junction), for ports 2 and 3 they were 30 mm (also approximately 20 mm of guide) and for port 4, 30 mm (again, approximately 20 mm of guide). These lengths were based on the results for other waveguide structures; the requirement is that there be sufficient length to allow evanescent modes to die out before the ports. As in our 2D eigenvalue problem, the waveguide was an X-band guide with dimensions 22.86 mm $\times$ 10.16 mm.

Results

This geometry has no simple analytical solution, as already discussed. To obtain data to compare with these results, ANSOFT's HFSS code was used to generate

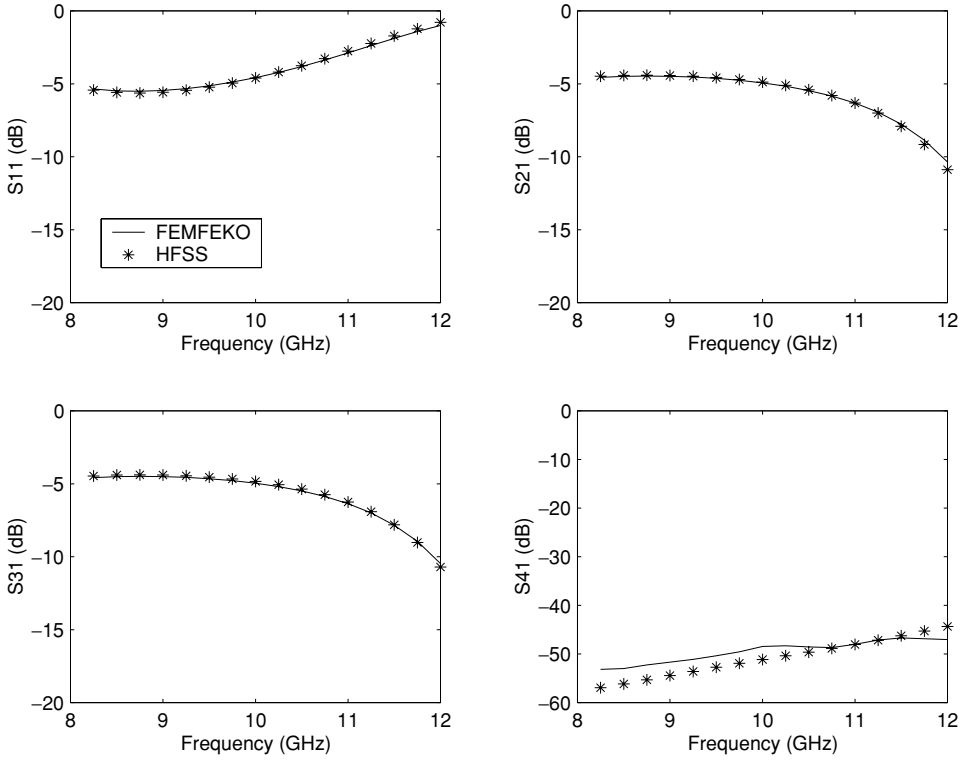


Figure 10.2 *S*-parameters of the Magic-T for port 1; magnitude.

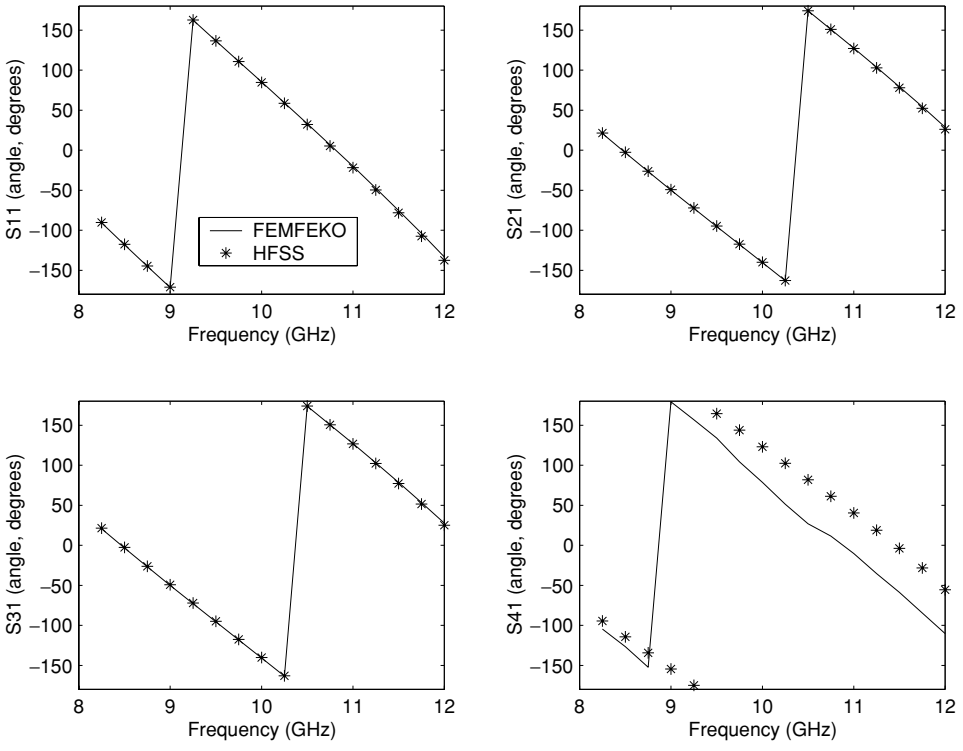


Figure 10.3 *S*-parameters of the Magic-T for port 1; phase.

another FEM solution of the problem. The HFSS model was identical in size to the FEMAP (and FEMFEKO) models, so that phase results could also be compared.

The *S*-parameter data for port 1 are presented in Figs. 10.2 and 10.3. Port 4 is indeed isolated; since S_{41} is very small, some discrepancy between the FEMFEKO and HFSS results is to be expected. Ports 2 and 3 show equal, in-phase, power splitting. Note, however, how far S_{11} departs from the theoretical ideal of no reflection at port 1. A brief consideration of the problem shows that this is not unexpected, since the waveguide fed by port 1 sees two identical waveguides in parallel at the junction (those connected to ports 2 and 3). Thus mismatch of around $1/3$ (about -10 db)[4] is to be expected. We see that the actual reflection coefficient (as computed) is worse than this.

The phase data computed by HFSS for ports 2, 3 and 4 originally had a 180° phase difference compared to the FEMFEKO results, due to the mode sense ambiguity discussed above. (HFSS has an option to define the mode sense, but this was not used.) This has been corrected in the results shown.

[4] The reflection coefficient of a system with a load equal to half the characteristic impedance.

The HFSS data used 1458 tetrahedra; the FEMFEKO result, using LT/QN elements, used 802 tetrahedra (with an average mesh length of around 6.5 mm). HFSS refines its solution using adaptive meshing techniques, so one has reasonable confidence that the results are accurate.

Conclusions

This discussion has demonstrated the application of two FEM codes to a Magic-T hybrid, a device whose complex geometry precludes approximate analytical solutions. Higher-order elements were used and very good results obtained. It was also shown that the device's performance (certainly in terms of S_{11}) departs significantly from the ideal found in textbooks, highlighting the importance of numerical simulation as a valuable tool in microwave engineering.

10.4.2 Application to a capacitive iris

A rationale for complete basis functions

In Section 10.1, complete vector basis functions were introduced, although little motivation was given for their use. The work of Webb is particularly useful in this context; [3] comprehensively discusses the motivation for both mixed-order and full-order elements. The main thrust of the argument can be summarized as follows: the variational functional which is rendered stationary by the finite element procedure consists (at its simplest) of two terms, one related to the curl of the electric field and one related to the electric field itself. (This discussion assumes the electric field is the working variable. The magnetic field can of course also be used.) The curl of the electric field is the time rate of change of the magnetic field. As already discussed in Section 10.1, the rationale behind mixed-order vector elements is to remove terms from the polynomial approximation of the electric field which do not contribute to the magnetic field. In problems where the electric and magnetic fields are of more or less equal importance, it makes sense only to use the polynomial terms which contribute to both fields, to obtain maximum accuracy for a given number of degrees of freedom.

However, there are a number of problems of interest in RF engineering where the fields are dominated by either electric or magnetic fields. In general, a sharp edge will result in a singularity in both the electric and the magnetic fields, but for certain field and discontinuity orientations, such as the capacitive iris problem to be discussed, the singularity is in the electric field alone, and hence the field is dominated by the quasi-static electric field behavior. Hence it can be expected that full-order elements should be useful for such problems.

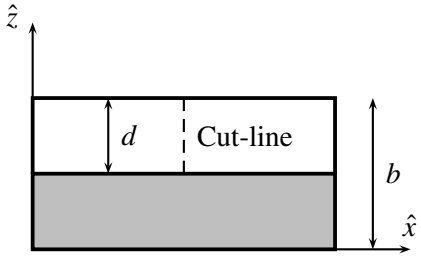


Figure 10.4 The capacitive iris. After [4], ©2003 IEEE, reprinted with permission.

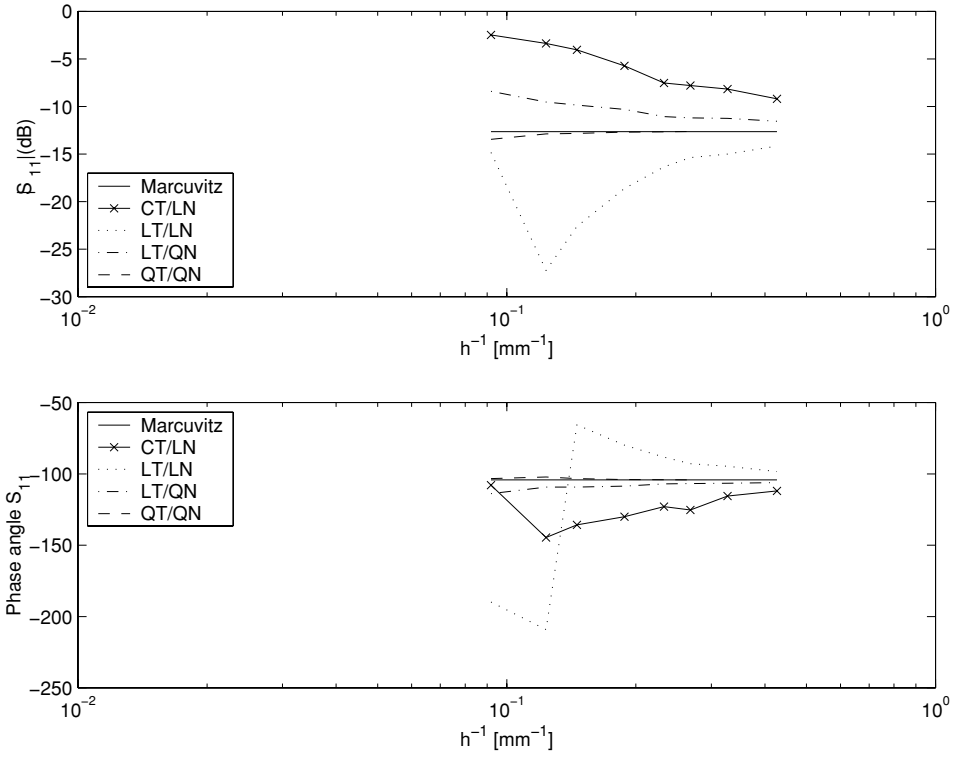


Figure 10.5 Results for a capacitive iris, compared with Marcuvitz's result, as a function of (inverse) mesh size. After [4], ©2003 IEEE, reprinted with permission.

Results

Here, a capacitive iris is considered.[5] The metallic iris, shown in Fig. 10.4, is half the height of the waveguide, and again, the analysis is performed at X-band. The results shown in Figs. 10.5 and 10.6 were computed at 8.25 GHz, towards the

[5] This example was first published as [4].

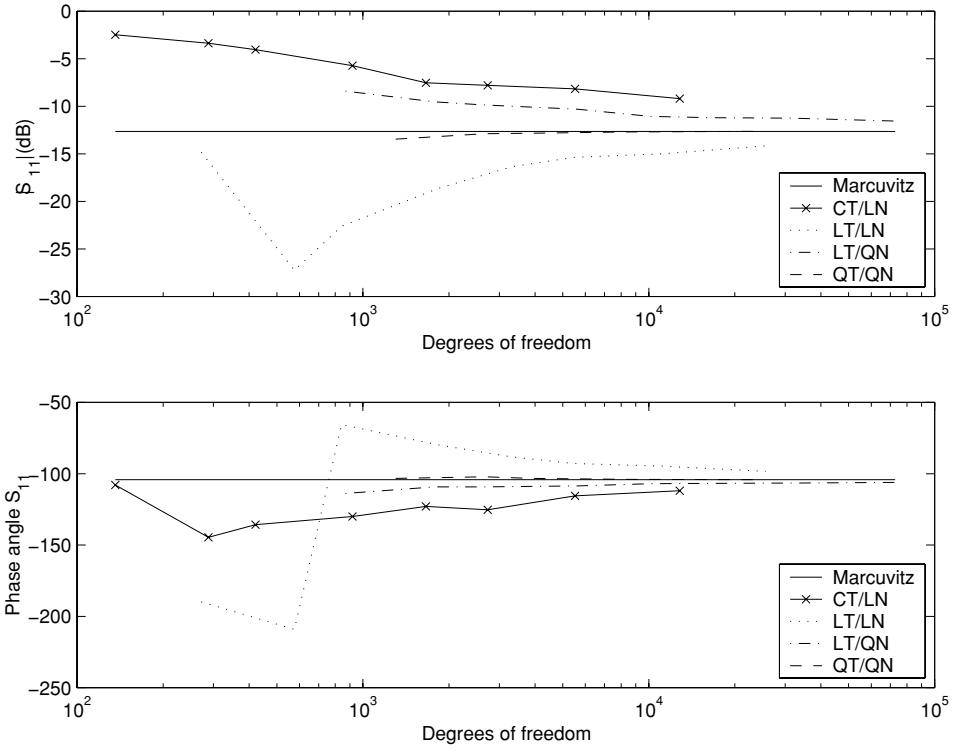


Figure 10.6 Results for a capacitive iris, compared with Marcuvitz's result, as a function of degrees of freedom. After [4], ©2003 IEEE, reprinted with permission.

bottom end of the X-band frequency range. A number of different meshes were generated for the problem; the average edge length in the mesh varied from around $h \approx \lambda_g/6$ for the coarsest mesh to $h \approx \lambda_g/25$ for the finest.

Of interest here are the excellent results for the polynomial complete QT/QN elements, which agree very well indeed with Marcuvitz's (approximate) results [13]. (Marcuvitz's models actually give equivalent circuit parameters. A discussion of how to convert these to *S*-parameters may be found in [15].) In the region $4b/\lambda_g < 1$, which is the case at this frequency in X-band waveguide, the error bound on Marcuvitz's results is given as within 1%, a result verified by this QT/QN FEM solution. It is clear that LT/QN elements converge very slowly to the correct solution for this problem. A commercial FEM code using conventional mixed-order elements also produced unconverged results for this problem, despite incorporating adaptive mesh refinement techniques.

In retrospect, it is clear that this is an especially difficult problem for general-purpose finite element solvers. Even with quite a fine mesh overall, it is likely that the mesh above the iris may only be two or three elements "thick" (this could be

improved by manual "seeding"). This of course is precisely the direction in which the field is changing most rapidly, and furthermore, the electric field is strongly dominated by the quasi-static field with singular behavior. To describe this field adequately, one would expect to need full-order elements, thus also approximating the gradient space as accurately as possible (for the given maximum element order available).

It is also of interest to note that the relative improvement of the QT/QN elements compared to the LT/QN ones appears more marked than the improvement of LT/LN over CT/LN. It is quite possible that the mesh in the vicinity of the iris (as discussed above) is limiting the performance of the linear elements – indeed, only the finest mesh in the above results had three elements "thickness" above the iris.

The above results, using *S*-parameters, concentrate on what are essentially integrated field quantities (also known as observables). It is also of interest to examine the actual field behavior in the vicinity of the iris. In Figs. 10.7 and 10.8, the

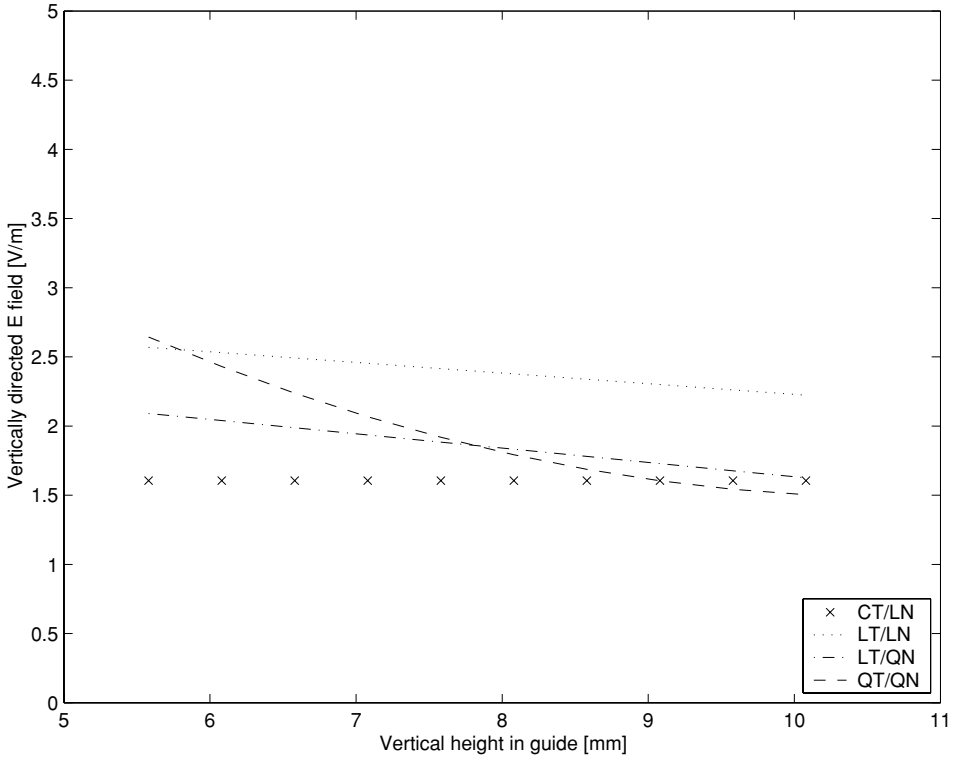


Figure 10.7 Vertically directed electric field along a line in the center of the guide, directly above the iris. Coarse mesh, $h \approx \lambda_g/6$. After [4], ©2003 IEEE, reprinted with permission.

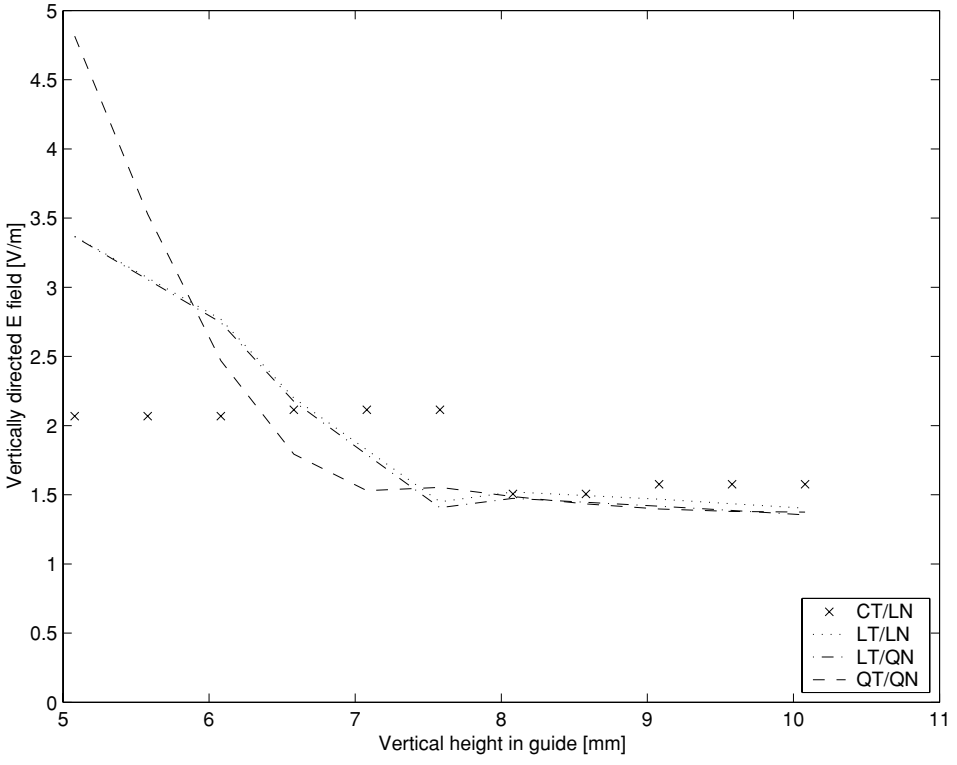


Figure 10.8 Vertically directed electric field along a line in the center of the guide, directly above the iris. Fine mesh, $h \approx \lambda_g/20$. After [4], ©2003 IEEE, reprinted with permission.

vertically directed electric field (the field component aligned with the TE_{10} mode electric field) on a vertical line directly above the iris is plotted. (The modal excitation E_0, see Eq. (10.15), at the port was 1 V/m, in this and subsequent plots.) The cut-line is located in the center of the width of the waveguide, shown in Fig. 10.4 by the dashed line. The half-height iris runs from 0 to 5.08 mm; the figures show the field from 5.08 to 10.16 mm, the roof of the guide. The superior performance of the QT/QN elements is clear in these figures; even in the fine mesh case, Fig. 10.8, the CT/LN results are poor, and evidence a considerable (and non-physical) discontinuity at around 7.5 mm. The LT/LN results are close to the LT/QN results, and the discontinuity evidenced by the CT/LN results has gone. The QT/QN results in both cases give the largest field value at the iris, indicating superior modelling of the field in this case.

Some further comments here, especially on the CT/LN results, are called for. In the coarse mesh case, the mesh generator produced only one row of elements above the iris; for the finer mesh, it produced two. In Fig. 10.7, the cut-line ran on

the boundary of an element, hence the uniform CT/LN result is to be expected. In Fig. 10.8, the cut-line went through four elements (the mesh was not symmetrical about the center-line), hence the four distinct and different values on the plot. In both these figures the CT/LN results are plotted *only* at the points where the field was computed, to avoid an incorrect linear interpolation being imposed by the plotting program.

It might be argued that this comparison is unfair, since obviously the QT/QN solution involves many more degrees of freedom than, for example, the CT/LN solution on the same mesh. This is not so in this case. The CT/LN results shown in Fig. 10.8 used 5523 degrees of freedom; the QT/QN solution in Fig. 10.7 used 1302, and the solution quality of the latter is clearly far better than that of the former, for fewer degrees of freedom. (The issue of the potentially slower convergence of the higher-order elements will not be considered, since appropriate preconditioners can rectify this problem [6].)

Discussion

This capacitive iris problem has clearly highlighted the utility of full-order elements for problems where quasi-static electric fields dominate the solution. Furthermore, electric field results for this problem have demonstrated that full-order elements can provide enhanced field modelling for a similar (or sometimes even smaller) computational effort in situations where the field itself, rather than an integrated quantity such as the transmission or reflection coefficient, is of primary concern. An interesting idea is to consider how finite element solvers might automatically identify the appropriate element type in different regions; some preliminary results show promise [12, Chapter 6; 17]. Work has also recently appeared on independently controlling the gradient and rotational polynomial orders [18].

10.5 Hybrid finite element/method of moments formulations

10.5.1 Introduction

As we have seen, finite element formulations offer powerful methods for the numerical solution of electromagnetic fields in inhomogeneous media. The major drawback for high-frequency simulation is the requirement for terminating the finite element mesh as a finite distance. Various mesh termination schemes have been proposed and implemented, including mathematical absorbing boundary conditions – requiring special treatment of "boundary" elements – and more recently, perfectly matched layers. In Chapter 3, we studied the application of both these methods within the context of the FDTD, and these methods have also been used in FEM approaches. In this section, we will instead consider an "exact" termination

scheme, which effectively uses the method of moments applied on the open boundary to terminate the FEM region, producing the FEM/MoM hybrid method. (This method is also sometimes called the boundary element/finite element method, or boundary integral/finite element method).

10.5.2 Theoretical background

Before addressing the theory of the FEM/MoM hybrid method, a connection between the Rao–Wilton–Glisson (RWG) element [19], widely used in MoM formulations, and the Whitney (CT/LN) element which we have discussed extensively here needs to be highlighted. Much earlier, in Chapter 6, it was commented that the RWG element [19] and the Whitney element are intimately connected. The relationship is the following: by simply taking the normal crossed with the Whitney element, the RWG element is obtained. It will be recalled that the Whitney element is also sometimes called "curl conforming"; the RWG element is an example of a "divergence conforming" element. (Nedelec's original work also considered such elements, although the RWG element was derived independently.) This close relationship is fortunate and not by any means serendipitous: the underlying requirements of field continuity are the reason for the close relationship. This is an important practical point, because it implies that edge-element FEM codes, with volumetric fields as unknowns, and RWG-based MoM codes, with surface currents as unknowns,[6] can at least potentially conform on a boundary.

With this background, we can now consider the FEM/MoM formulation. The following is based on the presentation in [1, Chapter 9]. Within a region Ω, with closure (bounding surface) S, and free space in the exterior region, a finite element discretization of Maxwell's equations, via the stationary functional as in Section 10.2, results in the following matrix equation:

$$[A]^E\{e\} + [B]^E\{h\}_S = \{c\}^E \tag{10.23}$$

In this equation, the superscript E indicates that the $\vec{E}$ field has been chosen as the main working variable. Matrix $[A]$ is the usual FEM matrix obtained from the bilinear functional applied throughout the volume; vector $\{e\}$ is the vector of unknown coefficients of the electric field in the volume; matrix $[B]$ represents the Neumann boundary condition applied on the surface;[7] and vector $\{h\}$ is the vector of unknown coefficients of the magnetic field on the closure. Finally, vector {c} accounts for current sources internal to the volume. Specifically, the elements of

[6] Recall that an equivalent surface current is obtained from the normal crossed with the appropriate tangential field component.

[7] Recall Eq. (10.8) and the connection with the tangential magnetic fields.

each are given by

$$A_{ij}^E = \int_\Omega \{\mu_r^{-1}(\nabla \times \vec{N}_i) \cdot (\nabla \times \vec{N}_j) - k^2 \epsilon_r \vec{N}_i \cdot \vec{N}_j\}\, d\Omega, \quad \forall\, i \text{ and } j = 1, \ldots, N \tag{10.24}$$

$$B_{ij}^E = jk\eta \oint_S \vec{N}_i \cdot (\vec{N}_j \times \hat{n})\, dS, \quad \forall\, i = 1, \ldots, N, \quad j = 1, \ldots, N_S \tag{10.25}$$

$$c^E = -\int_\Omega \vec{N}_i \cdot \{jk\eta \mathbf{J}^{\text{int}} + \nabla \times (\mu_r^{-1} \vec{K}^{\text{int}})\}\, d\Omega, \quad \forall\, i = 1, \ldots, N \tag{10.26}$$

In the above expressions, $\vec{N}_i$ and $\vec{N}_j$ are the element shape functions. The elements of $[A]$ are immediately recognized as the $[S]$ and $[T]$ matrix elements discussed in Section 9.7 (albeit now three- rather than two-dimensional), for which closed form expressions are available. $\vec{J}^{\text{int}}$ and $\vec{K}^{\text{int}}$ represent sources internal to Ω.

The problem is clear: there are $N + N_S$ degrees of freedom (N unknowns in $\{e\}$ and a further N_S unknowns in $\{h\}_S$, the latter is the $\vec{H}$ field on the boundary surface S). An additional constraint is required to connect the surface magnetic fields with the volumetric electric fields (which also of course exist on the closing surface). In the waveguide formulation, knowledge of the modal structure of the field was sufficient, but now a further matrix equation must be derived in terms of the surface fields.

Deriving essentially the EFIE and MFIE, one can obtain the following, suitable for an MoM representation on the boundary S:[8]

$$\vec{E}(\vec{r}) = \vec{E}^{\text{inc}}(\vec{r}) + \oint_S \Big(\nabla \times \bar{\bar{G}}(\vec{r}, \vec{r}') \cdot \{\hat{n}' \times E_S(\vec{r}')\} - jk\eta \bar{\bar{G}}(\vec{r}, \vec{r}') \cdot \{\hat{n}' \times H_S(\vec{r}')\} \Big)\, dS' \tag{10.27}$$

and

$$\vec{H}(\vec{r}) = \vec{H}^{\text{inc}}(\vec{r}) + \oint_S \Big(\nabla \times \bar{\bar{G}}(\vec{r}, \vec{r}') \cdot \{\hat{n}' \times H_S(\vec{r}')\} + \frac{jk}{\eta} \bar{\bar{G}}(\vec{r}, \vec{r}') \cdot \{\hat{n}' \times E_S(\vec{r}')\} \Big)\, dS' \tag{10.28}$$

Note that the S subscript refers to quantities on surface S, *not* the scattered field. $\bar{\bar{G}}$ is the dyadic free-space Green function, and $\hat{n}'$ is the outward directed normal.

[8] Also known as a *Huygen's* integral representation.

Writing these in a more compact notation, one obtains

$$-\vec{E} + L_{e1}^{S}(\vec{E}_s \times \hat{n}') + L_{e2}^{S}(\vec{H}_s \times \hat{n}') + \vec{E}^{\text{inc}}(\vec{r}) = 0 \tag{10.29}$$

Using a Galerkin procedure, this may be discretized as

$$[B]^{M}\{e\}_S + [P]^{E}\{e\}_S + [Q]^{E}\{h\}_S + \{y\}^{E} = 0 \tag{10.30}$$

The matrix $[B]^M$ in the above is of the same form as $[B]^E$ in Eq. (10.25): the only difference is that the constant term is $-jk/\eta$ instead of $jk\eta$ [1, pp. 408–409]. The other matrices are given by

$$P_{ij}^{E} = j\frac{k}{\eta}\oint\!\!\!\!\!\!\int_S \vec{N}_i \cdot \left\{ L_{e1}^{S}(\vec{N}_j \times \hat{n}) \times \hat{n} \right\} dS \tag{10.31}$$

$$Q_{ij}^{E} = j\frac{k}{\eta}\oint\!\!\!\!\!\!\int_S \vec{N}_i \cdot \left\{ L_{e2}^{S}(\vec{N}_j \times \hat{n}) \times \hat{n} \right\} dS \tag{10.32}$$

$$y_i^{E} = j\frac{k}{\eta}\oint\!\!\!\!\!\!\int_S \vec{N}_i \cdot (\vec{E}^{\text{inc}} \times \hat{n})\, dS \tag{10.33}$$

The matrix size of $[B]^M$ is $N \times N_S$, but for the boundary element terms in Eq. (10.30), only the relevant $N_S \times N_S$ submatrix is retained, so that the above matrix equation (10.30) is of dimension N_S. Similarly, Eq. (10.28) can be discretized to yield

$$[B]^{E}\{h\}_S + [P]^{M}\{h\}_S + [Q]^{M}\{e\}_S + \{y\}^{M} = 0 \tag{10.34}$$

Either Eq. (10.30) or (10.34) is sufficient to eliminate $\{h\}_S$ in terms of $\{e\}_S$, which is then substituted into Eq. (10.23). (Note that $\{e\}_S \subset \{e\}$, since these are just the components of electric field on the surface.)

The $[P]$ and $[Q]$ matrices are not straightforward to compute, since they involve integrals of Green's functions, containing integrable singularities, acting on the basis functions; see [1, p. 413]. (As we saw in Chapter 6, this is standard in MoM formulations involving a rigorous surface current treatment.) The case of a cavity in a conducting half-space has been worked further by Jin [14, Chapter 10]; his results are also summarized in [1, Chapter 9]. For more general problems, see [14, Chapter 10; 20, Chapter 11; 21, Chapter 7].

A computational problem which emerges is that the resulting system of linear equations is overwhelmingly sparse, but contains a dense submatrix repre-

senting the MoM (BEM) interactions. The overall matrix is also not, in general, symmetric.

10.6 An application of the FEM/MoM hybrid – GSM base stations

10.6.1 Applications of FEM/MoM hybrid formulations

The hybrid FEM/MoM formulation outlined above is applicable to many problems. In general antenna analysis, the FEM is not the method of choice for wire antennas, where the standard MoM formulation provides a straightforward and robust solution. However, when such antennas are radiating in the presence of electromagnetically penetrable bodies, the FEM/MoM hybrid comes into its own. Modelling the interaction of operators and personal communications systems, in particular cellular phones, has emerged as an important field of application of this formulation, and the example presented here is a variant on this theme. However, there are a number of other applications, which will now be outlined.

Cavity-backed antennas were one of the first applications of the FEM/MoM (BEM) hybrid formulations, see [14], and they continue to attract interest [21]. Although the original formulation assumed that the cavity was recessed into an infinite ground plane, recently work has extended this to cavities on elliptical shapes, permitting analysis of conformal airborne antennas. Microstrip antennas have also been efficiently analyzed using this approach; since the substrate, which is discretized with the FEM, need not be uniform in this approach, some interesting work has been done on the use of perforated substrates (a type of electromagnetic band-gap material) to reduce mutual coupling [22]. An important class of cavity-backed antenna is the spiral, both Archimedes and logarithmic. Again, stratified media MoM codes assume infinite planar media, whereas an FEM/BEM formulation need not.

General FEM/MoM hybrids also permit the analysis of microstrip antennas, removing the assumptions of infinite substrate and permitting the effect of edge diffraction to be studied. However, this is computationally quite expensive.

The use of CEM tools in what are often EMC problems can be problematic, due to the great complexity of the systems. Work by Hubing's group has proposed the use of the FEM for regions of geometric and material complexity, combined with a MoM treatment of the interconnects [23]. Work has also been done on the coupling of energy through deep slots using FEM/MoM hybrids.

Inhomogeneous objects buried in stratified media are another interesting application; perhaps the most obvious candidates here are landmines and other unexploded ordnance. The formulation required becomes extremely complex, since the "exterior" Green functions involve the Sommerfeld potentials. Eibert and Hansen present the necessary formulation in [24].

10.6.2 Human exposure assessment near GSM base stations

The widespread adoption of personal communication devices, in particular mobile (cellular) telephones, during the 1990s and the continuing growth in the present decade presented significant challenges for CEM analysis. When first introduced, there were widespread concerns over safety issues associated with the widespread and prolonged use of mobile handsets, perhaps triggered by the term "radiation."[9] After much research, it would appear that these concerns were fortunately unfounded, due primarily to the low power levels of the handsets. However, a case where there are indeed valid concerns for health issues is that of *base stations*, due to the much higher power levels encountered there (60 W is typical) and the requirement for maintenance workers to operate close to the antennas.

An aside – mobile telephony

Mobile telephony has been one of the most extraordinary technical success stories over the last decade. In many countries, in particular outside the First World, the number of mobile telephones now exceeds the number of fixed lines, and the Group Special Mobile (GSM) standard, originally operating at 900 MHz and now also 1800 MHz, has proven wildly popular everywhere apart from the USA. Indeed, at the start of 2004, figures from the International Telecommunication Union indicate that the number of mobile subscribers *worldwide* – 1.14 billion – has just overtaken the number of fixed-line subscribers, at 1.1 billion. When one considers that the current fixed-line infrastructure has been under development for the better part of a century, compared to that of less than one decade for cellular phones, this is an extraordinary and largely unheralded technical achievement, compared to the Internet, for instance. It has had a major impact on the lives of many people in less wealthy countries, who would otherwise no doubt still be waiting for a fixed line, frequently provided by parastatals with very limited capabilities.

The FDTD, FEM with ABC, FEM/MoM and also volume equivalence principle MoM formulations have all been used successfully for the analysis of human exposure assessment of radiation from handsets. However, for base stations, one has the problem both of complex wire antennas, typically mounted on a mast, and considerable distance between the human phantom and the antenna. Figure 10.9 shows an

[9] It must say something of human nature that a number of users who express such concerns are prepared to operate their mobile phones while driving, a well-known and much documented hazard, and illegal in many countries for precisely this reason!

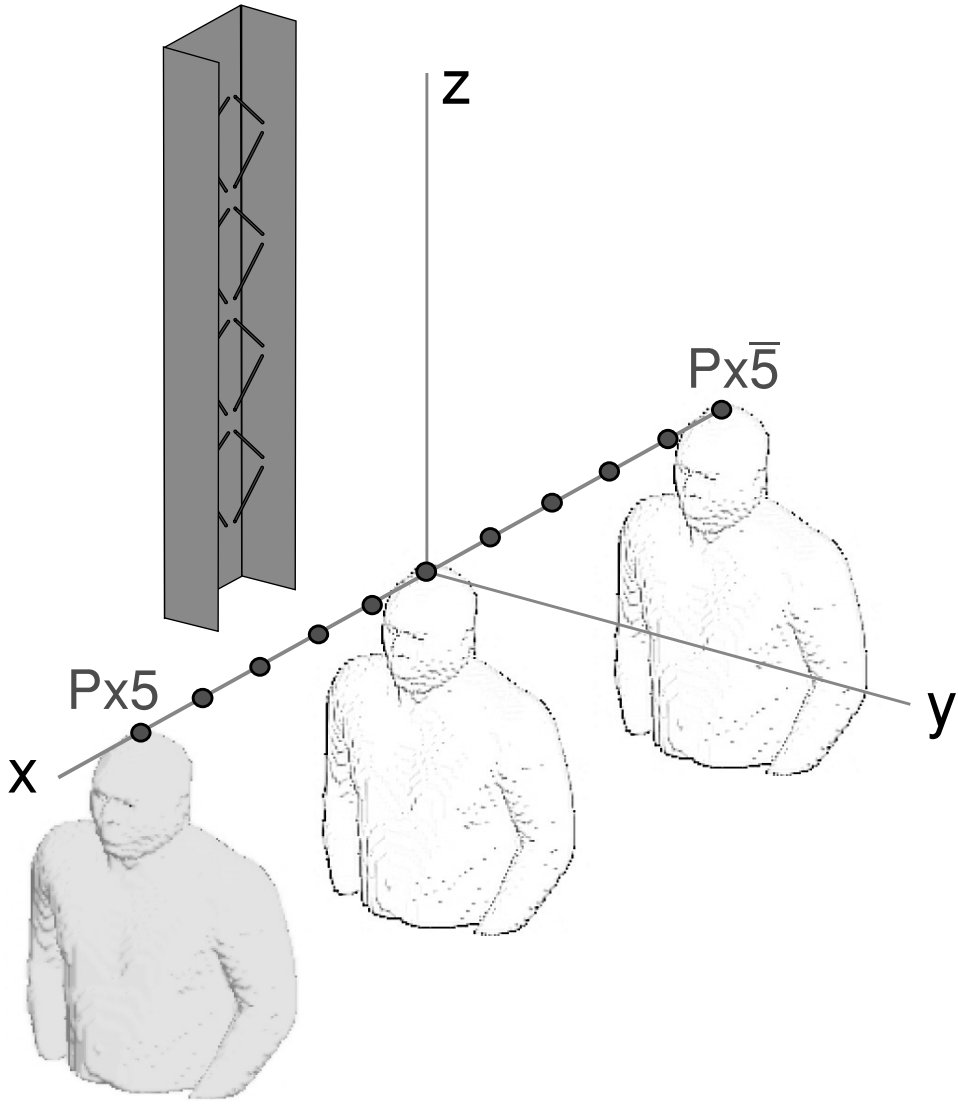


Figure 10.9 Near-field base-station exposure setup. After [25], ©2003 IEEE, reprinted with permission.

example of a typical setup. (Handsets are usually analyzed in very close proximity to the head, which has been the major health concern.) Although this was not discussed in our theoretical development, a very powerful feature of the FEM/MoM formulation is that the exterior region *may also contain scatterers/radiators*, it need not be purely free space, as shown in Fig. 10.10. These scatterers and radiators are treated with the MoM in a self-consistent manner.

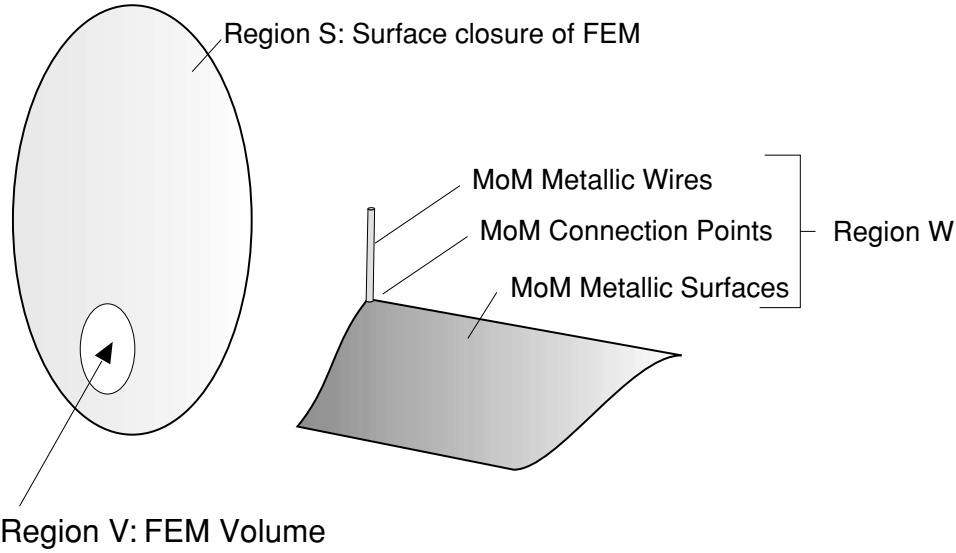


Figure 10.10 Hybrid FEM/MoM problem setup. After [25], ©2003 IEEE, reprinted with permission.

Meyer has implemented an outward-looking FEM/MoM hybrid (using a number of FEKO routines, as well as elements of the FEM code discussed in Section 10.4), and in [25], results are shown for base-station exposure assessment in terms of ICNIRP[10] guidelines. In that paper, results are also shown for careful validation of some smaller problems, using both an FDTD code and FEKO; readers are referred to the paper as a good example of this process for complex problems. Here, results for only the FEM/MoM hybrid will be shown. Of particular interest are exposure results for particular organs, shown in Figs. 10.11 and 10.12. As discussed in [25], this particular problem could not be analyzed in any way other than the FEM/MoM, since it was electrically too large for both the FDTD and the MoM volume equivalence principle.

10.7 The time domain FEM

Time domain finite elements are widely used in other fields of engineering, but have not seen especially widespread use in CEM. This probably reflects both the technological driving forces behind the development of CEM, which until the 1980s emphasized the development of frequency domain formulations (since most RF communication and radar systems were inherently narrowband), as well as the competing algorithm in the time domain, the FDTD, which is so firmly

[10] International Commission on Non-Ionizing Radiation Protection.

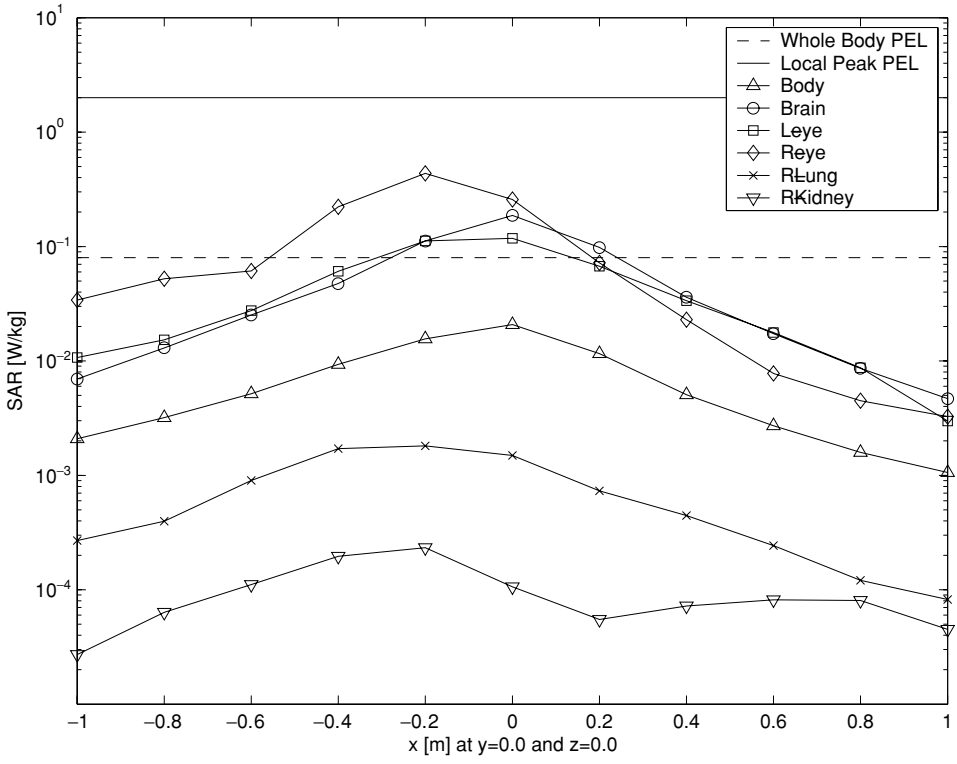


Figure 10.11 Average specific absorption rate (SAR) in different body organs compared with whole-body (0.08 W/kg) and spatial-peak (2 W/kg) ICNIRP basic restriction, x-direction (transverse across antenna), for the base-station/half-body problem shown in Fig. 10.9. $P_{\text{rad}} = 60$ W. Front of base-station antenna at $y = -0.428$ m and to-center of phantom head at $y = 0$ m. Adapted from [25], ©2003 IEEE, reprinted with permission.

established in CEM and has produced so many excellent results that it is difficult to "sell" another time domain formulation. Nonetheless, the finite element time domain (FETD) method has seen a considerable amount of work and development in CEM over the last decade. In particular for devices with fine geometrical detail, it can be expected to emerge as a competitor for specialized applications. Perhaps the most interesting use is as a hybrid form with the FDTD, which exploits the superior geometrical modelling ability of finite elements with the robustness and speed of the FDTD method; no commercial implementation is presently available, nor is likely to be for some time, but recent research has produced good results [26, 27]. In a book which is otherwise devoted to methods already implemented in widespread public domain and commercial codes, coverage of this method may seem slightly anomalous, but at least one interesting point which emerges is a more general view of the FDTD method, which is actually a special case of the general

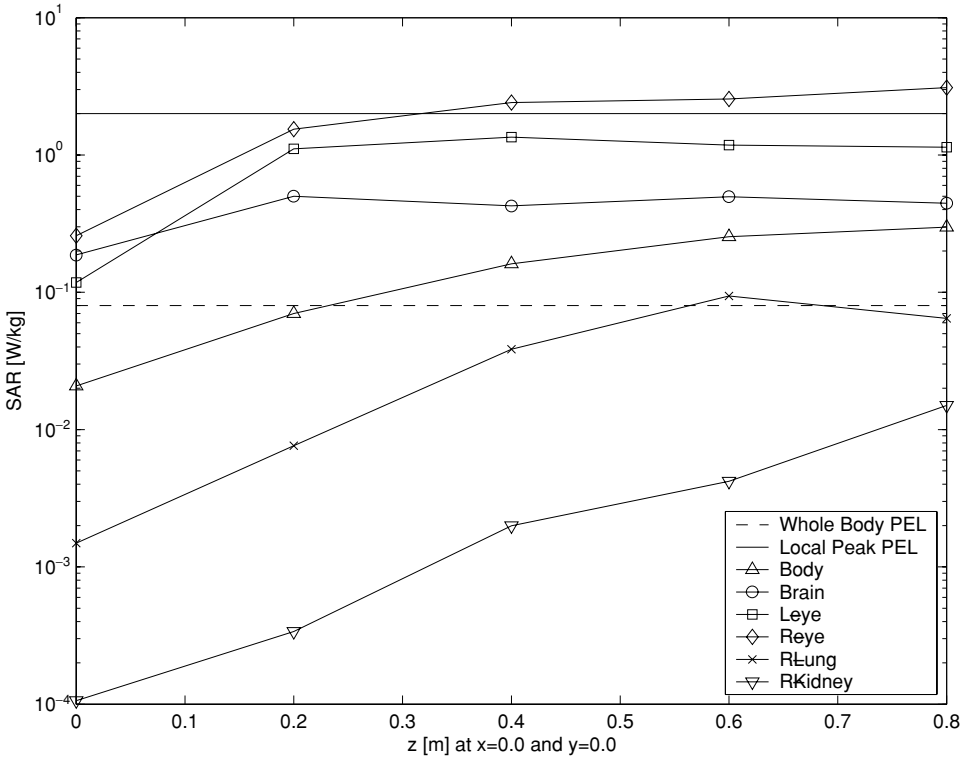


Figure 10.12 Average SAR specific absorption rate in different body organs for the z-direction (along antenna length), for the base-station/half-body problem. Details as in Fig. 10.11. Adapted from [25], ©2003 IEEE, reprinted with permission.

FETD formulation, and furthermore, this is a method which we can expect to see more of in the future.

10.7.1 Basic formulation and implementation

Basic finite element formulation

The following formulation is based on the second-order (curl-curl) wave equation approach, presented in [14, Section 12.1]:

$$\nabla \times \left[\frac{1}{\mu}\nabla \times \vec{E}(\vec{r},t)\right] + \varepsilon\frac{\partial^2}{\partial t^2}\vec{E}(\vec{r},t) + \sigma\frac{\partial}{\partial t}\vec{E}(\vec{r},t) = -\frac{\partial}{\partial t}\vec{J}_i(\vec{r},t), \qquad \vec{r} \in V \tag{10.35}$$

The boundary condition is

$$\hat{n} \times \left[\frac{1}{\mu}\nabla \times \vec{E}(\vec{r},t)\right] + Y\frac{\partial}{\partial t}\left[\hat{n} \times \hat{n} \times \vec{E}(\vec{r},t)\right] = \vec{U}(\vec{r},t), \qquad \vec{r} \in S \tag{10.36}$$

Y is the surface admittance of the boundary, $\hat{n}$ is the outward unit normal to S, and $\vec{U}$ is a known quantity representing the boundary source (if present).

The corresponding weak-form solution of the boundary value problem is given by:

$$\iiint_V \left\{ \frac{1}{\mu} \left[\nabla \times \vec{N}_i(\vec{r}) \cdot \nabla \times \vec{E}(\vec{r}, t) \right] + \varepsilon N_i(\vec{r}) \cdot \frac{\partial^2}{\partial t^2} \vec{E}(\vec{r}, t) \right.$$
$$\left. + \sigma N_i(\vec{r}) \cdot \frac{\partial}{\partial t} \vec{E}(\vec{r}, t) + N_i(\vec{r}) \cdot \frac{\partial}{\partial t} \vec{J}_i(\vec{r}, t) \right\} dV$$
$$+ \iint_S \left\{ Y \left[\hat{n} \times N_i(\vec{r}) \right] \cdot \left[\frac{\partial}{\partial t} \hat{n} \times \vec{E}(\vec{r}, t) \right] + \vec{N}_i(\vec{r}) \cdot \vec{U}(\vec{r}) \right\} dS = 0 \quad (10.37)$$

The electric field is expanded as

$$\vec{E}(\vec{r}, t) = \sum_{j=1}^{N} u_j(t) \vec{N}_j(\vec{r}) \quad (10.38)$$

with N the total number of unknowns, and $\vec{N}_j(\vec{r})$ the usual vector basis functions. Substituting this into Eq. (10.37), the following partial differential equation is obtained:

$$[T] \frac{\partial^2}{\partial t^2} \{u\} + ([R] + [Q]) \frac{\partial}{\partial t} \{u\} + [S]\{u\} + \{f\} = \{0\} \quad (10.39)$$

In the above, $\{u\} = [u_1, u_2, \ldots, u_N]^T$; and the matrices are given by:

$$T_{ij} = \iiint_V \varepsilon \vec{N}_i(\vec{r}) \cdot \vec{N}_j(\vec{r}) \, dV \quad (10.40)$$

$$R_{ij} = \iiint_V \sigma \vec{N}_i(\vec{r}) \cdot \vec{N}_j(\vec{r}) \, dV \quad (10.41)$$

$$Q_{ij} = \iint_S Y [\hat{n} \times \vec{N}_i(\vec{r})] \cdot [\hat{n} \times \vec{N}_j(\vec{r})] \, dS \quad (10.42)$$

$$S_{ij} = \iiint_V \frac{1}{\mu} [\nabla \times \vec{N}_i(\vec{r})] \cdot [\nabla \times \vec{N}_j(\vec{r})] \, dV \quad (10.43)$$

and $\{f\}$ is a column vector given by

$$f_i = \iiint_V \vec{N}_i(\vec{r}) \cdot \frac{\partial}{\partial t} \vec{J}_i(\vec{r}, t) \, dV + \iint_S \vec{N}_i(\vec{r}) \cdot \vec{U}(\vec{r}, t) \, dS \quad (10.44)$$

Equation (10.39) is an ordinary differential equation in the time domain and can be solved used a direct integration or finite difference method.

Before departing from this, it should be commented that these equations are essentially identical in form to those arising in standard frequency domain

formulations, and the matrices already computed within a typical finite element frequency domain code can be largely re-used.

Time integration

For the time domain discretization, the Newmark-β method is used. (An outline of the derivation of the method is given in Appendix B.) The equation to be solved at each time-step is the following:

$$\begin{aligned}\left\{\frac{1}{(\Delta t)^2}[T]+\frac{1}{2\Delta t}[T_\sigma]+\beta[S]\right\}\{u\}^{n+1}&\\ =\left\{\frac{2}{(\Delta t)^2}[T]-(1-2\beta[S])\right\}\{u\}^{n}&\\ -\left\{\frac{1}{(\Delta t)^2}[T]-\frac{1}{2\Delta t}[T_\sigma]+\beta[S]\right\}\{u\}^{n-1}&\\ -\left[\beta\{f\}^{n+1}+(1-2\beta)\{f\}^{n}+\beta\{f\}^{n-1}\right]&\end{aligned} \tag{10.45}$$

with

$$[T_\sigma]=[R]+[Q] \tag{10.46}$$

This can be more conveniently written as

$$[A]\{u\}^{n+1}=[B]\{u\}^{n}+[C]\{u\}^{n-1}-\left[\beta\{f\}^{n+1}+(1-2\beta)\{f\}^{n}+\beta\{f\}^{n-1}\right] \tag{10.47}$$

Clearly, the solution of this is:

$$\{u\}^{n+1}=[A]^{-1}\left([B]\{u\}^{n}+[C]\{u\}^{n-1}-\left[\beta\{f\}^{n+1}+(1-2\beta)\{f\}^{n}+\beta\{f\}^{n-1}\right]\right) \tag{10.48}$$

The matrix $[A]$ is time invariant, and may be factored once, each time-step requiring then just a backward and forward substitution to establish the next solution vector $\{u\}^{n+1}$. With $\beta \geq 0.25$, the method is unconditionally stable, i.e. the Courant limit does not apply.

10.7.2 Preliminary results

To test the time domain formulation, propagating a plane wave through the mesh is usually a good initial test, since one has a simple analytical solution to compare with the results. A differentiated Gaussian pulse with $\sigma = 1 \times 10^{-10}$ was used; as in Chapter 2, $m = 4\sigma$ was used. This produces a wideband pulse with

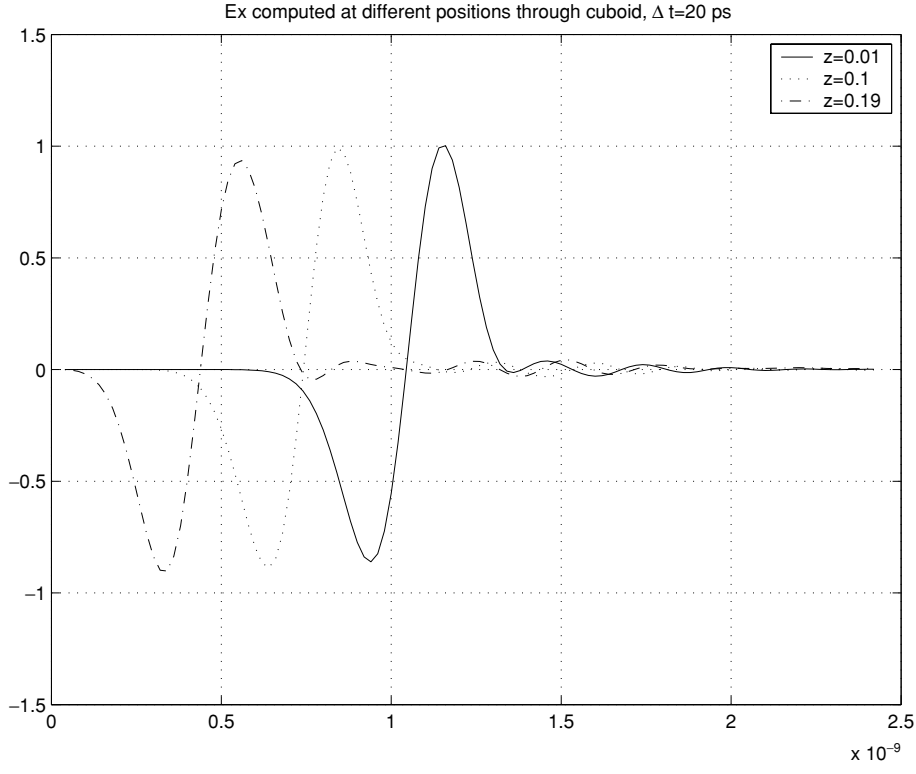


Figure 10.13 The differentiated Gaussian having propagated through a free-space "box" meshed using tetrahedral Whitney finite elements. $\Delta t = 20$ ps.

significant spectral content to around 3 GHz. A cuboidal free-space volume $0.1 \times 0.1 \times 0.2$ m^3 was meshed using tetrahedral elements; some 804 elements produced a mesh with an average edge length of about 0.0285 m. The plane wave was injected traveling in the $-\hat{z}$-direction. The result in Fig. 10.13 shows the plane wave at three points in the mesh. First $z = 0.19$ m is illuminated, then $z = 0.1$ m and finally $z = 0.01$ m. Measuring the distance between the first and last peaks shows a delay of 0.60 ps (within the accuracy with which the graph can be read); to cover a distance of 0.18 m at the speed of light takes 0.6 ps, so this is very accurate. In particular, considering the coarse mesh, the result is really surprisingly good; at 3 GHz the mesh density is less than four unknowns per wavelength. (At the center frequency of the signal, around 1.5 GHz, there are around seven, somewhat better but still a very coarse discretization using Whitney (CT/LN) elements.) Incidentally, the pulse may appear to have undergone a 180° phase reversal, but this was simply due to a coding convention. The late time signal is very likely a reflection from the absorbing boundary condition; the value is around 1/20 of the incident

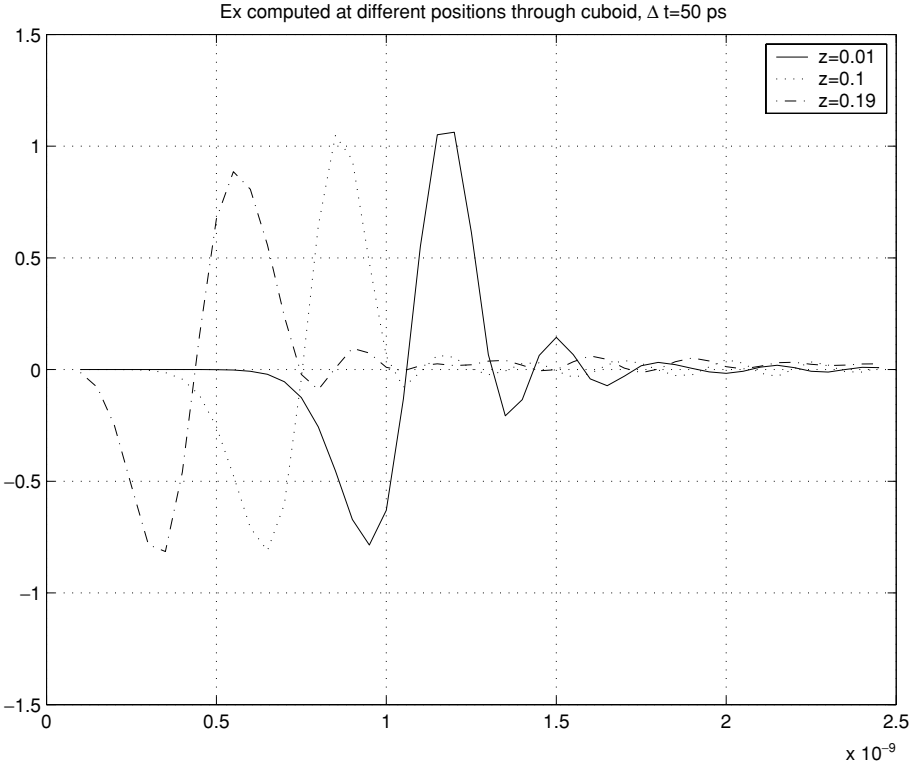


Figure 10.14 As for Fig. 10.13, but with $\Delta t = 50$ ps.

signal, or -26 dB, not by any means excellent absorber performance, but not out of line with what is expected from a first-order ABC.

Figures 10.14 and 10.15 show the results for $\Delta t = 50$ ps and 100 ps respectively. Clearly, the result in Fig. 10.15 is very poor, but it is still stable, and what is significant is the size of Δt. For a similar FDTD mesh with spatial step size 0.0285 m, the Courant limit would require $\Delta t < 54.8$ ps. The FETD code has remained stable at almost twice this limit. (Theoretically of course there *is* no limit for the Newmark-β scheme, but it is gratifying to have this confirmed by numerical experimentation.)

10.7.3 The FDTD method as a special case of the FETD

If the parameter β is set to zero, the Newmark algorithm reduces to the central difference algorithm. If, furthermore, we use Galerkin's method applied to edge elements defined on cubes rather than tetrahedra, and use 3D trapezoidal integration (i.e. sample the unknown function only at the center of each side when

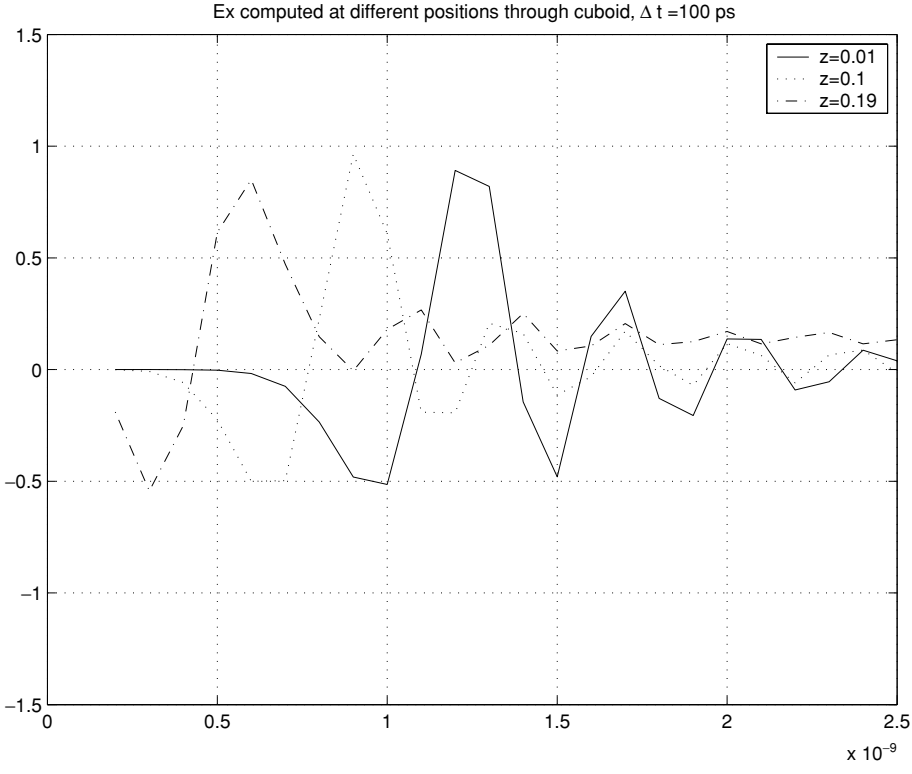


Figure 10.15 As for Fig. 10.13, but with $\Delta t = 100$ ps.

integrating), *the standard Yee FDTD algorithm emerges*. This may initially be a suprising result, since the FDTD appears to start from a different premise to the FETD, but has been noted by a number of workers. In the language of structural mechanics, this is a "lumping" method, where the mass and stiffness matrices are reduced to only diagonal elements. This of course implies that the matrix solution is trivial, which is why the FDTD method apparently has no matrix associated with it, and hence the explicit nature of the method. For more details, see, for example, [26].

10.8 Sparse matrix solvers

The development of an FEM code often goes through two major stages: the first concentrates on getting the code to work; the second concentrates on optimizing the code with regard to both memory usage and run-time. In Chapter 9, for instance, we focussed exclusively on the former. This process is frequently iterative, since new theoretical extensions must again be validated first, and then optimized. Furthermore, certain validation can only be undertaken once some optimization is

already in place. Since the finite element system matrices are usually highly sparse (i.e. have a very large number of zero entries), the efficiency of the sparse solver(s) is probably the single most important factor in determining the overall efficiency on an FEM code, since the matrix solution time usually dominates all other contributors to the total run-time, and FEM codes cannot work efficiently unless the sparsity of the finite element system matrix is properly exploited. There are two choices to make when exploiting sparsity.

Iterative solvers Iterative matrix solvers have the major advantage of requiring no additional memory beyond that required to store the coefficient matrices. They have the major disadvantage that each new solution of the system requires the iterative process to be repeated from scratch.

Direct solvers These are usually variations on the LU decomposition theme, and factor the matrix into a lower (and an upper, if the matrix is not symmetric) triangular matrix which permits very rapid subsequent solution of the system. However, they have the major disadvantage that the factorization process generates a number of non-zero entries in the matrix; this is known as "fill-in." Various methods are used to handle this; here, a method called "skyline storage" will be used.

Which choice is best is in general problem dependent; surprisingly, even in the case of a finite element time domain solver, where the same matrix is involved at each time-step, a direct solver is not necessarily the best solution. (In that specific case, the real valued system generated appears to be well conditioned, resulting in very rapid convergence of the iterative process. Dibben and Metaxas reported this in some of the earlier work in the field [28].) The memory overhead of the profiled storage scheme can also be prohibitive. For frequency domain solvers, the complex valued matrix can become very ill conditioned, and generally some form of preconditioning is required if an iterative solver is used.

First, two methods for storing a sparse matrix will be discussed.

10.8.1 Profile-in skyline storage

Consider the symmetrical matrix $[A]$:

$$[A] = \begin{bmatrix} a_{11} & a_{12} & & & \\ a_{21} & a_{22} & & a_{24} & a_{25} \\ & & a_{33} & 0 & 0 \\ & a_{42} & 0 & a_{44} & 0 \\ & a_{52} & 0 & 0 & a_{55} \end{bmatrix} \tag{10.49}$$

with $a_{12} = a_{21}$, etc. Here, an observation will be made. If this matrix is factored,

without pivoting, then *possible*[11] fill-ins will occur in $[L]$ to the right of the first non-zero entry in a row across to the diagonal (and similarly, in $[U]$ under the first non-zero entry in a column down to the diagonal). Hence, if all the zeros indicated above are stored, the factored matrix is guaranteed to fit into the data structure. This type of data structure is called a "skyline" matrix. There are several methods for storing the data: the one adopted here is called "profile-in," and what is stored is the elements in each row (column) from the first non-zero element to the diagonal (hence "in," since one moves inwards to the diagonal). Additionally, the index of the diagonal element is stored. For this matrix, the profile-in storage looks as follows:

$$AL = [a_{11}, a_{21}, a_{22}, a_{33}, a_{42}, 0, a_{44}, a_{52}, 0, 0, a_{55}]$$
$$IALDIAG = [1, 3, 4, 7, 11] \tag{10.50}$$

Since the matrix is symmetric, these structures could equally have been AU and $IAUDIAG$. The dimension of $IALDIAG$ is n. The dimension of AL is *at least* nz_s, the number of non-zeros in the lower (or upper) triangular half. Unfortunately, it is frequently many times this number.

10.8.2 Compressed row storage

Skyline storage is convenient when factoring a matrix but has a very high overhead, which only becomes clear when much larger finite element matrices are considered. The percentage of non-zero elements rapidly drops under one percent, but the profiled storage results in a very large number of zeros being stored, frequently an appreciable fraction of the original matrix. For iterative solvers, which require only a matrix-vector product, a much more efficient scheme is compressed row storage (CRS). Here, absolutely *only* the non-zero elements are stored. Since the storage requirements of a CRS matrix are so small, it is convenient to store each row completely, even if the matrix is symmetrical – this makes the sparse matrix-vector product far easier to write. In addition to an array storing the non-zero matrix elements, two other pointer arrays are needed. One stores the starting index of each row, the other stores the column indices. For the above matrix, the CRS equivalent is:

$$A\_CRS = [a_{11}, a_{12}, a_{21}, a_{22}, a_{24}, a_{25}, a_{33}, a_{42}, a_{44}, a_{52}, a_{55}]$$
$$JA = [1, 2, 1, 2, 4, 5, 3, 2, 4, 2, 5] \tag{10.51}$$
$$IA = [1, 3, 7, 8, 10, 12] \tag{10.52}$$

[11] Not all these positions will indeed be filled. More sophisticated methods do a better job of this process.

The $n+1$-element of IA is $nz+1$, where nz is the number of non-zeros. The dimension of IA is $n+1$, and the dimensions of both A_CRS and JA are nnz. These are known a priori, as soon as the matrix entries are known.

This storage scheme is also known as "general storage by rows."

A very similar storage scheme (and the one implemented in MATLAB) is compressed column storage; the procedure simply interchanges the storage direction. Since finite element matrices are generally symmetrical (unless one is dealing with non-reciprocal materials) the schemes are in practice essentially identical for finite element applications.

10.8.3 Implementation of matrix solution using these storage schemes

Sparse matrices are important for two reasons: firstly, to save memory, and secondly, to reduce run-time. Unfortunately, at the time of writing there is no analogy of the excellent public domain LAPACK routines for sparse matrices. If working with languages such as FORTRAN 90, sparse libraries may be available, either bundled with the compiler or for purchase separately.[12] However, actually *storing* the matrix in sparse form is a complex book-keeping task; one has firstly to establish the connections between all the degrees of freedom present (and this becomes increasingly more complex as higher-order elements are added) to determine the number of non-zero entries, following which the compressed matrices may then be filled as the matrix is assembled. Alternatively, and rather more easily, a full matrix may be generated first, and a sparse matrix then generated from this – the MATLAB function `sparse` does precisely this. However, the requirement to store the full matrix first wastes large amounts of memory, and is not practical for FEM codes designed for electromagnetically large problems.

It should be mentioned that especially higher-order elements appear to generate ill-conditioned matrices. When using iterative methods, such as conjugate gradient schemes (CG, Bi-CG), QMR and GMRES, convergence tends to be erratic. (For a description and discussion of these algorithms, see [14].) Some recent approaches have focussed on the use of more sophisticated preconditioners. Incomplete LU preconditioning is one possibility; another is the use of a direct solution of the CT/LN solution (which can generally be computed quite cheaply) as a preconditioner for the LT/QN matrix. This has been extended to higher-order schemes by [6]. Most of the more sophisticated preconditioner schemes trade off quicker convergence for increased matrix storage requirements.

Direct solvers have a place; generally, ill-conditioning is far less problematic, but the fill-ins can result in very large matrices indeed. Renumbering schemes

[12] As an example, the Compaq Visual Fortran (previously Digital Fortran) Fortran 90/95 compiler includes a library package called Compaq Extended Maths Libray (CXML). Included are direct and iterative solvers.

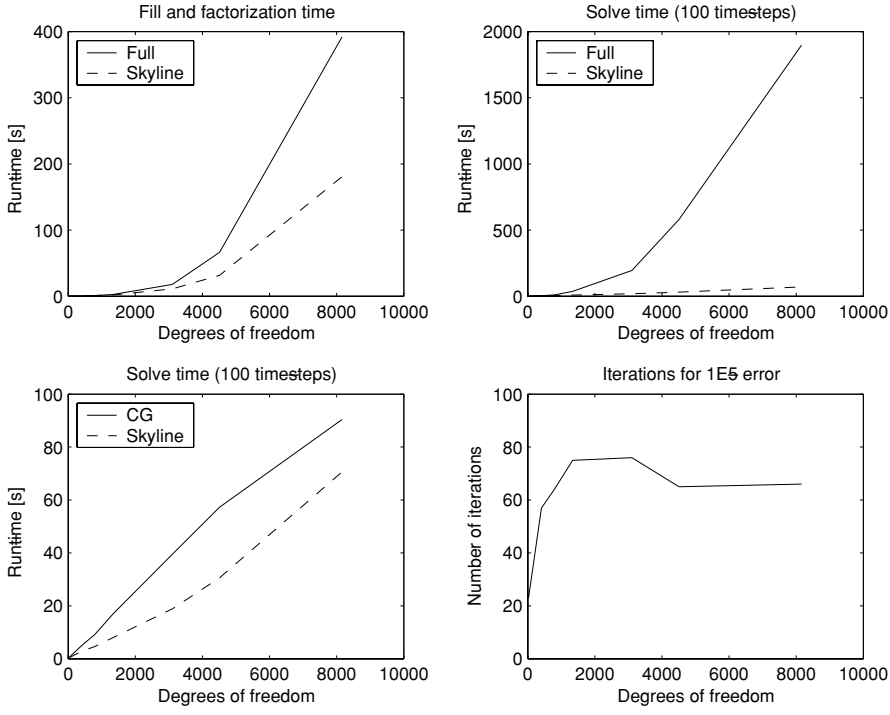


Figure 10.16 Solver times for the cubic example given in Section 10.7.2. Solve times are for 100 time-steps. The CG solver normalized residual target was 1×10^{-5}.

can ameliorate this, but unfortunately 3D finite element meshes tend to generate meshes with significant "bandwidth."

10.8.4 Results for sparse storage schemes

Some results illustrating the impact of sparse matrix solvers on an FEM code – in this case the FETD implementation by the author discussed in Section 10.7 – are shown in Figs. 10.16 and 10.17. The times shown in Fig. 10.16 compare the time using the sparse skyline or iterative CG solver (using CRS) with those of a full matrix solver (the latter not exploiting symmetry, i.e. worst case). Similarly, the memory percentages shown in Fig. 10.17 compare the relevant storage scheme with a full matrix scheme not using symmetry. The skyline storage is actually considerably less efficient than might be inferred from Fig. 10.17. Because with either of the sparse schemes, the $[B]$ and $[C]$ matrices in Eq. (10.47) can be (and are) stored in the much more efficient CRS form, whereas in a full matrix scheme they are of course stored as full matrices, there is already a saving by a factor of very close to three which is reflected in this figure. (CRS stored matrices require

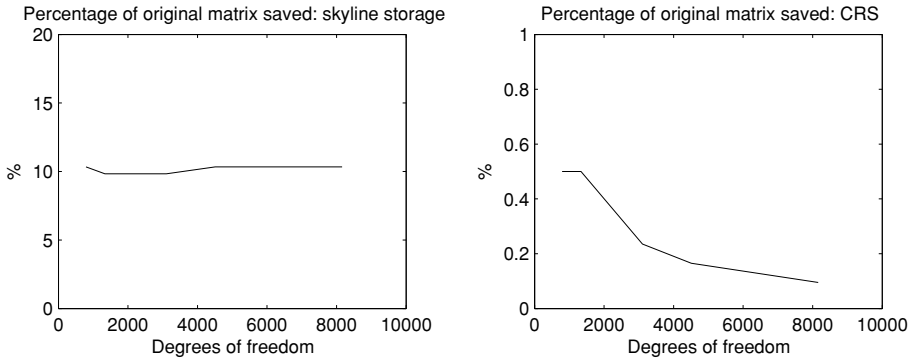


Figure 10.17 Memory usage for the cubic example given in Section 10.7.2. This is expressed as a percentage of the memory required by a full matrix scheme making no use of symmetry.

negligible storage compared to even skyline schemes, hence the factor of approximately three.)

These results are significant for code developers. Firstly, the timing results in Fig. 10.16 indicate that *any* sparse scheme is significantly better than none, as would be expected. Another interesting result is the comparison of the run-time of the CG solver with the skyline solver (lower left-hand plot in Fig. 10.16). The reason is that, for this problem at least, the number of iterations is almost constant, irrespective of problem size. This is shown in the lower right-hand graph in Fig. 10.16. (Although not shown on the figures, the number of iterations required also did not change from time-step to time-step.) One must be cautious of extrapolating this result to electromagnetically larger and more complex problems. These results were generated by increasingly refining the *same* problem. It is well known that the convergence rate of iterative solvers is a function of the ratio of maximum to minimum eigenvalues; furthermore, for any given electromagnetic problem, discretization beyond a certain point does not yield more significant eigenvalues. Electromagnetically larger problems may of course contain a wider eigenvalue spectrum. This note of caution notwithstanding, the results for the iterative solvers are highly encouraging, since no effort was made with these results to increase the rate of convergence, and an entire class of methods using various preconditioners exists which can still be applied. The memory savings of the iterative solver are of course very impressive (right-hand graph in Fig. 10.17) and imply that the limit on large problems is more likely to be run-time than memory.

A final comment on these results. The graphs comparing memory savings are actually in terms of memory *locations* required, rather than actual Mbytes of RAM used. The CXML libraries use double precision, so in RAM, the percentages are

twice that shown in the graphs. (Double precision was presumably used since the sparse factorizer does not apply pivoting. All the results were tested against a full matrix direct solver, and results were generally identical within working precision, around 4–5 significant figures after 100 time-steps. The CG solver normalized residual was set as 1×10^{-5} to ensure that inaccuracies did not accumulate during time-stepping.) This is a peculiarity of the particular implementation rather than the method per se. It also means that the computation times using the sparse schemes are slightly longer than would be the case if single precision were used.

10.9 A posteriori error estimation and adaptive meshing

As a final topic, some interesting recent work by Botha on the problem of estimating errors in the finite element solution will be outlined [12]. One of the main advantages of the FEM over the FDTD is that, theoretically at least, it is easy to *refine* a finite element mesh selectively. This can either be done by increasing the element order (p adaptation), decreasing the element size (h adaptation), or doing both ($h - p$ adaptation). In practice of course, mesh refinement does bring some complexity.

However, before one can undertake any form of mesh refinement, one needs an idea of in which part of the mesh the greatest benefits will be obtained. (Simply refining the entire mesh is of course a valid process, but computationally expensive. This is sometimes known as *uniform* mesh refinement.) It is here that the complex topic of *error estimation* comes to the fore. Here, one needs firstly to distinguish between a priori and a posteriori error estimates. The former are derived theoretically, and do not use the specific geometrical data represented by the mesh; examples are the analysis of dispersion error in a finite element or finite difference mesh. The latter are derived from the approximate solution, and it is these that will be considered here.

A posteriori error estimates can themselves be categorized as follows.

Explicit, residual-based These estimators are usually rigorously derived in the sense that the sum of the errors in each element is an upper bound on the error. (Here, we assume some suitable norm is available; often, the energy norm, discussed subsequently, is used.) Typically, field discontinuities at element edges and faces are evaluated.

Implicit, residual-based These estimators are based on the solution of local variational boundary value problems, usually on an element-wise basis. Usually, an estimate of the error is made using additional basis functions of higher order than

the initial solution. Since this is done on an element-by-element basis, this is not prohibitively expensive computationally – certainly not when compared to uniform refinement.

Estimation through post-processing These methods estimate the error in a derivative of the solution field, by comparing it with an improved version. Although this may seem counter-intuitive, some methods are available for computing improved versions of the solution field and its derivatives.

Targeted quantities These are also known as *goal-oriented* or *targeted* error estimation. They attempt to bound the error of a quantity based on some functional output of the the solution field. An example is the *S*-parameters discussed in the context of the waveguide formulation.

Botha's work focussed on explicit and implicit residual-based methods; the best results in general were obtained with the former, and a very brief summary of the method will now be presented.

10.9.1 Explicit, residual-based error estimators

Firstly, one must define the error in the solution as

$$\vec{e}_h = \vec{E} - \vec{E}_h \tag{10.53}$$

where $\vec{E}$ is the (usually unknown) exact solution of the problem, and $\vec{E}_h$ is the approximate, finite element computed solution. Botha showed that an estimate of the error in the CT/LN solution may be obtained as

$$||\vec{e}_h||^2_{E^a(V,\tau,1)} \leq C \sum_{i=1}^{N} \left(h_i^2 ||\vec{R}_v||^2_{L^2(K_i)} + 0.5 \sum_{f \subset \partial K_i} h_f ||\vec{R}_f||^2_{L^2(K_i)} \right) \tag{10.54}$$

N is the number of elements in the mesh; τ refers to the current discretization and solution, which will be used to compute the error indicators. The constant C is in general unknown, but is independent of solution field and source terms; error estimates usually contain such constants.

The term $||\vec{R}_v||^2_{L^2(K_i)}$ is the *volume* residual on element i, with volume K_i, measured in the L^2 norm – the space of square integrable functions. The volume residual in element i is computed from

$$\vec{R}_v = -\nabla \times \frac{1}{\mu_r} \nabla \times \vec{E}_h + k_0^2 \varepsilon_r \vec{E}_h - j k_0 Z_o \vec{J} \text{ in } K_i \tag{10.55}$$

In other words, this is the difference between the finite element computed solution, and the specified impressed current – in short, the residual of the vector wave equation. (If the latter is zero, then this term should of course be zero.) Were the solution exact, then this residual would be zero throughout the finite element volume.

The face residual on the surfaces of element i is computed from

$$\vec{R}_f = \hat{n}_{(12)} \times \left[\frac{1}{\mu_r^{(1)}} \nabla \times \vec{E}_h^{(1)} - \frac{1}{\mu_r^{(2)}} \nabla \times \vec{E}_h^{(2)} \right] \text{ on } f_m, \qquad m = 1, 4 \quad (10.56)$$

f_m is a specific face of the element, and the superscripts (1) and (2) indicate the elements shared by a particular face. In other words, this is the discontinuity in tangential magnetic field on the inter-element boundaries; again, were the solution exact, then this residual would be zero at all inter-element boundaries. Note that a special treatment, not shown here, is required at the Neumann boundary.

Whilst it may seem obvious that such residuals provide an indication of the error in the solution, some subtle mathematical arguments are required to show that the sum of residuals in Eq. (10.54) does indeed produce a bounded estimate of the overall error; the details may be found in [12, Chapter 5].

It should also be commented that the "norm" on $\vec{e}_h$ on the left-hand side of Eq. (10.54) is not a proper norm of the error field, but rather an approximate *energy norm*. (The reason that this does not conform to the usual definition is that this energy norm can be zero, without the field being zero. However, the converse is indeed true, i.e. the energy norm of a zero-valued field is zero.) The reason that this needs to be introduced is rooted in the complex valued nature of the functional. The approximate energy norm for space of maximum (but not necessarily complete) polynomial order p is defined as

$$||\vec{v}||_{E^a(V,\tau,p)} \equiv \frac{|\iiint_V \frac{1}{\mu_r} \nabla \times \vec{v} \cdot \nabla \times \vec{v} - k_0^2 \varepsilon_r \vec{v} \cdot \vec{v} \, dV|}{|\sum_{m=1}^{N} |v|^2_{(H^p(K_m))^3}|^{1/2}} \quad (10.57)$$

The term in the denominator, $|v|^2_{(H^p(K_m))^3}$, represents the vector Sobolev seminorm of derivative order p on domain K_m. Details of its evaluation may be found in [12].

10.9.2 An example of the application of an error estimator

An insightful example of the application of an error estimator may be found in [29]. The problem is an X-band waveguide filter (Fig. 10.18), with three metal septa along its center, normal to the broad walls of the waveguide (we have already

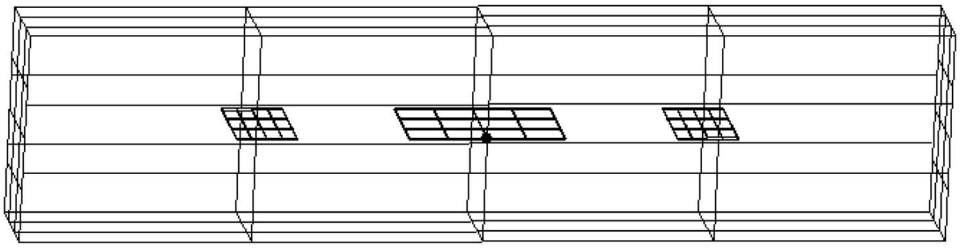


Figure 10.18 The waveguide filter geometry. After [29], ©2002 IEEE, reprinted with permission.

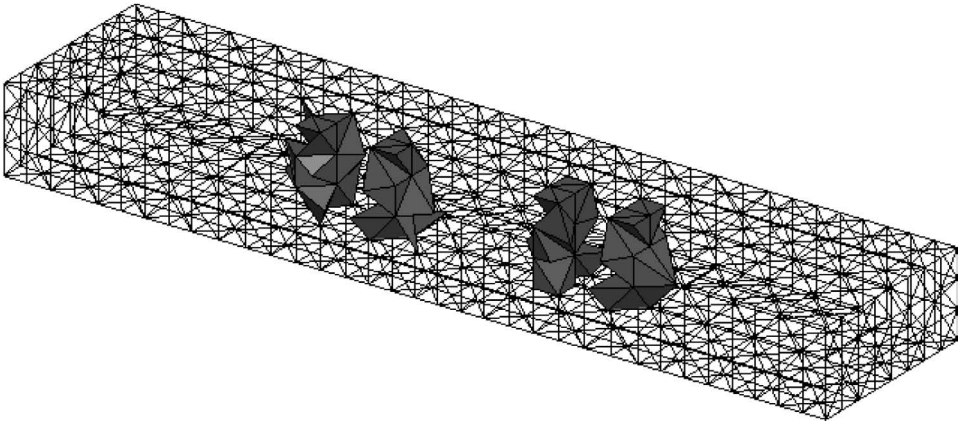


Figure 10.19 The waveguide filter: 2.5% elements with highest indicated error. After [29], ©2002 IEEE, reprinted with permission.

encountered this problem in Chapter 3). The explicit residual-based error indicator was applied, and results were obtained indicating where the computed errors were the highest. These are shown in Figs. 10.19–10.22. As expected, the errors cluster around the edges of the septa.

Once one has an indication of where the errors are most serious, one has various options to improve the solution. In this case, the results were used to drive a *p*-adaptive scheme, using the hierarchal elements of both mixed and complete order discussed earlier in this chapter. This permits a variety of possible schemes. The original solution was obtained with CT/LN elements; one possibility is to upgrade all the elements with the highest indicated error to QT/QN (which was the highest order available within the code); another is to upgrade to LT/QN elements; and a final possibility is a *graded* scheme, whereby the third of the elements with the highest errors are upgraded to QT/QN, then the next third to LT/QN and the last third to LT/LN. Results are shown in Fig. 10.23. The percentage error in center

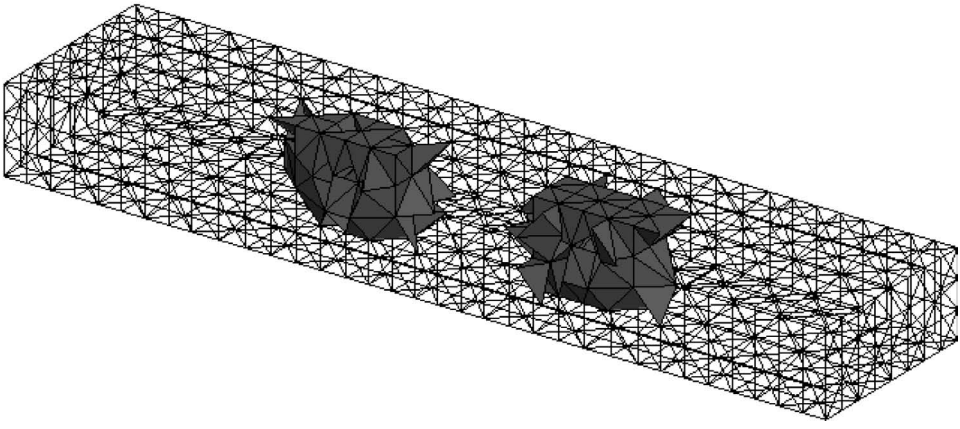


Figure 10.20 The waveguide filter: 5% elements with highest indicated error. After [29], ©2002 IEEE, reprinted with permission.

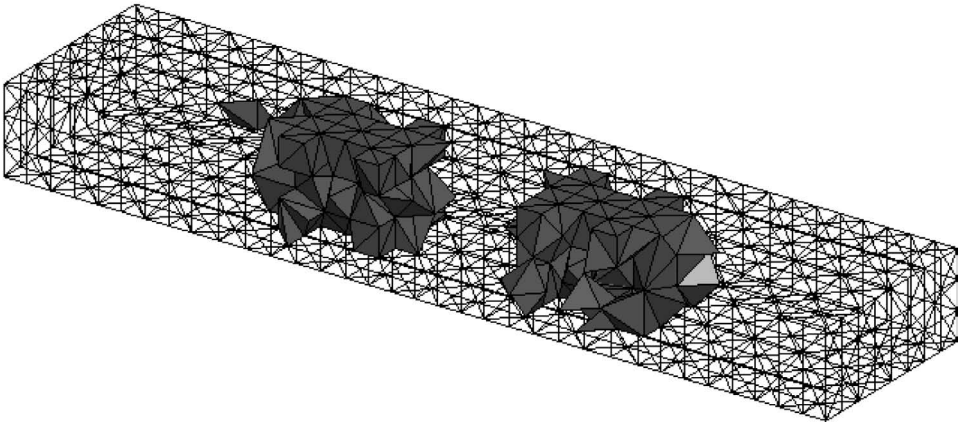


Figure 10.21 The waveguide filter: 10% elements with highest indicated error. After [29], ©2002 IEEE, reprinted with permission.

frequency is plotted against the number of degrees of freedom, which obviously grows as more and more elements are refined. Interestingly, the performance of the QT/QN and LT/QN schemes was similar, but the graded scheme was not very successful, primarily due to the inclusion of the LT/LN elements. It should be emphasized that this particular graded scheme is an heuristic one, and others could of course be proposed. These elements do not appear to be very beneficial in waveguide finite element analysis, a phenomenon noted and discussed in [4].

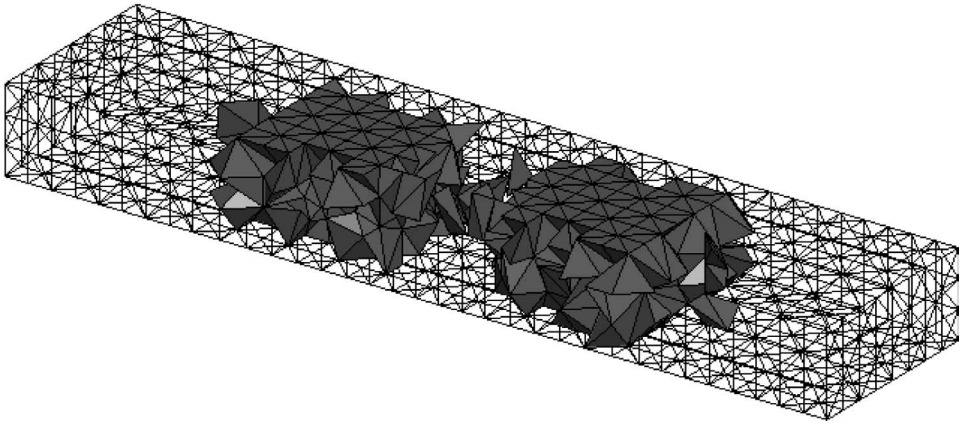


Figure 10.22 The waveguide filter: 20% elements with highest indicated error. After [29], ©2002 IEEE, reprinted with permission.

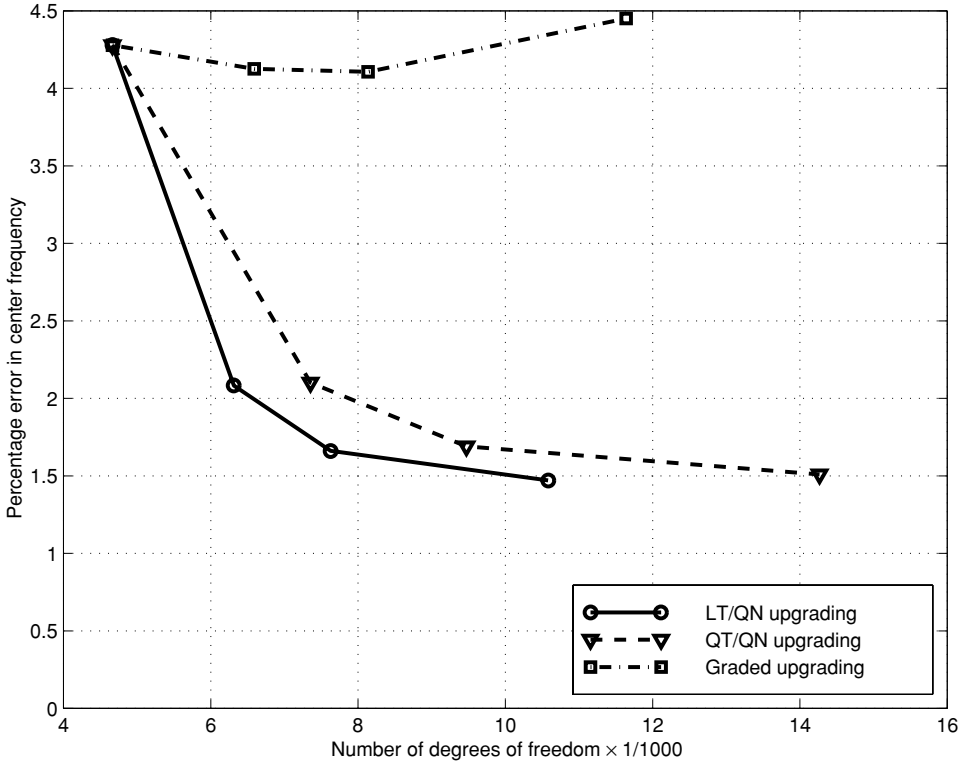


Figure 10.23 Waveguide filter center frequency versus number of degrees of freedom. A comparison of three upgrading schemes. After [29], ©2002 IEEE, reprinted with permission.

10.10 Further reading and conclusions

Most of the references cited in Section 9.9 are also of course relevant here. Jin's second edition [14] is probably the best single-volume reference in this regard, and includes a chapter devoted to time domain FEA and another to matrix solution.

In the context of higher-order vector elements, it should be noted that there is another school of thought regarding the construction of higher-order basis functions, which might be described as the *degree of freedom centered* approach (as opposed to that given in this chapter, which could be described as the *basis function centered* approach). Salazar-Palma *et al.* [30] use elements from the Nedelec polynomial space and enforce Lagrangian interpolatory properties on the degrees of freedom. This produces *interpolatory* elements with well-defined degrees of freedom at *points*, but at the time of writing, no-one had yet succeeded in doing this in general with higher-order *hierarchal* elements. Yioultsis and Tsiboukis take a similar degree of freedom centered approach, but working with simplex instead of Cartesian coordinates [31].

The work of Hiptmair should also be mentioned; he has also recently published a general scheme for the construction of higher-order elements, but from a far more mathematical viewpoint, and couched in the language of differential forms [32].

An important topic which we have not discussed is curvilinear elements. Whilst higher-order elements can do an excellent job of representing the fields very accurately, the limitations imposed by straight-sided triangular or tetrahderal elements in terms of accurate modelling of curved geometries can be very significant for many practical problems. There are in essence two questions to answer here: firstly, given a geometrical transformation, how does one implement this as a curvilinear element, and secondly, what transformation should be used. The former is the more theoretical issue, the latter a more practical one. Strangely, although curvilinear elements have been used in CEM, the literature on this is rather incomplete, in particular in the context of vector elements. The following references either deal with the issue in passing, touch on the issue, or summarize some aspects thereof [5, 33, 34, 35]. In the context of nodal elements, the discussion in [1, Chapter 7] is also useful. Recent work by Marais is amongst the more comprehensive treatments, although limited to two-dimensional problems [36].

Although an obvious application of the FEM, discontinuities in rectangular waveguides have not been as widely addressed in the literature as one might expect. Ise *et al.* [37] used "brick" elements of "first" order (CT/LN) to analyze both a dielectric post and a concentric step discontinuity in a rectangular waveguide; Jin presented a detailed formulation in [14, Chapter 8], also using CT/LN elements; Webb's review paper discussed a number of related issues [38]; and Pekel and Lee addressed theoretical aspects of mesh refinement using an empty piece of waveguides [39]. Scott addressed rotationally symmetric waveguides and obtained very

good results using special purpose higher-order elements [40]. The present author studied LT/QN elements in [15], and then considered the use of both mixed- and full-order elements in [4].

Ferrari has recently published a new formulation for the analysis of scattering discontinuities in waveguides, using an extended Huygens' principle [41]. The scatterer is discretized using finite elements, and the waveguide Green functions are used on the boundary of the scatterer, so this is a type of FEM/MoM hybrid. Geschke *et al.* reported the first successful implementation of the formulation in [42]; the details and many additional examples may be found in [43].

Regarding the FEM/MoM hybrid formulation, Peterson *et al.* [20] proposed that FEM/MoM hybrids be classified as either *outward-looking* or *inward-looking*. In the former case, the surface integral formulation represented by the MoM is used to augment the variational functional form of the vector wave equation, and this was the approach used in this chapter. It is also the most commonly encountered in the literature and in practice, since the effect is to increase the size of the FEM matrix somewhat. Furthermore, this outward-looking approach is also readily amenable to the introduction of approximate radiation boundary conditions, such as absorbing boundary conditions, rather than the rigorous Green function approach implicit in the MoM. Inward-looking formulations use the interior problem to constrain equivalent sources on the bounding surface. In this case, a large FEM matrix must first be solved before a smaller dense matrix can be constructed. Examples of the latter approach are the unimoment method, first suggested by Mei in 1974. More details on this topic may be found in [20, Chapter 3]. One problem with the outward-looking approach outlined here is that the matrix symmetry is generally destroyed. Botha and Jin have recently proposed a formulation which hybridizes the FEM and MoM on the formulation level (rather than on the matrix level) and which preserves the matrix symmetry [44]. It also offers some alternative approaches, including one which uses both $\vec{E}$ and $\vec{H}$ as working variable, permitting both fields to be computed to the same level of accuracy. The FEM/MoM formulation given in this chapter can suffer from internal resonances; the problem comes from the MoM treatment, and has already been mentioned in Section 6.9. The usual solution is to combine the EFIE and MFIE on the boundary. The formulation of Botha and Jin apparently also solves the problem, and computed results support the claim [44].

In terms of time domain formulations and applications, the paper by Gedney and Navsariwala [45] is one of the earlier in the field to discuss the FETD. It discusses an unconditionally stable formulation using the Newmark-β method. Although brief, it discusses most of the important topics and provides a stability analysis. The formulation is similar to that presented recently in [14]. The paper by Dibben and Metaxas [28] is also one of the earlier publications, and also uses the Newmark method. Together, these two represent well some of the earlier work on FETD

formulations and implementations within CEM. The review paper by Lee *et al.* [46] appeared in a special issue of the *IEEE Transactions on Antennas and Propagation* on numerical methods some years back, but presents a very good overview of the state of the art then – which it should be commented does not appear to have advanced enormously since, with the exception of boundary conditions. It presents an elegant theoretical framework for the general class of FETD methods, and is more general than the approach presented in [14], which focusses on the conventional curl-curl functional formulation. One very troublesome problem with FETD methods has been the development of efficient ABC-based boundary conditions; the paper by Jiao *et al.* [47] appears to have been the first to report a rigorous PML-type implementation for the FETD, although several workers, including the present author, have encountered problems with this implementation, in particular regarding stability.

Error estimation and mesh adaptation has a rather small bibliography in the engineering electromagnetics literature. Earlier work on this was done by Meyer, in the context of scalar, two-dimensional electromagnetic scattering and radiation problems, and results may be found in [48] and [49]. His results remain one of the most complete investigations of that specific problem. Some of the earlier work on the three-dimensional vector problem was done by Pekel and Lee [39].

In a field as large as finite elements, it is inevitable that there will be some important topics which we have not discussed at all. One which has produced important and interesting results is the analysis of dispersion error in finite element meshes (this is also sometimes called pollution error). The work of Cangellaris and Lee is an important reference here [50]; an overview of more recent work may be found in [14]. Also, modelling microwave ovens for commercial electro-heat applications has been a significant radio-frequency application of the FEM, using both eigenvalue and driven problem analysis. Details may be found in the books by Metaxas [51] and Chan and Reader [52].

Finally, serious students of the FEM who would like to read the large applied mechanics and applied mathematics literature will find that much of it uses the language of functional analysis. A very readable introduction is the text by Reddy [53], not least since it focusses on FEM formulations, unlike many of the more general texts on functional analysis.

References

[1] P. P. Silvester and R. L. Ferrari, *Finite Elements for Electrical Engineers*. Cambridge: Cambridge University Press, 3rd edn., 1996.

[2] J. P. Webb, "Edge elements and what they can do for you," *IEEE Trans. Magn.*, **29**, 1460–1465, March 1993.

[3] J. P. Webb, "Hierarchal vector basis functions of arbitrary order for triangular and tetrahedral finite elements," *IEEE Antennas Propag.*, **47**, 1244–1253, August 1999.
[4] D. B. Davidson, "An evaluation of mixed-order versus full-order vector finite elements," *IEEE Trans. Antennas Propag.*, **51**, 2430–2441, September 2003.
[5] R. D. Graglia, D. R. Wilton, and A. F. Peterson, "Higher order interpolatory vector bases for computational electromagnetics," *IEEE Trans. Antennas Propag.*, **45**, 329–342, March 1997.
[6] D. Sun and J. Lee, "Construction of nearly orthogonal Nedelec bases for rapid convergence with multilevel preconditioned solvers," *Siam. J. Sci. Comput.*, **23**(4), 1053–1076, 2001.
[7] J. P. Webb and B. Forghani, "Hierarchal scalar and vector tetrahedra," *IEEE Trans. Magn.*, **29**, 1495–1498, March 1993.
[8] J. C. Nedelec, "Mixed finite elements in $\Re^3$," *Numerische Mathematik*, **35**, 315–341, 1980.
[9] J. P. Webb, "Matching a given field using hierarchal vector basis functions," *Electromagnetics*, **24**, 113–122, January–March 2004.
[10] J. S. Savage, "Comparing high order vector basis functions," in *Proceedings of the 14th Annual Review of Progress in Applied Computational Electromagnetics*, pp. 742–749. Monterey, CA: March 1988.
[11] L. S. Andersen and J. L. Volakis, "Hierarchical tangential vector finite elements for tetrahedra," *IEEE Microwave Guided Wave Lett.*, **8**, 127–129, March 1998.
[12] M. M. Botha, Efficient finite element electromagnetic analysis of antennas and microwave devices: the FE-BI-FMM formulation and a posteriori error estimation for *p* adaptive analysis. Ph.D. Thesis, Department of Electrical and Electronic Engineering, University of Stellenbosch, 2002.
[13] N. Marcuvitz, *Waveguide Handbook*. London: Peter Peregrinus, on behalf of IEE, 1986. Originally published 1951.
[14] J. Jin, *The Finite Element Method in Electromagnetics*. New York: Wiley, 2nd edn., 2002.
[15] D. B. Davidson, "Higher-order (LT/QN) vector finite elements for waveguide analysis," Special Issue on Approaches to Better Accuracy/Resolution in Computational Electromagnetics, *Appl. Comput. Electromagn. Soc. J.*, **17**, 1–10, March 2002.
[16] D. M. Pozar, *Microwave Engineering*. New York: Wiley, 2nd edn., 1998.
[17] M. M. Botha and D. B. Davidson, "A quasi-static condition for enhancing *p*-adaptive, mixed-order element, FE analysis," *Electromagnetics*, **24**, 13–24, January–March 2004.
[18] J. P. Webb, "P-adaptive methods for electromagnetic wave problems using hierarchal tetrahedral edge elements," *Electromagnetics*, **22**, 443–451, July 2002.
[19] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, **30**, 409–418, May 1982.
[20] A. F. Peterson, S. L. Ray, and R. Mittra, *Computational Methods for Electromagnetics*. Oxford and New York: Oxford University Press and IEEE Press, 1998.
[21] J. Volakis, A. Chatterjee, and L. Kempel, *Finite Element Method for Electromagnetics: Antennas, Microwave Cicuits and Scattering Applications*. Oxford and New York: Oxford University Press and IEEE Press, 1998.
[22] M. M. Botha and D. B. Davidson, "Analyzing cavity backed, perforated substrate, microstrip patch antennas with a FMM, FE-BI hybrid formulation," in *Proceedings of the 2001 URSI International Symposium on Electromagnetic Theory*, pp. 627–629. May 2001.

[23] M. W. Ali, T. H. Hubing, and V. M. Drewniak, "A hybrid FEM/MOM technique for electromagnetic scatttering and radiation from dielectric objects with attached wires," *IEEE Trans. Electromagn. Compat.*, **39**, 304–314, November 1997.
[24] T. F. Eibert and V. Hansen, "3-D FEM/BEM-hybrid approach based on a general formulation of Huygens' principle for planar layered media," *IEEE Trans. Microwave Theory Tech.*, **45**, 1105–1112, July 1997.
[25] F. J. C. Meyer, D. B. Davidson, U. Jakobus, and M. A. Stuchly, "Human exposure assessment in the near field of GSM base-station antennas using the hybrid finite element/method of momemnts technique," *IEEE Biomed. Eng.*, **50**(2), 224–233, 2003.
[26] T. Rylander and A. Bondeson, "Stable FEM-FDTD hybrid method for Maxwell's equations," *Comput. Phys. Commun.*, **125**, 75–82, 2000.
[27] T. Rylander and A. Bondeson, "Stability of explicit-implicit hybrid time-stepping schemes for Maxwell's equations," *J. Comput. Phys.*, **179**, 426–438, July 2002.
[28] D. C. Dibben and R. Metaxas, "Time domain finite element analyis of multimode microwave applicators," *IEEE Trans. Magn.*, **32**, 942–945, May 1996.
[29] M. M. Botha and D. B. Davidson, "A posteriori error estimates for the FEM analysis of a waveguide filter," in *IEEE AFRICON-02 Proceedings*, pp. 541–544. October 2002.
[30] M. Salazar-Palma, T. K. Sarkar, L. E. García-Castillo, T. Roy, and Djordjević, *Iterative and Self-Adaptive Finite-Elements in Electromagnetic Modelling*. Boston, MA: Artech House, 1998.
[31] T. V. Yioultsis and T. D. Tsiboukis, "Development and implementation of second and third order vector finite elements in various 3-D electromagnetic field problems," *IEEE Trans. Magn.*, **33**, 1812–1815, March 1997.
[32] R. Hiptmair, "Canonical construction of finite elements," *Math. Comput.*, **68**, 1325–1346, May 1999.
[33] J. Wang and J. P. Webb, "Hierarchal vector boundary elements and p-adaption for 3-D electromagnetic scattering," *IEEE Trans. Antennas Propag.*, **45**, 1869–1879, December 1997.
[34] C. W. Crowley, P. P. Silvester, and H. Hurwitz, "Covariant projection elements for 3D vector field problems," *IEEE Trans. Magn.*, **24**, 397–400, 1988.
[35] A. F. Peterson and D. R. Wilton, "Curl-conforming mixed-order edge elements for discretizing the 2D and 3D vector Helmholtz equation," in *Finite Element Software for Microwave Engineering* (T. Itoh, G. Pelosi, and P. P. Silvester, eds.), Chapter 5, New York: Wiley, 1996.
[36] N. Marais, "Higher order hierarchal curvilinear triangular vector elements for the finite element method in computational electromagnetics." Master's Thesis, Department of Electrical and Electronic Engineering, University of Stellenbosch, April 2004.
[37] K. Ise, K. Inoue, and M. Koshiba, "Three-dimensional finite-element method with edge elements for electromagnetic waveguide discontinuities," *IEEE Trans. Microwave Theory Tech.*, **39**, 1289–1295, August 1991.
[38] J. P. Webb, "Finite element methods for junctions of microwave and optical waveguides," *IEEE Trans. Magn.*, **26**, 1754–1758, September 1990.
[39] Ü. Pekel and R. Lee, "An a posteriori error reduction scheme for the three-dimensional finite-element solution of Maxwell's equations," *IEEE Trans. Microwave Theory Tech.*, **43**, 421–427, February 1995.
[40] W. R. Scott, "Accurate modelling of axisymmetric two-port junctions in coaxial lines using the finite element method," *IEEE Trans. Microwave Theory Tech.*, **40**, 1712–1716, August 1992.

[41] R. L. Ferrari, "An extended Huygens' principle for modeling scattering from general discontinuities within hollow waveguides," *Int. J. Numerical Modelling: Electronic Networks, Devices Fields*, **14**(5), 411–422, 2001.
[42] R. H. Geschke, R. L. Ferrari, D. B. Davidson, and P. Meyer, "Application of extended Huygens' principle to scattering discontinuities in waveguide," in *IEEE AFRICON-02 Proceedings*, pp. 555–558. October 2002.
[43] R. Geschke, *Application of extended Huygens' principle to scattering discontinuities in waveguide*. Ph.D. Thesis, Department of Electrical and Electronic Engineering, University of Stellenbosch, 2004.
[44] M. M. Botha and J. Jin, "On the variational formulation of hybrid finite element–boundary integral techniques for time-harmonic electromagnetic analysis in 3D," *IEEE Trans. Antennas Propag.*, in press.
[45] S. D. Gedney and U. Navsariwala, "An unconditionally stable finite element time-domain solution of the vector wave equation," *IEEE Microwave Guided Wave Lett.*, **5**, 332–334, October 1995.
[46] J.-F. Lee, R. Lee, and A. C. Cangellaris, "Time-domain finite-element methods," *IEEE Trans. Antennas Propag.*, **45**, 430–442, March 1997.
[47] D. Jiao, J. Jin, E. Michielssen, and D. J. Riley, "Time-domain finite-element simulation of three-dimensional scattering and radiation problems using perfectly matched layers," *IEEE Trans. Antennas Propag.*, **51**, 296–305, February 2003.
[48] F. J. C. Meyer and D. B. Davidson, "A posteriori error estimates for two-dimensional electromagnetic field computations: boundary elements and finite elements," *Appl. Comput. Electromagn. Soc. J.*, **11**, 40–54, July 1996.
[49] F. J. C. Meyer and D. B. Davidson, "Adaptive-mesh refinement of finite-element solutions for two-dimensional electromagnetic problems," *IEEE Antennas Propag. Soc. Mag.*, **38**, 77–83, October 1996.
[50] A. C. Cangellaris and R. Lee, "On the accuracy of numerical wave simulations based on finite methods," *J. Electromagn. Waves Appl.*, **6**(12), 1635–1653, 1992.
[51] A. C. Metaxas, *Foundations of Electroheat: A Unified Approach*. Chichester: Wiley, 1996.
[52] T. V. C. T. Chan and H. C. Reader, *Understanding Microwave Heating Cavities*. Boston, MA: Artech House, 2000.
[53] B. D. Reddy, *Introductory Functional Analysis With Applications to Boundary Value Problems and Finite Elements*. New York: Springer-Verlag, 1998.

Appendix A

The Whitney element

The Whitney form $\lambda_i \nabla \lambda_j - \lambda_j \nabla \lambda_i$ is so widely used in vector elements that it is worth discussing in more detail. The development here is for two-dimensional elements, which has the benefit of simplicity; however, the essential argument is the same for the three-dimensional case.

Firstly, note the following very important property of the gradient of a simplex coordinate: it is constant, and is directed perpendicular to an edge. As an example, for the triangle shown in Fig. A1, $\nabla \lambda_1$ is perpendicular to edge 1, opposite vertex (node) 1. The formula is

$$\nabla \lambda_i = \frac{l_i}{2A} \hat{n}_i \tag{A.1}$$

with A the area of the triangle and $\hat{n}_i$ the normal on edge i.

We now investigate the properties of an approximation using the Whitney basis functions

$$\vec{E} \approx E_3(\lambda_1 \nabla \lambda_2 - \lambda_2 \nabla \lambda_1) + E_2(\lambda_1 \nabla \lambda_3 - \lambda_3 \nabla \lambda_1) + E_1(\lambda_2 \nabla \lambda_3 - \lambda_3 \nabla \lambda_2) \tag{A.2}$$

where E_1, E_2 and E_3 are constants whose physical meaning will shortly become clear.

We consider edge 3; anywhere on edge 3, $\lambda_3 \equiv 0$, and $\nabla \lambda_3$ is perpendicular to it. Finding the tangential component of the field on edge 3, we obtain:

$$\begin{aligned} \hat{e}_3 \cdot \vec{E} &= E_3(\lambda_1 \nabla \lambda_2 - \lambda_2 \nabla \lambda_1) + E_2 \cdot 0 + E_1 \cdot 0 \\ &= \vec{E}_{\tan}\big|_{\text{edge}_3} \end{aligned} \tag{A.3}$$

where the second and third terms are zero due either to $\lambda_3 = 0$ on edge 3 or $\nabla \lambda_3$ being perpendicular to this edge.

Using the sin rule for triangles (that the ratios of edge lengths and sines of opposite angles are equal) and Eq. (A.1), and the geometrical meaning of the dot product, we find

$$\begin{aligned} \vec{E}_{\tan}\big|_{\text{edge}_3} &= E_3(\lambda_1 \frac{\ell_2}{2A} \hat{n}_2 \cdot \hat{e}_3 - \lambda_2 \frac{\ell_1}{2A} \hat{n}_1 \cdot \hat{e}_3) \\ &= E_3 \frac{1}{2A}(\lambda_1 \ell_2 \sin\theta_1 + \lambda_2 \ell_1 \sin\theta_2) \end{aligned}$$

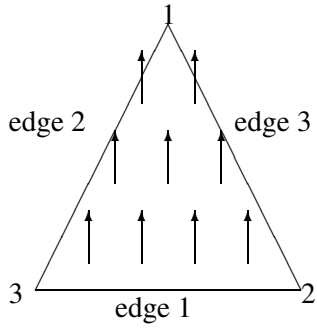


Figure A1 $\nabla\lambda_1$.

$$= E_3 \frac{1}{2A} \ell_2 \sin\theta_1 (\lambda_1 + \lambda_2)$$

$$= E_3 \frac{1}{2A} \ell_2 \sin\theta_1 \tag{A.4}$$

where the identity $\lambda_1 + \lambda_2 + \lambda_3 \equiv 1$ has been used in the last line (noting that $\lambda_3 = 0$ on this edge). $\sin\theta_i$ is the included angle at vertex i. Clearly, this is *constant*; now it is clear that E_3 is the tangential field on edge 3. Similar results follow for E_1 and edge 1, and E_2 and edge 2.

It can be simplified further by noting (from Fig. A1) that $\ell_2 \sin\theta_1$ is just the height of the triangle above edge 3. Since the area of the triangle is $A = (1/2)\ell h = (1/2)\ell_3 \ell_2 \sin\theta_1$, it follows that

$$\vec{E}_{\tan}\big|_{\text{edge}_3} = \frac{E_3}{\ell_3} \tag{A.5}$$

This is a well-known result, derived independently here. The general form for edge i is:

$$\vec{E}_{\tan}\big|_{\text{edge}_i} = \frac{E_i}{\ell_i} \tag{A.6}$$

If the vector basis function includes the edge length, as some published versions have, then the result is

$$\vec{E}_{\tan}\big|_{\text{edge}_i} = E_i \tag{A.7}$$

Appendix B

The Newmark-β time-stepping algorithm

The Newmark-β algorithm is rather more challenging to derive than is generally indicated in the literature, and its derivation is worth outlining. Most references cite the original paper by Newmark [1], which perhaps surprisingly does *not* derive the recurrence relationship, Eq. (10.45), which is generally associated with the name. This recurrence relationship was first given in a much later and very important paper by Zienkiewicz [2] published almost twenty years after the original method appeared. It is worth outlining the formulation, since it underlies the time-stepping approach implemented and does not appear to be available anywhere apart from Zienkiewicz's paper, which can be difficult to obtain.

The method is *only* relevant to the following differential equation representing a general second-order system with damping:

$$M\ddot{x} + C\dot{x} + Kx + f = 0 \tag{B.1}$$

It was derived for structural mechanics, where x is the displacement,[1] and $\dot{x}$ and $\ddot{x}$ the velocity and acceleration respectively. It is also based on a Taylor series expansion. For discrete samples at $t = n\Delta t$ and $t = (n+1)\Delta t$, the Taylor series expansion of the first derivative is

$$\dot{x}_{n+1} = \dot{x}_n + \ddot{x}_n \Delta t + \ddot{x}_n \frac{\Delta t^2}{2} + \cdots \tag{B.2}$$

Newmark proposed that for sufficiently smooth functions this can be evaluated as

$$\dot{x}_{n+1} = \dot{x}_n + \widehat{\ddot{x}}\Delta t \tag{B.3}$$

where $\widehat{\ddot{x}}$ represents some value of $\ddot{x}$ (in structural dynamics, the acceleration) intermediate between $\ddot{x}_n$ and $\ddot{x}_{n+1}$. This is where the parameter γ in Newmark's scheme is introduced:

$$\dot{x}_{n+1} = \dot{x}_n + (1-\gamma)\ddot{x}_n \Delta t + \gamma \ddot{x}_{n+1} \Delta t \tag{B.4}$$

Clearly, this is a second-order accurate scheme (for sufficiently smooth functions). The Newmark-β scheme uses $\gamma = 1/2$, hence the approximation of the second differential places equal weight on the values at n and $n+1$. The function itself (in structural

[1] The extension to two and three dimensions is straightforward, x is replaced by $\vec{x}$.

dynamics, the displacement) is approximated in a similar fashion, although in this case retaining an additional term in the Taylor series:

$$x_{n+1} = x_n + \dot{x}_n \Delta t + (1 - 2\beta)\ddot{x}_n \Delta t^2/2 + 2\beta \ddot{x}_{n+1} \Delta t^2/2 \tag{B.5}$$

Note that this is *not* the time integral of the approximate velocity, but rather the expansion of the displacement.

Most textbooks which discuss the technique indicate that by writing Eq. (B.1) at time-step $n + 1$

$$M\ddot{x}_{n+1} + C\dot{x}_{n+1} + Kx_{n+1} + f_{n+1} = 0 \tag{B.6}$$

and by also using Eqs. (B.4) and (B.5), one obtains values for x_{n+1}, $\dot{x}_{n+1}$ and $\ddot{x}_{n+1}$ in terms of x_n, $\dot{x}_n$ and $\ddot{x}_n$ and this is what Newmark implied in his original paper. This, however, is *not* the desired recurrence relation, Eq. (10.45). Zienkiewicz indicates the process required to obtain this. One writes the governing equation, Eq. (B.1), additionally at the time-steps n and $n - 1$; further, the integration formulas, Eqs. (B.4) and (B.5), are written at time-step $n - 1$, n. This provides seven equations in nine unknowns (three displacements, three velocities and three accelerations) from which all the velocities and accelerations can be eliminated to produce the conventional recurrence scheme:

$$\begin{aligned}
&\left[M + \gamma \Delta t C + \beta \Delta t^2 K\right] x_{n+1} \\
&+ \left[-2M + (1 - 2\gamma)\Delta t C + \left(\frac{1}{2} + \gamma - 2\beta\right) \Delta t^2 K\right] x_n \\
&+ \left[M + (-1 + \gamma)\Delta t C + \left(\frac{1}{2} - \gamma + 2\beta\right) \Delta t^2 K\right] x_{n-1} \\
&+ (\beta \Delta t^2) f_{n+1} \\
&+ \left(\frac{1}{2} + \gamma - 2\beta\right) f_n \Delta t^2 + \left(\frac{1}{2} - \gamma + 2\beta\right) f_{n-1} \Delta t^2 = 0
\end{aligned} \tag{B.7}$$

The derivation as outlined above does not appear ever to have been published, only the results.

Importantly, Zienkiewicz then proposed that this recurrence relation can alternatively be derived by applying a weighted residual process to Eq. (B.1). In addition to providing an independent check of Eq. (B.7), this procedure permits a far more general approach to the problem, and proceeds as follows. Firstly, x is approximated by the three-term expansion:

$$x \approx \sum_i N_i x_i, \qquad i = n - 1, n, n + 1 \tag{B.8}$$

Obviously, this will support a second-order expansion in time, as required by the second-order derivative in Eq. (B.1). The shape functions N_i (which represent the temporal expansion functions) are the usual node-based quadratic functions and are given in detail in [2, Eq. (10)]. It is further assumed that x_n and x_{n-1} are known, and that the only unknown is x_{n+1}. Hence only one weighting function is required. Replacing the interval $[-\Delta t; \Delta t]$ with the normalized variable $-1 \leq \xi = t/\Delta t \leq 1$, Zienkiewicz shows

that if we identify

$$\gamma = \left[\int_{-1}^{1} W\xi \, d\xi \Big/ \int_{-1}^{1} W \, d\xi \right] + \frac{1}{2}$$

$$\beta = \frac{1}{2} \int_{-1}^{1} W\xi(1+\xi) \, d\xi \Big/ \int_{-1}^{1} W \, d\xi = \frac{1}{2}\left(\gamma - \frac{1}{2}\right) + \frac{1}{2} \int_{-1}^{1} W\xi^2 \, d\xi \Big/ \int_{-1}^{1} W \, d\xi \tag{B.9}$$

then we obtain Eq. (B.7). This is a very useful result, since it makes the approximations involved far clearer. It also permits us to extend the Newmark scheme if necessary. Zienkiewicz used the result to show how a variety of weighting functions yield different three "time-stations" time-stepping schemes, of which the Newmark scheme is the most general. For instance, with $\gamma = \frac{1}{2}$ and $\beta = 0$, the weighting function is a Dirac delta at $t = n$, and the central difference scheme results. The Newmark-β scheme, on the limit of stability with $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{4}$, corresponds to the "average acceleration" scheme and the weighting function is the linear function $|\xi|$, zero at the center of the interval ($t = n$) and unity at the ends of the interval ($n - 1$ and $n + 1$) [2, Fig. 1]. It is also possible to produce higher-order schemes. Using cubic functions, for instance, a third-order scheme can be derived with four time-stations and Zienkiewicz also outlines this.

References

[1] N. M. Newmark, "A method of computation for structural dynamics," *J. Eng. Mech. Div., Am. Soc. Civil Eng.*, **85**, 67–94, July 1959.

[2] O. C. Zienkiewicz, "A new look at the Newmark, Houboult and other time stepping formulas, a weighted residual approach," *Earthquake Eng. Struct. Dynamics*, **5**, 413–418, 1977.

Appendix C

On the convergence of the MoM

Throughout this book, checking convergence numerically has been continually emphasized. However, we have not discussed the more theoretical issues of whether the underlying numerical formulations are indeed convergent, in the sense that the approximate numerical solution f^N of the continuous operator equation $Lf = g$ has the property $f^N \to f$ as $N \to \infty$. The aim of this appendix is to give a brief summary of the current status of this – which readers may be surprised to learn is far from a closed subject.

With the FDTD, the Lax equivalence theorem (discussed in Chapter 2) provides us with confidence that refining the FDTD mesh will indeed result in a convergent solution. With the FEM, work in applied mechanics has provided a rich set of convergence results – although we should note that convergence for high-frequency electromagnetics problems is often in terms of the energy norm, as discussed in Chapter 10. This is a slightly weaker statement of convergence, since the energy norm does not satisfy all the properties of the norm. Also, these proofs are usually in terms of interpolation error; as has been noted, dispersion (or pollution) error is a different problem specific to the differential equation based solvers, but can usually be controlled by adequate meshing. (Integral equation formulations using exact Green functions do not suffer from this problem of cumulative error resulting from dispersion error [1, p. 200].)

However, with the MoM, the problem has been studied somewhat less, presumably since the Green function is specific to electromagnetics. Rather surprisingly, only one form of operator has been rigorously shown to be convergent. (A recent summary may be found in [1, Chapter 5], which we summarize very briefly here.) This is the "identity plus compact" operator, of which the (two-dimensional) TE MFIE is an example. Proofs follow either via Galerkin's method, or via degenerate kernel analysis. Other types of operators are "compact" (the TM EFIE) and "unbounded" (the TE EFIE) – for neither of these do rigorous convergence proofs currently exist. (Incidentally, this nomenclature derives from the behavior of the eigenvalues of the operators.)

On the one hand, this is a somewhat disturbing situation, since important engineering designs are based on a field of mathematics which it transpires is far from complete. On the other hand, some forty years of development of the MoM has produced methods which have solved an enormous number of practical engineering problems with great accuracy, so it would appear most likely that what is missing is a convergence proof, rather than a fundamental problem. It would be satisfying were such proofs to be provided – or if they exist, popularized in the engineering literature. Here, we can but quote Peterson *et al.* [1, p. 224].

Our previous experience with integral equation formulations supports the notion that, if constructed with sufficient care, numerical solutions appear to converge under much more general conditions. Despite this observation, the authors are not aware of more general convergence proofs applicable to the specific integral operators arising in electromagnetic scattering.

On the subject of convergence, another topic which has aroused controversy is whether the Galerkin formulation is superior to other forms of testing procedure. The controversy arose because the far-zone characteristics of the antenna or scatterer can be expressed as quadratic functionals of the surface current, which can sometimes be defined in such a way that they have a stationary point at the true solution. The work of Peterson *et al.* [1, Section 5.12] has shed new light on this matter: they have shown that provided the testing functions have similar accuracy properties as the basis functions, the overall error from either a true stationary functional (as can be obtained using a Galerkin procedure) or a general continuous functional form is of similar magnitude. They took this further, by numerical tests using high-order spline basis and testing functions; their results support the contention that the error is actually a function of the combined order of basis (P) and testing function (Q), and that a Galerkin solution with $P = Q$ is no more accurate than a non-Galerkin solution with the same total $P + Q$.

Reference

[1] A. F. Peterson, S. L. Ray, and R. Mittra, *Computational Methods for Electromagnetics*. Oxford and New York: Oxford University Press and IEEE Press, 1998.

Appendix D

Suggested exercises and assignments

For graduate level courses, the following are suitable exercises. Most have been tested over the years by the author in a classroom environment. The approximate time required by a student to complete the assignment is also indicated, to assist in planning. This must be treated as only a guideline; it can change significantly, perhaps by as much as a factor of two either way, depending in particular on the programming ability of students or their familiarity with a particular code. The times given are for code development from scratch, and are based on the time the author and/or typical students have spent developing the routines or models; if some existing material is made available to the students, these can be greatly reduced. A number of MATLAB files, .pre files etc. are available to assist readers.

Chapter 2

1D FDTD analysis

1. Write a program to implement the 1D FDTD analysis of a transmission line, as discussed in this chapter. In particular, repeat the results given for the single-frequency source (Fig. 2.6), and also for the wideband source (Figs. 2.20, 2.21 and 2.22). Also investigate the effects of other termination conditions, such as a matched load. [20 hours]

Chapter 3

2D FDTD analysis

1. Repeat the TE_z scattering analysis discussed in this chapter using longer (in time) pulses and shorter pulses. Explain the time domain results obtained with each of these. Keep the grid size at 800×400 and $M = 1024$ so that run-times remain minutes rather than hours. [20–30 hours]
2. Modify the code to compute TM_z scattering from a cylinder. Does the TM_z polarization also show creeping waves? [10 hours]
3. Finally, extend the code (either TM or TE) to use the PML ABC. Since one needs to verify the PML, this is quite time consuming. [20–30 hours]

3D FDTD analysis

1. A ring hybrid, or *rat-race*, is a four-port device which functions as a 180° hybrid. (These are discussed in some detail in Section 10.4.) Descriptions may be found in many books on

microwave circuits, such as [1, Section 7.8]. Using a commercial package, predict the behavior of such a device fabricated in microstrip. [5–10 hours, depending on the ease of use of the package, and also the student's familiarity with it]

Partial solution

The device was designed to operate at 1.8 GHz. One must first obtain the dimensions for the microstrip; this was based on the example in [1, p. 163]. The substrate was chosen as $d = 1.27$ mm thick, $\epsilon_r = 2.2$. For the 50 Ω feedlines, the strip width to substrate thickness ratio W/d is 3.0981, hence $W = 3.91$ mm. For the 70.7 Ω components in the rat-race, W/d works out as 1.768, hence $W = 2.24$ mm. The effective dielectric constant in the 70.7 Ω section is 1.82 (it is slightly dependent on W/d). Hence, at 1.8 GHz, a quarter-wavelength in the dielectric is 30.8 mm. The average radius of the ring is thus 29.4 mm.

The four ports were modelled as discrete ports on the ends of sections of 50 Ω feedline approximately 15 mm long. In this case, the standard planar coupler template was used, and the space on top is five times the substrate thickness, as recommended by a MWS tutorial. In this case, however, open boundaries should be used (apart obviously from the ground plane).

At the design frequency, the results show the expected good match at port 1, 3 dB coupling to ports 2 and 3, and some 45 dB of isolation with respect to port 4 (Fig. D1).

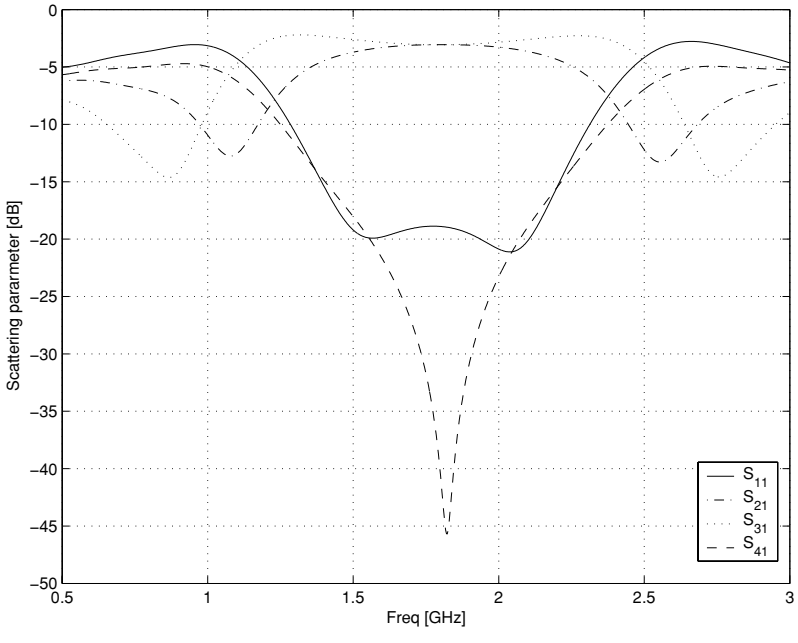


Figure D1 A MWS simulation of a rat-race hybrid coupler in microstrip.

Chapter 4

1D MoM analysis

1. Using the theory presented in Section 4.3, develop a thin-wire MoM code for a $\hat{z}$-directed dipole. Use sinusoidal weighting functions and collocation, so that Eq. (4.33) is applicable. Use both the delta-gap and magnetic frill source models, and replicate Fig. 4.3. [10 hours]

Chapter 5

Application of FEKO and NEC-2

This chapter consists largely of material which lends itself to assigning as tasks, as well as simple variants on the designs presented. Most of these will take 5–10 hours if the FEKO or NEC-2 models are developed from scratch. If time is pressing, a good alternative is to make available an existing model and ask students to modify them for a different geometry, frequency range etc.

Chapter 6

2D MoM analysis and hybrid methods

The material in this chapter does not readily lend itself to tasks.

Chapter 7

Sommerfeld potentials

1. Develop a code to replicate the results in Figs. 7.4 and 7.9, and then Figs. 7.10, 7.11 and 7.12. [40 hours]
2. Using this, develop an MoM code for a thin printed dipole and repeat the results of Fig. 7.13. [10 hours]
3. **As an advanced task** Instead of using the quasi-static approximation of Eq. (7.82), evaluate this rigorously as well. [Estimate 20 hours]

Chapter 8

Practical application of the Sommerfeld potentials

Again, this chapter consists largely of material which lends itself to assigning as tasks, as well as simple variants on the designs presented. Similar comments apply as for Chapter 5.

Chapter 9

2D finite elements

1. Using the theory developed in this chapter, develop a code to compute the TM eigenmodes. (Note that in this case, the problem is formulated in terms of the $\vec{H}$ field, and one uses the natural boundary condition on the waveguide walls.) [30–40 hours]

Chapter 10

Advanced FEM topics

As with Chapter 6, the material in this chapter does not readily lend itself to tasks.

Reference

[1] D. M. Pozar, *Microwave Engineering*. New York: Wiley, 2nd edn., 1998.

Appendix E

Useful formulas for simplex coordinates

Basic properties

On a triangle:

$$\lambda_1 + \lambda_2 + \lambda_3 = 1 \tag{E.1}$$

On a tetrahedron:

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1 \tag{E.2}$$

Integration

Integration over a triangle:

$$\iint_S \lambda_1^i \lambda_2^j \lambda_3^k \, dS = \frac{2!\, i!\, j!\, k!}{(2+i+j+k)!} A \tag{E.3}$$

A is the area of the triangle.

Integration over a tetrahedron:

$$\iiint_V \lambda_1^i \lambda_2^j \lambda_3^k \lambda_4^l \, dS = \frac{3!\, i!\, j!\, k!\, l!}{(3+i+j+k+l)!} V \tag{E.4}$$

V is the volume of the tetrahedron.

Gradient

Gradient on a triangle:

$$\nabla \lambda_i = \frac{l_i}{2A} \hat{n}_i \tag{E.5}$$

with A the area of the triangle, l_i the length of edge i and $\hat{n}_i$ the normal on edge i, pointing into the triangle.

Gradient on a tetrahedron:

$$\nabla\lambda_i = \frac{\vec{A}_i}{3V} \tag{E.6}$$

with V the volume of the tetrahedron and $\vec{A}_i$ the area of face $\{j, k, l\}$, with normal pointing into the tetrahedron.

Appendix F

Web resources

These sites, which include a number of commercial companies, were correct as of 2004 – web sites do change from time to time. This list is far from exhaustive, but gives a flavor of the variety of CEM products on offer, as well as the international technology base in this regard.

Ansoft Corporation A Pittsburgh, USA based company specializing in commercial FEM code suites.
URL: `http://www.ansoft.com/`

Applied Computational Electromagnetics Society An organization supporting the development, validation, and distribution of numerical EM modelling codes. Presently hosted by the University of Mississippi. Contains a number of very useful CEM links, including links to the public domain code NEC-2.
URL: `http://aces.ee.olemiss.edu/`

Computer Simulation Technology Based in Darmstadt, Germany, this company specializes in commercial Finite Integration Technique (largely FDTD) code suites, in particular MWS.
URL: `http://www.cst.de/` or
`http://www.cst-world.com/`

COMSOL A Swedish company, their main product is FEMLAB, a multi-physics FEM solver.
URL: `http://www.comsol.se/`

EMLIB This site, maintained at JPL, has been created for the free distribution of electromagnetics software and related information. This related information includes relevant conference information, a list of other EM sites, and a user-defined searchable directory of people working in the EM field.
URL: `http://emlib.jpl.nasa.gov/`

EMSS (Electromagnetic Software and Systems) Originally based in Stellenbosch, South Africa, this company now also has a German branch and US offices. Their main product is FEKO. They also provide a free GUI for NEC-2, Wiregrid for Windows.
URL: `http://www.emss.co.za/` or
`http://www.feko.info/`

MININEC website EM Scientific, Inc market a professional version of this code.
URL: `http://www.emsci.com/`

NEC-2 homepage An unofficial homepage with a number of links, as well as much of the NEC-2 documentation.
URL: `http://www.nec2.org/`

REMCON A US company, offering XFDTD, an FDTD-based package.
URL: `http://www.remcom.com/`

Poynting Software Another South African company, based in Johannesburg, offering SuperNEC.
URL: `http://www.supernec.com/`

The Schneider–Schlager FDTD database An exhaustive bibliography of published work dealing primarily with applications of, or extensions to, the FDTD method.
URL: `http://www.fdtd.org/`

Zeland Software Based in California, their best known product is probably IE3D, a planar and 3D MoM simulation package. It is widely used for microstrip structure simulation.
URL: `http://www.zeland.com/`

Index