



Wenhua Yu
Xiaoling Yang
Yongjun Liu
Raj Mittra
Akira Muto

ADVANCED
FDTD
METHODS

PARALLELIZATION,
ACCELERATION,
AND ENGINEERING
APPLICATIONS

Advanced FDTD Methods

**Parallelization, Acceleration,
and Engineering Applications**

For a listing of recent titles in the *Artech House Electromagnetics Series*,
turn to the back of this book.

Advanced FDTD Methods

**Parallelization, Acceleration,
and Engineering Applications**

Wenhua Yu
Xiaoling Yang
Yongjun Liu
Raj Mittra
Akira Muto



**ARTECH
HOUSE**

BOSTON | LONDON
artechhouse.com

Library of Congress Cataloging-in-Publication Data

A catalog record for this book is available from the U.S. Library of Congress.

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library.

ISBN-13 978-1-60807-176-0

Cover design by Igor Valdman

© 2011 ARTECH HOUSE

685 Canton Street

Norwood, MA 02062

All rights reserved. Printed and bound in the United States of America. No part of this book may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the publisher. All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Artech House cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

10 9 8 7 6 5 4 3 2 1

Contents

Preface	ix
Chapter 1 Computational Electromagnetic Methods	1
1.1 FDTD METHOD	1
1.1.1 FDTD Update Equations	1
1.1.2 Stability Analysis	6
1.1.3 Boundary Conditions	7
1.2 METHOD OF MOMENTS	11
1.3 FINITE ELEMENT METHOD	13
1.3.1 Scalar Formulation	14
1.3.2 Vector Formulation	14
1.4 FINITE INTEGRATION TECHNIQUE	15
References	17
Chapter 2 FDTD Optimization and Acceleration	19
2.1 INTRODUCTION TO CPU ARCHITECTURE	20
2.2 SSE INSTRUCTION SET	21
2.3 CACHE OPTIMIZATION	25
2.4 TASK PARALLELIZATION AND BUNDLING	26
2.5 PREFETCH	26
2.6 READING OR WRITING COMBINATION	27
2.7 MATERIAL LOOP-UP TABLE	28
2.8 NUMA OPTIMIZATION	28
2.9 IMPLEMENTATION OF VALU FDTD METHOD	30
References	35
Chapter 3 Parallel FDTD Method and Systems	37

3.1	PARALLEL FDTD METHOD	37
3.2	OPENMP FOR MULTICORE PROCESSORS	40
3.3	MPI TECHNIQUE	41
3.4	NETWORK CARD, SWITCH, AND CABLE	43
	References	46
Chapter 4	Electromagnetic Simulation Techniques	47
4.1	MESH GENERATION TECHNIQUES	47
4.2	BASIC SIMULATION PROCEDURE	50
4.3	DIPOLE ANTENNA	51
4.4	VIVALDI ANTENNA SIMULATION	55
4.5	BANDED MICROWAVE CONNECTOR	60
4.6	PARALLEL LINES	64
4.7	TWO-PORT ANTENNA	67
4.8	SLOT COUPLING	69
4.9	MICROWAVE FILTER	72
4.10	OPTIMIZATION AND PARAMETER SCAN	74
4.11	PERIODIC STRUCTURE SIMULATION	78
4.12	GROUND PENETRATING RADAR MODEL	81
4.13	MICROWAVE CONNECTOR	84
	References	87
Chapter 5	EM Simulation Software Benchmarks	89
5.1	BASIC STEPS IN EM SIMULATION	89
5.1.1	HFSS	89
5.1.2	CST	90
5.1.3	FEKO	90
5.1.4	GEMS	91
5.2	HARDWARE PLATFORMS	92
5.3	PATCH ANTENNA	92
5.4	VIVALDI ANTENNA	99

5.5	SCATTERING OF DIELECTRIC SPHERE _____	108
5.6	CELL PHONE ANTENNA _____	117
5.7	ELECTROMAGNETIC BANDGAP STRUCTURE _____	126
5.8	STANDARD SAR TEST _____	131
5.9	WAVEGUIDE FILTER _____	140
	References _____	146
Chapter 6	Large Multiscale Problem Solving _____	147
6.1	RADIO FREQUENCY PROTECTION _____	147
6.2	COMB-SHAPED ANTENNA ARRAY _____	153
6.3	TACKING REFLECTOR ANTENNA _____	159
6.4	PARTIAL SYMMETRIC PROBLEM SIMULATION _____	165
6.5	FLAT REFLECTOR ANTENNA _____	171
6.6	SIGNAL INTEGRITY _____	174
6.7	CARD READER EMI ANALYSIS _____	179
6.8	CLOAKING STRUCTURE _____	183
	References _____	185
Chapter 7	Summary _____	187
7.1	FDTD METHOD _____	187
7.2	FDTD CODE DEVELOPMENT _____	189
7.3	FDTD SIMULATION TECHNIQUES _____	190
7.4	HARDWARE PLATFORM _____	192
	References _____	193
Appendix A	Antenna Power and Efficiency _____	195
	References _____	199
Appendix B	Active Reflection Coefficient _____	201
	References _____	204
Appendix C	Total Active Reflection Coefficient _____	205
	References _____	206
Appendix D	MEG and ECC _____	207

D.1 PROPAGATION ENVIRONMENTS	207
D.2 ECC	208
D.3 MEG	208
References	209
Appendix E Lossy Dielectric Simulation Technique	211
References	212
Appendix F Circular Polarization Decomposition	213
References	215
Appendix G Vector Fitting Technique	217
References	219
Appendix H Partially Symmetric Problem Simulation	221
References	223
Appendix I Time-Domain Reflectometry (TDR)	225
References	227
Appendix J S-Parameter Extraction	229
J.1 MATCHED S-PARAMETER	229
J.2 NORMALIZED S-PARAMETER	230
J.3 MODE S-PARAMETER	232
References	233
Appendix K Debye Model Construction	235
References	237
Appendix L Geometry Mapping Technique	239
References	246
Appendix M PC Cluster Optimization	247
References	248
About the Authors	249
Index	253

Preface

Due to its flexibility and universality, the finite-difference time-domain (FDTD) method has become a major computational electromagnetic tool for various electromagnetic problems and phenomena. A large number of publications have reported on the FDTD method and engineering applications. Unlike existing books, this book introduces the optimization and acceleration techniques based on the vector arithmetic logic unit (VALU) that is a vector unit inside the regular CPU, and, demonstrates the difference of commercial software packages based on FDTD, finite integration technique (FIT), method of moments (MoM), and finite element method (FEM) methods for some typical problems in terms of accuracy, computer resource requirement, and performance. In addition, this book shows readers how to analyze and obtain the most important characteristics when the FDTD method is applied to solve practical problems.

Each core in the modern CPU has at least one VALU that can handle four operations simultaneously besides the floating-point unit (FPU) that can handle only one operation at one time. All the traditional electromagnetic software can only use floating-point unit or only one element of the VALU today. The VALU can accelerate the simulation four times without requiring any extra hardware devices. Since the VALU is a regular element inside a CPU, the FDTD code based on the VALU can fully use the L_1 , L_2 , and L_3 caches in the CPU to achieve the best performance. This book introduces practical FDTD simulation techniques and shows how to develop a high-performance FDTD code that is accelerated by using the VALU.

There exist a large number of commercial electromagnetic simulation software packages on the market, including popular electromagnetic software packages including GEMS based on the parallel conformal FDTD method, HFSS based on FEM, CST based on FIT, and FEKO based on MoM. Although they have their own strong areas, they overlap in various electromagnetic problem simulations. Though most electromagnetic software manufacturers have their own white book to introduce the background of their software packages, none of the current electromagnetic books compare the difference between existing software packages from the user's point of view. This book will fill this gap by comparing the performance of different software packages. This book starts with the basic concepts of the FDTD, MoM, FEM, and FIT methods, and is followed by the FDTD acceleration technique using the VALU and the benchmarks of the VALU and GPU. In Chapter 3, we introduce the parallel FDTD method and hardware systems. In Chapter 4, we present the important considerations in the

electromagnetic simulations through some representative examples. In Chapter 5, we present typical examples, which are used as the benchmarks simulated by using the commercial software based on the FDTD, FEM, MoM, and FIT methods. The authors do not take any responsibilities for the accuracy and performance experienced by the user. Finally, in Chapter 6, we present several large examples that are simulated by using the parallel conformal FDTD method.

The appendixes present some FDTD-related techniques that can help readers understand the results generated by using commercial software packages, or help extend the ability of the home-grown FDTD code.

We thank Cindy Zhou for her help in the manuscript preparation and encouragement for the publication of this book. We also thank T-Wave Co. (South Korea), Oriental Institute of Technology (Taiwan), High Performance Computing Center of CUC (China), CT Systemes Sarl (France), and HPCC of Xu Zhou Normal University (China), Zheng Li of Beijing Jiaotong University, Li Li of Communication University of China, Neng-Tien Huang of Yuan Ze University (Taiwan), Lai-Qing Ma of Foxconn Electronics, Duixian Liu of IBM, and Qinjiang Rao of RIM (Canada) for their help in code improvement and providing some test examples.

Chapter 1

Computational Electromagnetic Methods

Developed with computer technology, computational electromagnetics (CEM) has become the primary tool for the design of antenna and microwave circuit components, EMC/EMI analysis, and the prediction of radio propagation. A wide variety of packages are currently available, and some of the popular commercial EM software packages include high frequency simulation solver (HFSS) (<http://www.ansoft.com>) that is based on the finite element method (FEM); computer simulation solver (CST) (<http://www.cst.com>) microwave studio that utilizes the finite integration techniques (FIT); general electromagnetic solver (GEMS) (<http://www.2comu.com>) that is built on a parallelized conformal FDTD algorithm; and, FEKO (<http://www.feko.com>) that is based on the method of moments (MoM) and the fast multiple method (FMM). In addition, many wonderful home-grown CEM codes, for example, Professor Allen Taflové's group at Northwestern University, Professor Raj Mittra's group at Pennsylvania State University, Professor Qin Liu's group at Duke University, and Professor Yang Hao's group at Queen Mary University of London, have been successfully applied to the various practical problems.

1.1 FDTD METHOD

The FDTD method utilizes the central difference approximation to discretize the two Maxwell's curl equations, namely, Faraday's and Ampere's laws, in both the time and spatial domains, and then solves the resulting equations numerically to derive the electric and magnetic field distributions at each time step using an explicit leapfrog scheme. The FDTD solution, thus derived, is second-order accurate and is stable if the time step is chosen to satisfy the certain condition.

1.1.1 FDTD Update Equations

In Yee's scheme [1], the computational domain is discretized by using a rectangular grid. The electric fields are located along the edges of the electric elements, while the magnetic fields are sampled at the centers of the electric element surfaces and are oriented normal to these surfaces, this being consistent

with the duality property of the electric and magnetic fields in Maxwell's equations. A typical electric unit is shown in Figure 1.1.

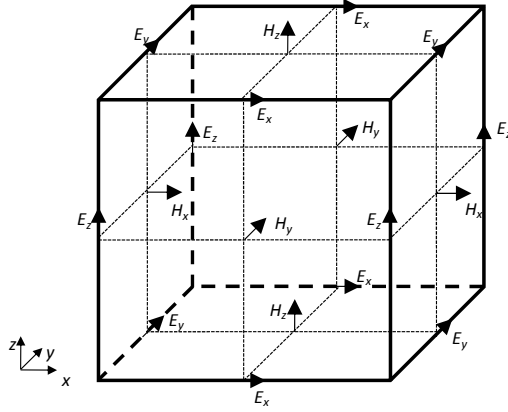


Figure 1.1 Positions of the electric and magnetic fields defined in Yee's FDTD scheme.

The FDTD utilizes rectangular pulses as base functions in both the time and spatial domains, implying that the electric field is uniformly distributed along the edge of electric element, while the distribution of the magnetic fields is uniform on the surface of the electric unit. In addition, in the time domain, the electric fields are sampled at times $n\Delta t$, and are assumed to be uniform in the time period of $(n-1/2)\Delta t$ to $(n+1/2)\Delta t$. Similarly, the magnetic fields are sampled at $(n+1/2)\Delta t$, and are assumed to be uniform in the period of $n\Delta t$ to $(n+1)\Delta t$. The FDTD algorithm constructs a solution to the following two Maxwell's curl equations:

$$\nabla \times \vec{E} = -\mu \frac{\partial \vec{H}}{\partial t} - \sigma_M \vec{H} \quad (\text{Faraday's law}) \quad (1.1a)$$

$$\nabla \times \vec{H} = \epsilon \frac{\partial \vec{E}}{\partial t} + \sigma \vec{E} \quad (\text{Ampere's law}) \quad (1.1b)$$

In the Cartesian coordinate system, we can rewrite (1.1a) and (1.1b) as the following six coupled partial differential equations:

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu_x} \left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} - \sigma_{Mx} H_x \right) \quad (1.2a)$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu_y} \left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} - \sigma_{My} H_y \right) \quad (1.2b)$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu_z} \left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - \sigma_{Mz} H_z \right) \quad (1.2c)$$

$$\frac{\partial E_x}{\partial t} = \frac{1}{\epsilon_x} \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - \sigma_x E_x \right) \quad (1.2d)$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\epsilon_y} \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - \sigma_y E_y \right) \quad (1.2e)$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\epsilon_z} \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma_z E_z \right) \quad (1.2f)$$

where ϵ and σ , μ and σ_M are the electric and magnetic parameters of the material, respectively. The anisotropic material can be described by using different values of dielectric parameters along the different directions. Equations (1.2a) to (1.2f) form the foundation of the FDTD algorithm for modeling interaction of the electromagnetic waves with arbitrary three-dimensional objects embedded in arbitrary media and excited by a given excitation source. Using the conventional notations, the discretized fields in the time and spatial domains can be written in the following format:

$$E_x^n(i + 1/2, j, k) = E_x((i + 1/2)\Delta x, j\Delta y, k\Delta z, n\Delta t) \quad (1.3a)$$

$$E_y^n(i, j + 1/2, k) = E_y(i\Delta x, (j + 1/2)\Delta y, k\Delta z, n\Delta t) \quad (1.3b)$$

$$E_z^n(i, j, k + 1/2) = E_z(i\Delta x, j\Delta y, (k + 1/2)\Delta z, n\Delta t) \quad (1.3c)$$

$$\begin{aligned} H_x^{n+1/2}(i, j + 1/2, k + 1/2) \\ = H_x(i\Delta x, (j + 1/2)\Delta y, (k + 1/2)\Delta z, (n + 1/2)\Delta t) \end{aligned} \quad (1.3d)$$

$$\begin{aligned} H_y^{n+1/2}(i + 1/2, j, k + 1/2) \\ = H_y((i + 1/2)\Delta x, j\Delta y, (k + 1/2)\Delta z, (n + 1/2)\Delta t) \end{aligned} \quad (1.3e)$$

$$\begin{aligned} H_z^{n+1/2}(i + 1/2, j + 1/2, k) \\ = H_z((i + 1/2)\Delta x, (j + 1/2)\Delta y, k\Delta z, (n + 1/2)\Delta t) \end{aligned} \quad (1.3f)$$

It is useful to note that the electric and magnetic fields in the discretized version are staggered in both time and space. For instance, the electric and magnetic fields are sampled at the time steps $n\Delta t$ and $(n+1/2)\Delta t$, respectively, and are also displaced from each other in space, as shown in Figure 1.1. Therefore, we need to interpolate the sampled electric and magnetic fields in order to measure the electric and magnetic fields in the continuous spatial and time domains. Ignoring this field-sampling offset in the Fourier transforms may result in a significant error at high frequencies.

Using the notations in (1.3a) to (1.3f), we can represent Maxwell's equations (1.2a) to (1.2f) in the following explicit formats [2–4]:

$$H_x^{n+1/2}(i, j+1/2, k+1/2) = \frac{\mu_x - 0.5\Delta t\sigma_{Mx}}{\mu_x + 0.5\Delta t\sigma_{Mx}} H_x^{n-1/2}(i, j+1/2, k+1/2) + \frac{\Delta t}{\mu_x + 0.5\Delta t\sigma_{Mx}} \left[\frac{E_y^n(i, j+1/2, k+1) - E_y^n(i, j+1/2, k)}{\Delta z} - \frac{E_z^n(i, j+1, k+1/2) - E_z^n(i, j, k+1/2)}{\Delta y} \right] \quad (1.4a)$$

$$H_y^{n+1/2}(i+1/2, j, k+1/2) = \frac{\mu_y - 0.5\Delta t\sigma_{My}}{\mu_y + 0.5\Delta t\sigma_{My}} H_y^{n-1/2}(i+1/2, j, k+1/2) + \frac{\Delta t}{\mu_y + 0.5\Delta t\sigma_{My}} \left[\frac{E_z^n(i+1, j, k+1/2) - E_z^n(i, j, k+1/2)}{\Delta x} - \frac{E_x^n(i+1/2, j, k+1) - E_x^n(i+1/2, j, k)}{\Delta z} \right] \quad (1.4b)$$

$$H_z^{n+1/2}(i+1/2, j+1/2, k) = \frac{\mu_z - 0.5\Delta t\sigma_{Mz}}{\mu_z + 0.5\Delta t\sigma_{Mz}} H_z^{n-1/2}(i+1/2, j+1/2, k) + \frac{\Delta t}{\mu_z + 0.5\Delta t\sigma_{Mz}} \left[\frac{E_x^n(i+1/2, j+1, k) - E_x^n(i+1/2, j, k)}{\Delta y} - \frac{E_y^n(i+1, j+1/2, k) - E_y^n(i, j+1/2, k)}{\Delta x} \right] \quad (1.4c)$$

$$E_x^{n+1}(i+1/2, j, k) = \frac{\varepsilon_x - 0.5\Delta t\sigma_x}{\varepsilon_x + 0.5\Delta t\sigma_x} E_x^n(i+1/2, j, k) + \frac{\Delta t}{\varepsilon_x + 0.5\Delta t\sigma_x} \left[\frac{H_z^{n+1/2}(i+1/2, j+1/2, k) - H_z^{n+1/2}(i+1/2, j-1/2, k)}{\Delta y} - \frac{H_y^{n+1/2}(i+1/2, j, k+1/2) - H_y^{n+1/2}(i+1/2, j, k-1/2)}{\Delta z} \right] \quad (1.4d)$$

$$E_y^{n+1}(i, j+1/2, k) = \frac{\varepsilon_y - 0.5\Delta t\sigma_y}{\varepsilon_y + 0.5\Delta t\sigma_y} E_y^n(i, j+1/2, k) + \frac{\Delta t}{\varepsilon_y + 0.5\Delta t\sigma_y} \left[\frac{H_x^{n+1/2}(i, j+1/2, k+1/2) - H_x^{n+1/2}(i, j+1/2, k-1/2)}{\Delta z} - \frac{H_z^{n+1/2}(i+1/2, j+1/2, k) - H_z^{n+1/2}(i-1/2, j+1/2, k)}{\Delta x} \right] \quad (1.4e)$$

$$E_z^{n+1}(i, j, k+1/2) = \frac{\varepsilon_z - 0.5\Delta t\sigma_z}{\varepsilon_z + 0.5\Delta t\sigma_z} E_z^n(i, j, k+1/2) + \frac{\Delta t}{\varepsilon_z + 0.5\Delta t\sigma_z} \left[\frac{H_y^{n+1/2}(i+1/2, j, k+1/2) - H_y^{n+1/2}(i-1/2, j, k+1/2)}{\Delta x} - \frac{H_x^{n+1/2}(i, j+1/2, k+1/2) - H_x^{n+1/2}(i, j-1/2, k+1/2)}{\Delta y} \right] \quad (1.4f)$$

We point out that, for simplicity, we have omitted the explicit indices for the material parameters, which share the same indices with the corresponding field components. Equations (1.4a) through (1.4f) do not contain any explicit boundary information, and we need to augment them with an appropriate boundary condition in order to truncate the computational domain. In the FDTD simulation, some of the commonly used boundary conditions include those associated with the perfect electric conductor (PEC), the perfect magnetic conductor (PMC), the absorbing boundary condition (ABC), and the periodic boundary condition (PBC). In addition to the above boundary conditions, we also need to handle the interfaces between different media in an inhomogeneous environment. In accordance with the assumption of the locations of the electric and magnetic fields, the magnetic field is located along the line segment joining the two centers of adjacent cells. Consequently, the effective magnetic parameter corresponding to this magnetic field is the weighted average of the parameters of the material that fills the two adjacent cells. Unlike the magnetic field, the loop used to compute the electric field is likely to be distributed among four adjacent cells; therefore, the effective electric parameter corresponding to this electric field is equal to the weighted average of electric parameters of the material that fills these four cells. In addition, the curved PEC and dielectric surfaces require the use of conformal FDTD technique [5, 6] for accurate modeling.

In recent years, research on the FDTD method has focused on the following three topics: improving the conventional FDTD algorithm and employing the conformal version instead in order to reduce the error introduced by the staircasing approximation; using a subgridding scheme in the FDTD technique to increase the local resolution [7–9]; and employing the alternative direction implicit (ADI) FDTD algorithm [10, 11] to increase the time step. In addition, new FDTD algorithms such as the multiresolution time-domain (MRTD) method [12] and the pseudospectrum time-domain (PSTD) technique [13] have been proposed with a view to lowering the spatial sampling. Yet another strategy, which has been found to be more robust as compared to the MRTD and PSTD, is to parallelize the conformal code [4, 14–16] and enhance it with either subgridding, the ADI algorithm, or both. The parallel FDTD algorithm gains the computational efficiency by distributing the burden on a cluster. It also enables one to solve large problems that could be beyond the scope of a single processor because of CPU time limitations.

1.1.2 Stability Analysis

One of the critical issues that we must address when developing a code that utilizes the marching-on-time technique is the stability of the algorithm. The stability characteristic of the FDTD algorithm depends upon the nature of the physical model, differencing technique employed, and the quality of the mesh structure. To understand the nature of the stability characteristic, we express the dispersion relationship as follows [4]:

$$\omega = \frac{2}{\Delta t} \sin^{-1} \left(c \Delta t \sqrt{\frac{1}{\Delta x^2} \sin^2 \left(\frac{k_x \Delta x}{2} \right) + \frac{1}{\Delta y^2} \sin^2 \left(\frac{k_y \Delta y}{2} \right) + \frac{1}{\Delta z^2} \sin^2 \left(\frac{k_z \Delta z}{2} \right)} \right) \quad (1.5)$$

If ω is an imaginary number, the electromagnetic wave, $\Psi(t, r) = \Psi_0 e^{j(\omega t - \vec{k} \cdot \vec{r})}$, either will attenuate rapidly to zero or will grow exponentially and become divergent, depending on whether the imaginary part is positive or negative. In order to ensure that ω is a real number instead, the expression inside the round bracket in (1.5) must satisfy the condition:

$$c \Delta t \sqrt{\frac{1}{\Delta x^2} \sin^2 \left(\frac{k_x \Delta x}{2} \right) + \frac{1}{\Delta y^2} \sin^2 \left(\frac{k_y \Delta y}{2} \right) + \frac{1}{\Delta z^2} \sin^2 \left(\frac{k_z \Delta z}{2} \right)} \leq 1 \quad (1.6)$$

Since the maximum possible value of the sine-square term under the square root is one, the time step must satisfy:

$$\Delta t \leq \frac{1}{c \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}}} \quad (1.7)$$

in order for the solution to be stable. The criterion above is called the stability condition for the FDTD method, and it is referred to as Courant condition (or the Courant, Friedrichs, and Lewy criterion) [17]. Equation (1.7) indicates that the time step is determined by the cell sizes in the x -, y -, and z -directions, and the speed of light in the medium.

To help the reader gain further physical insight into (1.7), we simplify (1.7) to a 1-D case, where the Courant condition is simplified to $\Delta t \leq \Delta x/c$. The time required for the field to propagate from the n th to $(n+1)$ th node is obviously $\Delta t = \Delta x/c$. In the FDTD simulations, let us suppose we choose a time step $\Delta t \geq \Delta x/c$, then the algorithm yields a nonzero field value before the wave can reach the $(n+1)$ th node, traveling at the speed of light. This would violate causality and result in an unstable solution.

1.1.3 Boundary Conditions

It is well known that boundary conditions play a very important role in FDTD simulations, because they are used to truncate the computational domain when modeling an open region problem. Though the original FDTD algorithm was proposed as early as 1966, it was not really used to solve practical problems until the early 1980s when Mur's absorbing boundary [18] was proposed. Though Mur's absorbing boundary condition is relatively simple, and while it has been successfully used to solve many engineering problems, it has room for improvement in terms of accuracy of the solution it generates. To improve its accuracy, Mei and Fang [19] introduced the so-called superabsorption technique, while Chew [20] proposed to employ Liao's boundary condition—both of which exhibit better characteristics than Mur's, especially for obliquely incident waves. However, many of these absorbing boundary conditions were found to suffer from either an instability problem or inaccurate solution, and the quest for robust and effective boundary conditions continued unabated until the perfectly matched layer (PML) was introduced by J. P. Berenger [21], and several other versions [22–24] have been proposed since then. In contrast to the other boundary conditions such as those of Mur and Liao, the PML can absorb the incoming waves at all frequencies as well as for all incident angles. The perfect electric conductor (PEC) is a natural boundary for electromagnetic waves, since it totally reflects the waves falling upon it. When the PEC condition is applied to truncate the FDTD computational domain, it simply forces the tangential electric fields on the domain boundary to be zero. In common with the PEC, the perfect magnetic conductor (PMC) also is a natural type of boundary condition for electromagnetic waves, and it is also totally reflecting. However, unlike the PEC, the PMC is not physical, but it is merely an artifice. Both the PEC and PMC are often used to take advantage of the symmetry of the object geometry with a view to reducing the size of the computational domain. In this section, we focus on the time convolution PML [25], one of the most popular PML formats. Although the PML is named to be an absorbing boundary condition, in fact, it is an anisotropic material—albeit mathematical—which is inserted in the periphery of the computational domain in order to absorb the outgoing waves.

Before introducing the PML boundary conditions, we first investigate their role in the FDTD simulations. In the FDTD, Maxwell's equations that govern the relationship between the electric and magnetic fields in the time and spatial domains are translated into difference equations, which do not explicitly contain any boundary information. It is necessary, therefore, to combine the difference equations with the appropriate boundary conditions in order to carry out the mesh truncation as a preamble to solving these equations. Generally speaking, there are two types of boundary conditions that we require in the FDTD simulations: the interface condition between different media, and the outer boundary condition for mesh truncation. In this chapter, we only discuss the latter, namely, the boundary that is used to truncate the computational domain.

The time convolution PML is based on the stretched coordinate PML [23, 24], and the six coupled Maxwell's equations in the time convolution PML can be written in the following form:

$$j\omega\varepsilon\tilde{E}_x + \sigma_x\tilde{E}_x = \frac{1}{S_y} \frac{\partial\tilde{H}_z}{\partial y} - \frac{1}{S_z} \frac{\partial\tilde{H}_y}{\partial z} \quad (1.8a)$$

$$j\omega\varepsilon\tilde{E}_y + \sigma_y\tilde{E}_y = \frac{1}{S_z} \frac{\partial\tilde{H}_x}{\partial z} - \frac{1}{S_x} \frac{\partial\tilde{H}_z}{\partial x} \quad (1.8b)$$

$$j\omega\varepsilon\tilde{E}_z + \sigma_z\tilde{E}_z = \frac{1}{S_x} \frac{\partial\tilde{H}_y}{\partial x} - \frac{1}{S_y} \frac{\partial\tilde{H}_x}{\partial y} \quad (1.8c)$$

$$j\omega\mu_x\tilde{H}_x + \sigma_{Mx}\tilde{H}_x = \frac{1}{S_z} \frac{\partial\tilde{E}_y}{\partial z} - \frac{1}{S_y} \frac{\partial\tilde{E}_z}{\partial y} \quad (1.8d)$$

$$j\omega\mu_y\tilde{H}_y + \sigma_{My}\tilde{H}_y = \frac{1}{S_x} \frac{\partial\tilde{E}_z}{\partial x} - \frac{1}{S_z} \frac{\partial\tilde{E}_x}{\partial z} \quad (1.8e)$$

$$j\omega\mu_z\tilde{H}_z + \sigma_{Mz}\tilde{H}_z = \frac{1}{S_y} \frac{\partial\tilde{E}_x}{\partial y} - \frac{1}{S_x} \frac{\partial\tilde{E}_y}{\partial x} \quad (1.8f)$$

To derive the update equations for the time convolution PML from (1.8a), we first take its Laplace transform to obtain the following equation in the time domain:

$$\varepsilon_x \frac{\partial E_x}{\partial t} + \sigma_x E_x = \bar{S}_y(t) * \frac{\partial H_z}{\partial y} - \bar{S}_z(t) * \frac{\partial H_y}{\partial z} \quad (1.9)$$

where \bar{S}_y and \bar{S}_z are the Laplace transforms of $1/S_y$ and $1/S_z$, respectively.

The time convolution PML is derived by converting (1.9) to a form that is suitable for explicit updating. Furthermore, to overcome the shortcomings of the split field and unsplit PMLs insofar as the effectiveness at the low frequencies and the absorption of the surface waves are concerned, we modify S_x , S_y , and S_z as follows:

$$S_x = K_x + \frac{\sigma_{x,\text{PML}}}{\alpha_x + j\omega\varepsilon_0}, \quad S_y = K_y + \frac{\sigma_{y,\text{PML}}}{\alpha_y + j\omega\varepsilon_0}, \quad S_z = K_z + \frac{\sigma_{z,\text{PML}}}{\alpha_z + j\omega\varepsilon_0}$$

where $\alpha_{x,y,z}$ and $\sigma_{x,y,z,\text{PML}}$ are real numbers, and K is greater than 1. \bar{S}_x , \bar{S}_y , and \bar{S}_z can be obtained from the Laplace transforms:

$$\bar{S}_x = \frac{\delta(t)}{K_x} - \frac{\sigma_x}{\varepsilon_0 K_x} \exp \left[- \left(\sigma_{x,\text{PML}} / \varepsilon_0 K_x + \alpha_{x,\text{PML}} \right) t u(t) \right] = \frac{\delta(t)}{K_x} + \xi_x(t) \quad (1.10)$$

$$\bar{S}_y = \frac{\delta(t)}{K_y} - \frac{\sigma_y}{\varepsilon_0 K_y} \exp \left[- \left(\sigma_{y,\text{PML}} / \varepsilon_0 K_y + \alpha_{y,\text{PML}} / \varepsilon_0 \right) t u(t) \right] = \frac{\delta(t)}{K_y} + \xi_y(t) \quad (1.11)$$

$$\bar{S}_z = \frac{\delta(t)}{K_z} - \frac{\sigma_z}{\varepsilon_0 K_z} \exp \left[- \left(\sigma_{z,\text{PML}} / \varepsilon_0 K_z + \alpha_{z,\text{PML}} / \varepsilon_0 \right) t u(t) \right] = \frac{\delta(t)}{K_z} + \xi_z(t) \quad (1.12)$$

where $\delta(t)$ and $u(t)$ are an impulse function and a step function, respectively. Substituting (1.11) and (1.12) into (1.9), we have:

$$\varepsilon_x \varepsilon_0 \frac{\partial E_x}{\partial t} + \sigma_x E_x = \frac{1}{K_y} \frac{\partial H_z}{\partial y} - \frac{1}{K_z} \frac{\partial H_y}{\partial z} + \xi_y(t) * \frac{\partial H_z}{\partial y} - \xi_z(t) * \frac{\partial H_y}{\partial z} \quad (1.13)$$

It is not numerically efficient to compute the convolution directly appearing in (1.13), and to address this issue we introduce a quantity $Z_{0y}(m)$ in order to calculate it efficiently, as follows:

$$\begin{aligned} Z_{0y}(m) &= \int_{m\Delta t}^{(m+1)\Delta t} \xi_y(\tau) d\tau \\ &= - \frac{\sigma_y}{\varepsilon_0 K_y^2} \int_{m\Delta t}^{(m+1)\Delta t} \exp \left[- \left(\sigma_{y,\text{PML}} / \varepsilon_0 K_y + \alpha_{y,\text{PML}} / \varepsilon_0 \right) \tau \right] d\tau \\ &= a_y \exp \left[- \left(\sigma_{y,\text{PML}} / K_y + \alpha_{y,\text{PML}} \right) \left(m\Delta t / \varepsilon_0 \right) \right] \end{aligned} \quad (1.14)$$

where

$$a_y = \frac{\sigma_{y,\text{PML}}}{\sigma_{y,\text{PML}} K_y + K_y^2 \alpha_y} \left(\exp \left[- \left(\sigma_{y,\text{PML}} / K_y + \alpha_y \right) \left(m\Delta t / \varepsilon_0 \right) \right] - 1 \right) \quad (1.15)$$

A similar expression can be derived for $Z_{0z}(m)$. Using (1.14) and (1.15), (1.13) can be rewritten as:

$$\begin{aligned} \varepsilon_x \varepsilon_0 \frac{E_x^{n+1}(i+1/2, j, k) - E_x^n(i+1/2, j, k)}{\Delta t} + \sigma_x \frac{E_x^{n+1}(i+1/2, j, k) + E_x^n(i+1/2, j, k)}{2} \\ = \frac{H_z^{n+1/2}(i+1/2, j+1/2, k) - H_z^{n+1/2}(i+1/2, j-1/2, k)}{K_y \Delta y} \\ - \frac{H_y^{n+1/2}(i+1/2, j, k+1/2) - H_y^{n+1/2}(i+1/2, j, k-1/2)}{K_z \Delta z} \\ + \sum_{m=0}^{N-1} Z_{0y}(m) \frac{H_z^{n-m+1/2}(i+1/2, j+1/2, k) - H_z^{n-m+1/2}(i+1/2, j-1/2, k)}{K_y \Delta y} \\ - \sum_{m=0}^{N-1} Z_{0z}(m) \frac{H_y^{n-m+1/2}(i+1/2, j, k+1/2) - H_y^{n-m+1/2}(i+1/2, j, k-1/2)}{K_z \Delta z} \end{aligned} \quad (1.16)$$

Finally, the updated formula of (1.16) takes the form:

$$\begin{aligned}
 \varepsilon_x \varepsilon_0 \frac{E_x^{n+1}(i+1/2, j, k) - E_x^n(i+1/2, j, k)}{\Delta t} + \sigma_x \frac{E_x^{n+1}(i+1/2, j, k) + E_x^n(i+1/2, j, k)}{2} \\
 = \frac{H_z^{n+1/2}(i+1/2, j+1/2, k) - H_z^{n+1/2}(i+1/2, j-1/2, k)}{K_y \Delta y} \\
 - \frac{H_y^{n+1/2}(i+1/2, j, k+1/2) - H_y^{n+1/2}(i+1/2, j, k-1/2)}{K_z \Delta z} \\
 + \psi_{\text{exy}}^{n+1/2}(i+1/2, j, k) - \psi_{\text{exz}}^{n+1/2}(i+1/2, j, k)
 \end{aligned} \tag{1.17}$$

where

$$\begin{aligned}
 \psi_{\text{exy}}^{n+1/2}(i+1/2, j, k) = b_y \psi_{\text{exy}}^{n-1/2}(i+1/2, j, k) \\
 + a_y \frac{H_z^{n+1/2}(i+1/2, j+1/2, k) - H_z^{n+1/2}(i+1/2, j-1/2, k)}{\Delta y}
 \end{aligned} \tag{1.18}$$

$$\begin{aligned}
 \psi_{\text{exz}}^{n+1/2}(i+1/2, j, k) = b_z \psi_{\text{exz}}^{n-1/2}(i+1/2, j, k) \\
 + a_z \frac{H_y^{n+1/2}(i+1/2, j, k+1/2) - H_y^{n+1/2}(i+1/2, j, k-1/2)}{\Delta z}
 \end{aligned} \tag{1.19}$$

$$b_x = \exp \left[- \left(\sigma_{x, \text{PML}} / K_x + \alpha_x \right) \left(\Delta t / \varepsilon_0 \right) \right] \tag{1.20a}$$

$$b_y = \exp \left[- \left(\sigma_{y, \text{PML}} / K_y + \alpha_y \right) \left(\Delta t / \varepsilon_0 \right) \right] \tag{1.20b}$$

$$b_z = \exp \left[- \left(\sigma_{z, \text{PML}} / K_z + \alpha_z \right) \left(\Delta t / \varepsilon_0 \right) \right] \tag{1.20c}$$

Equation (1.17) is the desirable updated equation that we have been seeking to retain the advantages of the unsplit PML, and overcome its drawback at the same time. In common with the conventional FDTD, the electric field updating inside the PML region only requires the magnetic fields around it and the value of ψ at the previous time step. The same statement is true for the magnetic field update as well. The time convolution PML does not require additional information exchange in the parallel FDTD simulation, over and above that in the conventional FDTD.

Suppose that x is the distance measured from the outer boundary of the computational domain, then the conductivity distribution in the PML region is given by:

$$\sigma(x) = \sigma_{\max} \left(\frac{d-x}{d} \right)^m \quad (1.21)$$

where the index m is taken to be either 2 or 4. In addition, in (1.21), d is the thickness of the PML region and σ_{\max} is the maximum value of the conductivity, which can be expressed as:

$$\sigma_{\max} = \frac{m+1}{200\pi\sqrt{\epsilon_r}\Delta x} \quad (1.22)$$

Suppose that y is a distance measured from the outer boundary of the computational domain, then the distribution of k_y is given by:

$$K_y(y) = 1 + (K_{\max} - 1) \frac{|d-y|^m}{d^m} \quad (1.23)$$

The implementation of the time convolution PML in the FDTD code is relatively simpler than most of the other types of PMLs. Also, this type of PML does not depend on the properties of the materials being simulated. In addition, it has a good performance at the low frequencies if we properly select $\alpha_{x,y,z}$ in the expression of \bar{S}_x , \bar{S}_y , and \bar{S}_z .

1.2 METHOD OF MOMENTS

During the last several decades, there has been a proliferation of numerical methods in electromagnetics. In 1968, Harrington identified a unifying concept for them and called it the method of moments (MoM) [26, 27], which we will briefly summarize next. The MoM is a widely used numerical technique for electromagnetic problems such as radiation and scattering problems. In this method, we solve the problem by first casting it into an operator equation, usually of the integral or integro-differential type. The unknowns are then expanded in terms of a finite number of well-chosen basis functions; this process is called discretization. A set of matrix equations is then generated by performing a symmetric or scalar product between the operator equation and a set of selected testing functions. The resulting matrix equation is solved by using a direct or iteration method. The latter is often the preferred choice for large matrices.

A typical operator equation can be expressed as:

$$Lf = h \quad (1.24)$$

where L is a linear operator, which may be either a differential or integral type, or a combination thereof; h is a known function; and f is an unknown function that will be solved for. These functions can be either a scalar or a vector, and their domain can be 1-D, 2-D, or 3-D space. The basic steps used to solve the electromagnetic problems using MoM are described as follows:

1. Select f_1, f_2, \dots, f_N as a set of basis functions, and let the unknown function $f(x)$ be expressed in terms of these as:

$$f(x) \approx a_1 f_1(x) + a_2 f_2(x) + \dots + a_N f_N(x) = \sum_{n=1}^N a_n f_n(x) \quad (1.25)$$

where a_n ($n = 1, 2, \dots, N$) are the coefficients, which will be solved for. This formula will become accurate for a larger number N . Introducing (1.25) into (1.24) we will get the following expression:

$$\sum_{n=1}^N a_n L f_n(x) \approx h(x) \quad (1.26)$$

2. Select a set of weight functions, w_1, w_2, \dots, w_n , and multiply each term in (1.25) with these functions, and integrate them in the domain space to generate a set of linear equations:

$$\sum_{n=1}^N Z_{mn} L a_n = b_n, \quad m = 1, 2, 3, \dots, N \quad (1.27)$$

where

$$Z_{mn} = \int w_m(x) L f_n(x) dx \quad (1.28)$$

$$b_m = \int w_m(x) h(x) dx \quad (1.29)$$

We can calculate the unknown function $f(x)$ when the coefficients a_n ($n = 1, 2, \dots, N$) in (1.25) is solved.

If a problem only includes the PEC objects, the unknown current functions are located on the surface of objects, MoM actually solves a 2-D problem though the objects are 3-D. For a problem involved with dielectric, MoM will not be efficient since the unknown functions will be distributed in a 3-D region. Though some commercial software packages today claim that the unknown functions can be on the surface of dielectric objects for the simple shaped objects, the simulation time spent on the dielectric objects will be increased significantly. Generally speaking, MoM is good for the scattering problems of pure PEC objects.

In the MoM method, the memory usage is in direct proportion to the N^2 if the number of unknowns is N , namely, $O(N^2)$. The number of floating operations using

the Gaussian elimination algorithm to solve the matrix equations will be in direct proportion to N^3 , namely, $O(N^3)$. However, the number of floating operations using the conjugate method or other iteration techniques to solve the matrix equations will be $O(N^2)$, where the index i is the preset number for the error control. Compared to the finite-difference methods, MoM is relatively simulation time and memory consuming.

The fast multiple method (FMM) is a mathematical technique that was developed to speed up the calculation of long-ranged forces in the N -body problem. It does this by expanding the system's Green function using a multiple expansion, which allows one to group sources that lie close together and treat them as if they are a single source.

The FMM has also been applied in accelerating the iterative solver in MoM as applied to computational electromagnetics problems. FMM was first introduced in this manner by Greengard and Rokhlin and is based on the multiple expansion of the vector Helmholtz equation. By treating the interactions between faraway basis functions using the FMM, the corresponding matrix elements do not need to be explicitly stored, resulting in a significant reduction in required memory. If the FMM is then applied in a hierarchical manner, it can improve the complexity matrix-vector product in an iterative solver from $O(N^2)$ to $O(N \log N)$. This has expanded the area of applicability of the MoM to far greater problems than were previously possible.

FMM, introduced by Greengard and Rokhlin, has been acclaimed as one of the top ten algorithms of the twentieth century. The FMM algorithm dramatically reduces the complexity of matrix-vector multiplication involving a certain type of dense matrix, which can arise out of many physical systems.

1.3 FINITE ELEMENT METHOD

The finite element method [28, 29] has become a powerful tool throughout engineering for its flexibility and versatility, being used in complicated structural, thermal, fluid flow, semiconductor, and electromagnetic problems. This method is particularly advantageous for electromagnetic field problems, because of its applicability to waveguides with arbitrary shape, arbitrary refractive index profile, and even anisotropy. However, FEM is more suitable for the closed space problems such as waveguide and cavities because it does not have a high performance absorbing boundary as FDTD does.

Finite element formulations are usually established via a variational or a Galerkin approach. The latter is more flexible, but when it is possible, it is advantageous to take a variational approach, especially when one global parameter (like propagation constant) is needed. Steps of the finite element method are listed here:

1. Find an equivalent variational equation for the differential equation.
2. Discretize the whole domain of the problem into a number of sub-domains (for the region where the function changes sharply, the lattice

- should be densified). Number the elements and the nodes sequentially.
3. Interpolate the function within each subdomain by the function values at the surrounding nodes.
 4. Derive the linear equations from the equivalent variational problem.
 5. Solve the linear equations.

We will briefly introduce the basic idea of finite element methods; namely, scalar finite element method and vector finite element method.

1.3.1 Scalar Formulation

Several different variational formulations have been proposed for use with the finite element method. The simplest is a scalar one. In the finite element approximation, the primary dependent variables are replaced by a system of discretized variables over the domain of consideration. We first divide the entire domain into the elements, usually triangles or quadrilaterals. Triangles are commonly used because they are easy to adapt to complex shapes. Using many elements, any cross section with a complex boundary and with arbitrary permittivity distribution can be accurately approximated. The simplest triangular element assumes a linear interpolation between the field values at the vertices of the triangle. By expressing the fields in terms of nodal values and with the assumed shape functions, the resulting field components are continuous over the whole domain. For example, to calculate the potential ϕ generated by a charge ρ distributed in the domain Ω , we need to solve the Poisson equation

$$\nabla \cdot (\epsilon \nabla \phi) = -\rho \quad (1.30)$$

To find these nodal fields, the usual (Rayleigh-Ritz) procedure is to force stationarity of the functional with respect to each nodal variable. This yields a matrix eigenvalue equation, with vector x of nodal variables:

$$K\phi = b \quad (1.31)$$

where K is an $N \times N$ symmetric matrix; ϕ is an $N \times 1$ vector, which is unknown; and b is a known vector, which is determined by the charge and boundary condition. This equation can be solved by one of various standard subroutines to get K .

1.3.2 Vector Formulation

The single scalar formulation can be generalized to vector formulation, which is very important because most 3-D problems need to be expressed in a vector field. However, some vector formulations are affected by spurious or nonphysical solutions that appear mixed with the correct ones in the computations.

For example, we calculate the field distribution in the domain Ω generated by a current density J_{imp} , and the domain Ω is filled with the material ε and μ . The field in the domain Ω should be satisfied with the equation:

$$\nabla \times \left(\frac{1}{\mu_r} \nabla \times \vec{E} \right) - k_0^2 \varepsilon_r \vec{E} = -jk_0 Z_0 J_{imp} \quad (1.32)$$

where ε_r and μ_r are the permittivity and permeability in free space, $K_0 = \omega \sqrt{\varepsilon_0 / \mu_0}$ is the propagation constant in free space, and $Z = \sqrt{\varepsilon_0 / \mu_0}$ is the wave impedance.

In accordance with the scalar formulation, applying the boundary condition on (1.32), we can get the following equation through a weighted function:

$$\vec{K} \vec{E} = \vec{b} \quad (1.33)$$

where \vec{E} is the unknown vector that will be solved, \vec{K} is the matrix associated with the system, and the \vec{b} is the known vector that is determined by the charge and boundary condition.

1.4 FINITE INTEGRATION TECHNIQUE

The finite integration technique (FIT) is a consistent formulation for the discrete representation of Maxwell's equations on spatial grids. First proposed by Weiland [30] in 1977, the finite integration technique can be viewed as a generalization of the FDTD method. It is also similar to the finite element method. Weiland proposed an exact algebraic analog to Maxwell's equations that guarantee physical properties of computed fields and lead to a unique solution. By discretizing the integral form of Maxwell's equations on a pair of dual interlaced discretization grids, the finite integration technique generates so-called Maxwell's grid equations (MGEs) that guarantee the physical properties of computed fields and lead to a unique solution.

$$Ce = -\frac{d}{dt}b \quad (1.34)$$

$$\tilde{C}h = \frac{d}{dt}d + j \quad (1.35)$$

$$Sb = 0 \quad (1.36)$$

$$\tilde{S}b = q \quad (1.37)$$

where e is the electric voltage between the grid points and $h\hat{A}$ is the magnetic voltage between dual grid points. d , b , and $j\hat{A}$ are fluxes over grid or dual grid faces.

The allocation of the voltage and flux components on the dual grids is shown in Figure 1.2.

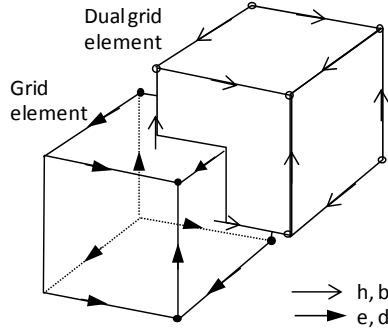


Figure 1.2 Locations of the voltage and flux components in the finite integration technique.

Due to the consistent transformation, the analytical properties of the fields are maintained resulting in corresponding discrete topological operators on the staggered grid duplet. The topology matrices C , \tilde{C} , $\hat{S}A$, and $\tilde{\hat{S}}\hat{A}$ correspond to the curl- and the div-operators. The tilde means that the operator is performed on the dual grid.

After discretization, the material property relations become:

$$d = M_e e \quad (1.38)$$

$$b = M_\mu h \quad (1.39)$$

$$j = M_k e + j_A \quad (1.40)$$

where M_e , M_μ , and M_k are matrices describing the material properties. The relations in (1.34) through (1.37) are exact on a given mesh; however, the material matrices contain the unavoidable approximations of any numerical procedure. In addition, these matrices have diagonal form.

Employing a so-called leapfrog scheme that samples values of e and h at times separated by a half time step, the MGEs can be rewritten as a set of two recursion formulas:

$$h^{i+1} = h^i - \Delta t M_\mu^{-1} C e^{i+1/2} \quad (1.41)$$

$$e^{i+3/2} = e^{i+1/2} + \Delta t M_e^{-1} (\tilde{C} e^{i+1} - j^{i+1}) \quad (1.42)$$

The recursion is stable if the time step inside an equidistant grid is restricted by the Courant criterion to

$$\Delta t \leq \frac{1}{c \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}}} \quad (1.43)$$

The calculation of each further time step only requires one matrix-vector multiplication. Thus, it has the advantage of being an explicit algorithm. FIT can be applied to different mesh types. On Cartesian grids, the time-domain FIT is equivalent to FDTD.

References

- [1] K. Yee, "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media," *IEEE Transactions on Antennas and Propagation*, Vol. 14, No. 5, May 1966, pp. 302–307.
- [2] A. Taflov and S. Hagness, (eds.), *Computational Electromagnetics: The Finite-Difference Time-Domain Method*, 3rd ed., Norwood, MA: Artech House, 2005.
- [3] Y. Hao and R. Mittra, *FDTD Modeling of Metamaterials: Theory and Applications*, Norwood, MA: Artech House, 2008.
- [4] W. Yu, et al., *Parallel Finite-Difference Time-Domain Method*, Norwood, MA: Artech House, 2006.
- [5] W. Yu, and R. Mittra, "A Conformal FDTD Software Package for Modeling of Antennas and Microstrip Circuit Components," *IEEE Antennas and Propagation Magazine*. Vol. 42, No. 5, October 2000, pp. 28–39.
- [6] W. Yu and R. Mittra, "A Conformal Finite-Difference Time-Domain Technique for Modeling Curved Dielectric Surfaces," *IEEE Microwave and Guided Wave Letters*, Vol. 11, No. 1, January 2001, pp. 25–27.
- [7] B. Wang, et al., "A Hybrid 2-D ADI-FDTD Subgridding Scheme for Modeling On-Chip Interconnects," *IEEE Transactions on Advanced Packaging*, Vol. 24, No. 11, November 2001, pp. 528–533.
- [8] W. Yu and R. Mittra, "A New Subgridding Method for Finite-Difference Time-Domain (FDTD) Algorithm," *Microwave and Optical Technology Letters*, Vol. 21, No. 5, June 1999, pp. 330–333.
- [9] M. Marrone, R. Mittra, and W. Yu, "A Novel Approach to Deriving a Stable Hybrid FDTD Algorithm Using the Cell Method," *Proc. IEEE AP-S URSI*, Columbus, OH, 2003.
- [10] T. Namiki, "A New FDTD Algorithm Based on Alternating-Direction Implicit Method," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 47, No. 10, October 1999, pp. 2003–2007.
- [11] F. Zheng, Z. Chen, and J. Zhang, "Toward the Development of a Three-Dimensional Unconditionally Stable Finite-Difference Time-Domain Method," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 48, No. 9, September 2000, pp. 1550–1558.
- [12] Y. Chao, Q. Cao, and R. Mittra, *Multiresolution Time-Domain Scheme for Electromagnetic Engineering*, New York: John Wiley & Sons, 2005.
- [13] Q. Liu, "The PSTD Algorithm: A Time-Domain Method Requiring Only Two Cells Per Wavelength," *Microwave and Optical Technology Letters*, Vol. 15, 1997, pp. 158–165.

-
- [14] C. Guiffaut and K. Mahdjoubi, "A Parallel FDTD Algorithm Using the MPI Library," *IEEE Antennas and Propagation Magazine*, Vol. 43, No. 2, April 2001, pp. 94–103.
 - [15] W. Yu, et al., "New Direction in Computational Electromagnetics Solving Large Problems Using the Parallel FDTD on the BlueGene/L Supercomputer Yielding Teraflop-Level Performance," *IEEE Antennas and Propagation Magazine*, Vol. 50, No. 23, April 2008, pp. 20–42.
 - [16] W. Yu, Y. Liu, T. Su, and R. Mittra, "A Robust Parallel Conformal Finite-Difference Time-Domain Processing Package Using MPI Library," *IEEE Antennas and Propagation Magazine*, Vol. 47, No. 3, June 2005, pp. 39–59.
 - [17] R. Courant, K. Friedrichs, and H. Lewy, "Über die partiellen Differenzengleichungen der mathematischen Physik," *Mathematische Annalen*, Vol. 100, No. 1, 1928, pp. 32–74.
 - [18] G. Mur, "Absorbing Boundary Conditions for the Finite-Difference Approximation of the Time-Domain Electromagnetic Field Equations," *IEEE Transactions on Electromagnetic Compatibility*, Vol. 23, No. 3, 1981, pp. 377–382.
 - [19] K. Mei and J. Fang, "Superabsorption - A Method to Improve Absorbing Boundary Conditions," *IEEE Transactions on Antennas and Propagation*, Vol. 40, No. 9, September 1992, pp. 1001–1010.
 - [20] Z. Liao, et al., "A Transmitting Boundary for Transient Wave Analyzes," *Scientia Sinica*, (Series A), Vol. 27, No. 10, October 1984, pp. 1062–1076.
 - [21] J. Berenger, "A Perfectly Matched Layer Medium for the Absorption of Electromagnetic Waves," *Journal of Computational Physics*, Vol. 114, October 1994, pp. 185–200.
 - [22] S. Gedney, "An Anisotropic Perfectly Matched Layer-Absorbing Medium for the Truncation of FDTD Lattices," *IEEE Transactions on Antennas and Propagation*, Vol. 44, No. 12, December 1996, pp. 1630–1639.
 - [23] W. Chew and W. Wood, "A 3-D Perfectly Matched Medium from Modified Maxwell's Equations with Stretched Coordinates," *Microwave and Optical Technology Letters*, Vol. 7, 1994, pp. 599–604.
 - [24] W. Chew, J. Jin, and E. Michielssen, "Complex Coordinate Stretching as a Generalized Absorbing Boundary Condition," *Microwave and Optical Technology Letters*, Vol. 15, No. 6, August 1997, pp. 363–369.
 - [25] J. Roden and S. Gedney, "Convolution PML (CPML): An Efficient FDTD Implementation of the CFS-PML for Arbitrary Medium," *Microwave and Optical Technology Letters*, Vol. 27, No. 5, 2000, pp. 334–339.
 - [26] R. F. Harrington, *Time-Harmonic Electromagnetic Fields*, New York: Wiley-IEEE Press, 2001.
 - [27] A. Peterson, and R. Mittra, *Computational Methods for Electromagnetics*, New York: Wiley-IEEE Press, 1997.
 - [28] B. Rahman, F. Fernandez, and J. Davies, "Review of Finite Element Methods for Microwave and Optical Waveguides," *Proceedings of the IEEE*, Vol. 79, No. 10, October 1991.
 - [29] J. M. Jin, *The Finite Element Method in Electromagnetics*, New York: John Wiley & Sons, 2002.
 - [30] M. Clemens and T. Weiland, "Discrete Electromagnetics with the Finite Integration Technique," *PIERS*, Vol. 32, 2001, pp. 65–87.

Chapter 2

FDTD Optimization and Acceleration

In this chapter, we introduce the FDTD acceleration technique based on the vector arithmetic logic unit (VALU) [1–4] that is built in the regular CPU. Unlike other hardware acceleration techniques, the VALU acceleration technique does not require any extra hardware device for the acceleration. Each core in the regular CPU includes one floating-point unit (FPU) and one VALU. A regular FDTD code in C or FORTRAN only uses the FPU or one element in the VALU. The FDTD code can be accelerated significantly if we can use the VALU for the FDTD simulation since the VALU can work on the four operations simultaneously.

The performance of a regular computer has been dramatically improved in the past several years [5–9]. Due to the use of 64-bit operating systems, today we can easily install 8-GB or 16-GB memory in a regular computer to solve the larger EM problems. However, the simulation speed for the practical problems becomes one of the major bottlenecks. The multiple core processor, computer cluster, and special GPU are designed for the simulation acceleration. In this chapter, we investigate the performance of the parallel FDTD method on the different hardware platforms such as the CPU with VALU acceleration, computer cluster, and GPU. Among the computational electromagnetic techniques, the FDTD method is parallel in nature since it requires only the field exchanging on the interface between the adjacent subdomains; therefore, the FDTD method has the higher parallel efficiency than other methods that require 3-D data exchanging. The parallel performance depends not only on the way we develop the FDTD code and physics model, but also on the hardware platform such as CPU type, network system, input/output (I/O) system, and the operating system as well. The parallel FDTD code is developed based on either open multiple processing (OpenMP) [10, 11] or message passing Interface (MPI) library [12, 13]. OpenMP is developed for the efficient use of multicore processors and the MPI library is developed to use the distributed resource. The single instruction and multiple data (SIMD) [14–19] combined with OpenMP and MPI are used for accelerating the FDTD code on both the GPU [20–23] and VALU.

Multicore processor and computer cluster have been successfully used for various large and complex electromagnetic problems for many years. Although a

large number of publications have been focused on the GPU acceleration, there are many challenging issues today for GPU to solve practical electromagnetic problems. In this chapter, we focus on a new acceleration technique that is based on the VALU built in the regular CPU. The readers who desire to learn the detailed computer cluster and GPU acceleration techniques can refer to the literature on those topics. Each core in the modern computer processor includes one FPU and one VALU. The VALU is operated by using the streaming SIMD extensions (SSE) in the Intel processor. Intel introduced the SSE in early 1999 with the Prescott revision of their Pentium 3 CPU. In April 2005, AMD introduced a subset of SSE3 in revision E of Athlon 64 CPUs. The regular FDTD code without using the VALU acceleration only uses FPU or one of four VALU components; however, the VALU in principle will accelerate the FDTD code four times since the VALU can handle four operations simultaneously. The performance of FDTD code with the VALU acceleration strongly depends on the FDTD structure including the update equations, boundary condition, material array structure, near-to-far-field transformation, dispersive medium types, periodic boundary condition, and conformal technique.

2.1 INTRODUCTION TO CPU ARCHITECTURE

Each core in the multicore processor has its own cache, FPU and VALU, as shown in Figure 2.1. Unlike FPU, VALU allows us to operate on four blocks of data at the same time. We use the VALU through the SIMD instruction set to accelerate the parallel FDTD method.

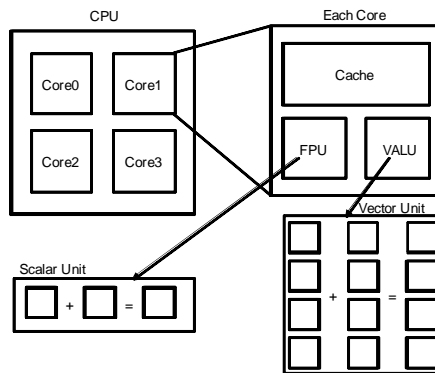


Figure 2.1 CPU architecture including the floating-point unit (FPU) and the vector arithmetic logic unit (VALU).

The Pentium 4 processor, and each core on a Pentium D CPU, could execute one SIMD instruction every two clock cycles. Intel streamlines the SSE execution unit, making it 128 bits wide throughout. In addition, Core 2 offers three SSE execution units. The net result is that 128-bit-wide SSE instructions can be

executed in a single clock cycle. This can substantially speed up the FDTD simulation that uses Intel's streaming SIMD (SSE/SSE2/SSE3) instruction set.

2.2 SSE INSTRUCTION SET

SSE is a newer SIMD extension to the Intel Pentium III and AMD AthlonXP microprocessors. SSE [4] adds a separate register space to the microprocessor. Because of this, SSE can only be used on operating systems that support it. All versions of Windows since Windows 98 support SSE, as do Linux kernels since 2.2.

SSE adds eight new 128-bit registers, divided into four 32-bit (single precision) floating-point values. These registers are called XMM0–XMM7. An additional control register, MXCSR, is also available to control and check the status of SSE instructions.

SSE gives us access to 70 new instructions that operate on these 128-bit registers, MMX registers, and sometimes even regular 32-bit registers.

Table 2.1 Basic SSE Operations

Pnemonic	Bit Location	Description
FZ	Bit 15	Flush to zero
R+	Bit 14	Round positive
R-	Bit 13	Round negative
RZ	Bits 13 and 14	Round to zero
RN	Bits 13 and 14 are 0	Round to nearest
PM	Bit 12	Precision mask
UM	Bit 11	Underflow mask
OM	Bit 10	Overflow mask
ZM	Bit 9	Divide by zero mask
DM	Bit 8	Denormal mask
IM	Bit 7	Invalid operation mask
DAZ	Bit 6	Denormals are zero
PE	Bit 5	Precision flag
UE	Bit 4	Underflow flag
OE	Bit 3	Overflow flag
ZE	Bit 2	Divide-by-zero flag
DE	Bit 1	Denormal flag
IE	Bit 0	Invalid operation flag

FZ mode causes all underflowing operations to simply go to zero. This saves some processing time, but loses precision.

The R+, R-, RN, and RZ rounding modes determine how the lowest bit is generated. Normally, RN is used.

PM, UM, MM, ZM, DM, and IM are masks that tell the processor to ignore the exceptions that happen, if they do. This keeps the program from having to deal with problems, but might cause invalid results.

DAZ tells the CPU to force all denormals to zero. A denormal is a number that is so small that FPU cannot renormalize it due to limited exponent ranges. They are just like normal numbers, but they take considerably longer to process. Note that not all processors support DAZ.

PE, UE, ME, ZE, DE, and IE are the exception flags that are set if they happen, and are not unmasked. Programs can check these to see if something interesting happened. These bits are “sticky,” which means that once they are set, they stay set forever until the program clears them. This means that the indicated exception could have happened several operations ago, but nobody bothered to clear it.

DAZ was not available in the first version of SSE. Since setting a reserved bit in MXCSR causes a general protection fault, we need to be able to check the availability of this feature without causing problems. To do this, one needs to set up a 512-byte area of memory to save the SSE state to, using `fxsave`, and then one needs to inspect bytes 28 through 31 for the `MXCSR_MASK` value. If bit 6 is set, DAZ is supported; otherwise, it is not.

SSE — OpCode List

<code>addps -</code>	Adds four single-precision (32-bit) floating-point values to four other single-precision floating-point values.
<code>addss -</code>	Adds the lowest single-precision values, top three remain unchanged.
<code>subps -</code>	Subtracts four single-precision floating-point values from four other single-precision floating-point values.
<code>subss -</code>	Subtracts the lowest single-precision values, and the top three remain unchanged.
<code>mulps -</code>	Multiplies four single-precision floating-point values with four other single-precision values.
<code>mulss -</code>	Multiplies the lowest single-precision values, and the top three remain unchanged.
<code>divps -</code>	Divides four single-precision floating-point values by four other single-precision floating-point values.
<code>divss -</code>	Divides the lowest single-precision values, and the top three remain unchanged.
<code>rcpps -</code>	Reciprocates (1/x) four single-precision floating-point values.
<code>rcpss -</code>	Reciprocates the lowest single-precision values, and the top three remain unchanged.
<code>sqrtps -</code>	Square root of four single-precision values.

<code>sqrts</code> -	Square root of the lowest value, and the top three remain unchanged.
<code>rsqrtps</code> -	Reciprocal square root of four single-precision floating-point values.
<code>rsqrtss</code> -	Reciprocal square root of the lowest single-precision value, and the top three remain unchanged.
<code>maxps</code> -	Returns maximum of two values in each of four single-precision values.
<code>maxss</code> -	Returns maximum of two values in the lowest single-precision value. And the top three remain unchanged.
<code>minps</code> -	Returns minimum of two values in each of four single-precision values.
<code>minss</code> -	Returns minimum of two values in the lowest single-precision value, top three remain unchanged.
<code>pavgb</code> -	Returns average of two values in each of eight bytes.
<code>pavgw</code> -	Returns average of two values in each of four words.
<code>psadbw</code> -	Returns sum of absolute differences of eight 8-bit values, which results in the bottom 16 bits.
<code>pextrw</code> -	Extracts one of four words.
<code>pinsrw</code> -	Inserts one of four words.
<code>pmaxsw</code> -	Returns maximum of two values in each of four signed word values.
<code>pmaxub</code> -	Returns maximum of two values in each of eight unsigned byte values.
<code>pminsw</code> -	Returns minimum of two values in each of four signed word values.
<code>pminub</code> -	Returns minimum of two values in each of eight unsigned byte values.
<code>pmovmskb</code> -	Builds mask byte from top bit of 8-byte values.
<code>pmulhuw</code> -	Multiplies four unsigned word values and stores the high 16-bit result.
<code>pshufw</code> -	Shuffles four word values. Takes two 128-bit values (source and dest) and an 8-bit immediate value, and then fills in each Dest 32-bit value from a source 32-bit value specified by the immediate. The immediate byte is broken into four 2-bit values.
Logic:	
<code>andnps</code> -	Logically ANDs four single-precision values with the logical inverse (NOT) of four other single-precision values.
<code>andps</code> -	Logically ANDs four single-precision values with four other single-precision values.
<code>orps</code> -	Logically ORs four single-precision values with four other single-precision values.
<code>xorps</code> -	Logically XORs four single-precision values with four other single-precision values.
Compare:	
<code>cmpxxps</code> -	Compares four single-precision values.
<code>cmpxxss</code> -	Compares lowest two single-precision values.

-
- comiss - Compares the lowest two single-precision values and stores result in EFLAGS.
- ucomiss - Compares lowest two single-precision values and stores result in EFLAGS. (QNaNs do not throw exceptions with ucomiss, unlike comiss.)

Compare Codes (the xx parts above):

- eq - Equal to.
- lt - Less than.
- le - Less than or equal to.
- ne - Not equal.
- nlt - Not less than.
- nle - Not less than or equal to.
- ord - Ordered.
- unord - Unordered.

Conversion:

- cvtpi2ps - Converts two 32-bit integers to 32-bit floating-point values. Top two values remain unchanged.
- cvtps2pi - Converts two 32-bit floating-point values to 32-bit integers.
- cvtsi2ss - Converts a 32-bit integer to 32-bit floating-point value. Top three values remain unchanged.
- cvtss2si - Converts a 32-bit floating-point value to 32-bit integer.
- cvttps2pi - Converts two 32-bit floating-point values to 32-bit integers using truncation.
- cvtss2si - Converts a 32-bit floating-point value to 32-bit integer using truncation.

State:

- fxrstor - Restores FP and SSE State.
- fxsave - Stores FP and SSE State.
- ldmxcsr - Loads the mxcsr register.
- stmxcsr - Stores the mxcsr register.

Load/Store:

- movaps - Moves a 128-bit value.
- movhlps - Moves high half to a low half.
- movlhps - Moves low half to upper halves.
- movhps - Moves a 64-bit value into top half of an xmm register.
- movlps - Moves a 64-bit value into bottom half of an xmm register.
- movmskps - Moves top bits of single-precision values into bottom four bits of a 32-bit register.
- movss - Moves a single-precision FP value from the source operand to the destination operand.
- movups - Moves a 128-bit value. Address can be unaligned.

- maskmovq - Moves a 64-bit value according to a mask.
- movntps - Moves a 128-bit value directly to memory, skipping the cache. (NT stands for “nontemporal.”)
- movntq - Moves a 64-bit value directly to memory, skipping the cache.

Shuffling:

- shufps - Shuffles four single-precision values. Complex.
- unpckhps - Unpacks single-precision values from high halves.
- unpcklps - Unpacks single-precision values from low halves.

Cache Control:

- prefetchT0 - Fetches a cache line of data into all levels of cache.
- prefetchT1 - Fetches a cache line of data into all but the highest levels of cache.
- prefetchT2 - Fetches a cache line of data into all but the two highest levels of cache.
- prefetchNTA - Fetches data into only the highest level of cache, not the lower levels.
- sfence - Guarantees that all memory writes issued before the sfence instruction are completed before any writes after the sfence instruction.

With prefetching, it is okay to access an invalid memory location (i.e., off the end of an array); however, generating the address must not fault.

2.3 CACHE OPTIMIZATION

Cache optimization plays an important role in the FDTD code optimization, which will decide how we can efficiently use the CPU calculation units. In the cache, L1 cache is faster than any other cache and memory, as shown in Figure 2.2. It is the large L1, L2, and L3 caches that make the CPU with several cores faster than a GPU in the general FDTD applications.

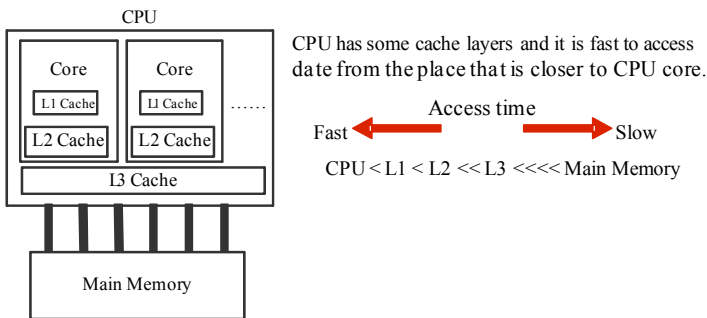


Figure 2.2 Memory and cache priority levels in the CPU during the simulation. The cache closer to the calculating unit is faster.

2.4 TASK PARALLELIZATION AND BUNDLING

Using OpenMP, the FDTD simulation will be split into the several segments based on the available number of cores that are assigned to each core. However, the processes will be switched among the cores, as shown in Figure 2.3.

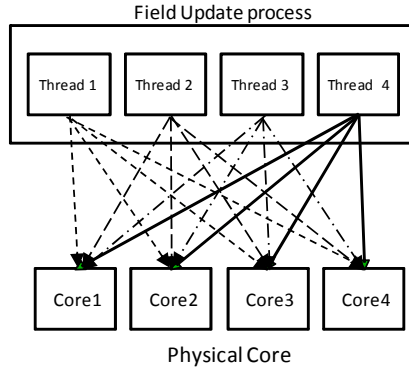


Figure 2.3 OpenMP working diagram on the multicore processor. The jobs are switched between the cores, namely, the operating system needs to allocate the job distribution dynamically.

The working way shown in Figure 2.3 has a very low efficiency because the jobs are switched from one core to another one. We can allocate each thread to a specified physical core, respectively, with processor affinity. The efficiency is much higher than the way shown in Figure 2.4 by improving the cache hit ratio.

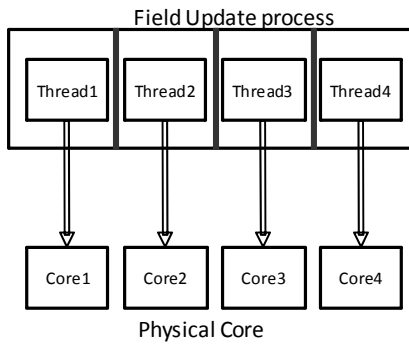


Figure 2.4 After the job bundling treatment, the jobs are bundled on the specified cores during the simulation.

2.5 PREFETCH

In addition to the SIMD instructions, both Intel and AMD also provide the instructions that let us specify how we want the data to interact with the system cache. The prefetching allows us to preload a specific address into a special

nontemporal area of the L_1 cache, which is guaranteed not to displace any data that was loaded into the cache due to more legitimate reasons. Essentially, we use prefetching to indicate that we want the data at the specified address to be in the cache, so we can read to (or write from) it exactly once, at which point it will not be used again. For example, we load the magnetic fields into the cache before calculating the electric fields at the first step, as shown in Figure 2.5. When the CPU calculates the electric fields we load the magnetic fields for the calculation of the next electric fields.

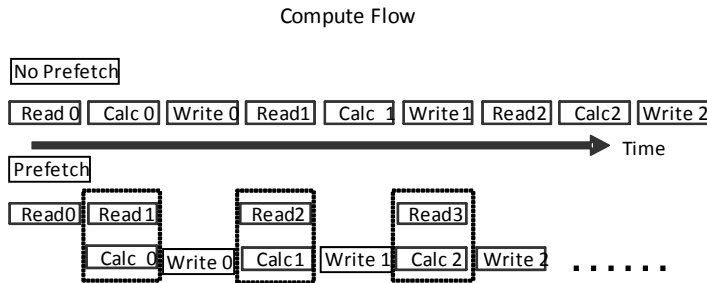


Figure 2.5 Prefetching reduces the executing time by preparing the calculation data during the simulation to reduce the memory transfer time since the memory transfer time is dominating in the FDTD simulation.

2.6 READING OR WRITING COMBINATION

Reading or writing two objects from memory one right after the other is faster than reading or writing one object, then running a lot of code, then reading or writing the other object. With a bit of reorganization, we can apply a similar method to the VALU multiplier to accelerate the simulation, as shown in Figure 2.6.

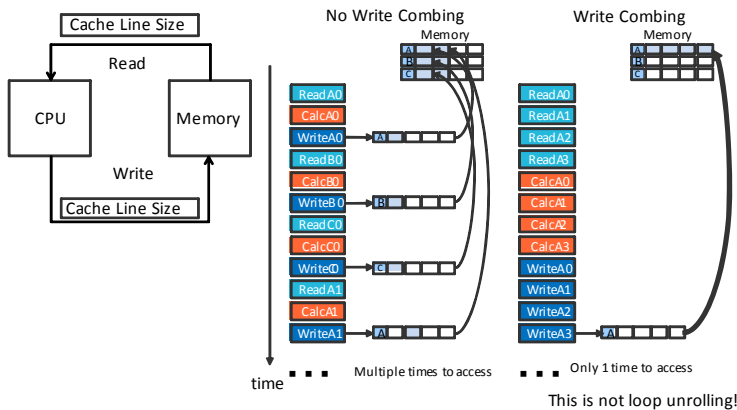


Figure 2.6 Control the reading or writing procedure and reduce the number of the reading and writing operations.

Writing data to memory during the simulation is a common operation. If each communication setup is only for one data writing and the communication setup is only one time for each data, the efficiency will be extremely slow. For example, if we are doing things like transforming entire data into memory, this will be much faster than if we write the data one by one.

CPU accesses to memory by cache line unit, especially to write memory. CPU tries to write data to cache and then the memory control will write it at an appropriate time.

2.7 MATERIAL LOOP-UP TABLE

The material distribution for each component is expressed as a sparse 3-D array. This representation has two major disadvantages: (1) the memory usage is expensive, and (2) the cache hit rate is relatively low. An efficient way is to express the material distribution as a table plus an integer 3-D array, as shown in Figure 2.7.

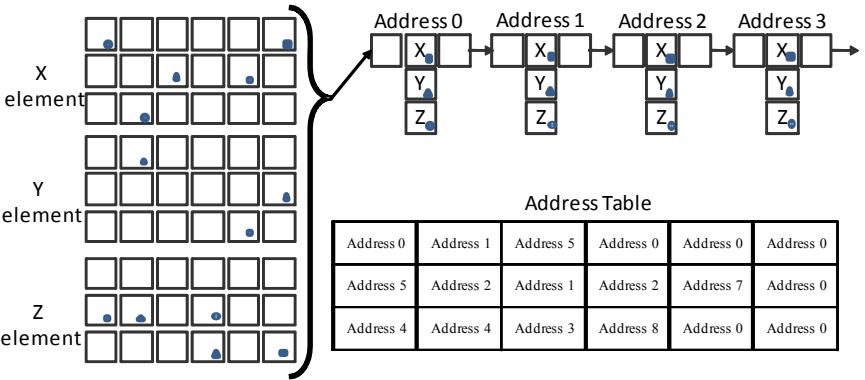


Figure 2.7 Create a material chain table and map the material type into the table to improve the cache hit rate.

2.8 NUMA OPTIMIZATION

The memory priority in the symmetric multiprocessing (SMP) system is in the same level to all the processors in the system, as shown in Figure 2.8. The OpenMP and MPI assign each subdomain to the cores in the system and the performance will be desired but the high-performance system is very costly.

In contrast to the SMP system, the nonuniform memory access or nonuniform memory architecture (NUMA) [24–30], as shown in Figure 2.9, is a computer memory design used in multiple processors, where the memory access time depends on the memory location relative to a processor. Under the NUMA architecture, a processor can access its own local memory faster than nonlocal memory, that is, memory local to another processor or memory shared between

processors. To achieve better software performance, the multiple processor workstations today use the NUMA architecture.

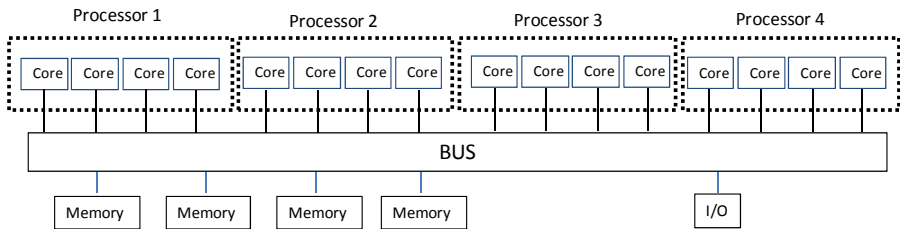


Figure 2.8 In an SMP system, each processor has the same priority to access to all the memory installed in the system.

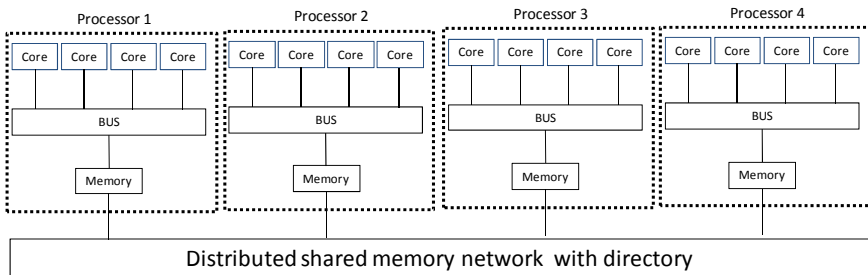


Figure 2.9 Each processor has its own memory in the nonuniform memory access system, and the local memory is much faster than the memory that is accessed through the network.

In the FDTD simulation, we need to place the data associated with one processor in its own memory to achieve the best simulation performance. The special memory allocation technique is required for this type of memory optimization control in the multiple processor computer.

Intel and AMD use the different NUMA architecture. Intel QuickPath technology, as shown in Figure 2.10, is a platform architecture that provides high speed (up to 25.6 GB/s), point-to-point connections between processors, and between processors and the I/O hub. Each processor has its own dedicated memory that it accesses directly though an integrated memory controller. In cases where a processor needs to access the dedicated memory of another processor, it can do so through a high-speed Intel QuickPath interconnect (Intel QPI) that links all the processors.

Unlike Intel QuickPath technology, AMD uses the 16-bit Hyper Transport 3 technology (H3) links, up to 6.4 GT/s per link, as shown in Figure 2.11.

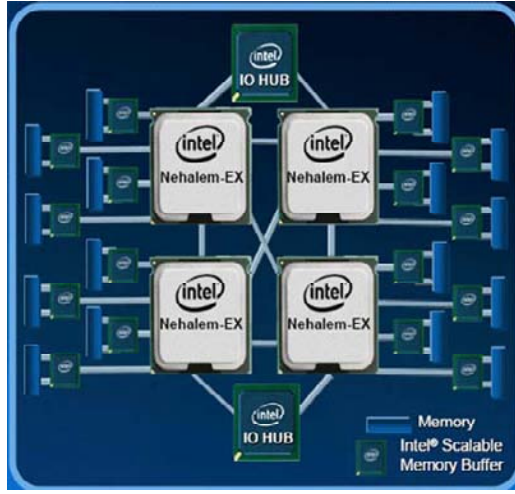


Figure 2.10 Intel QuickPath Architecture in which each processor has its own memory. (Image from <http://www.intel.com>.)

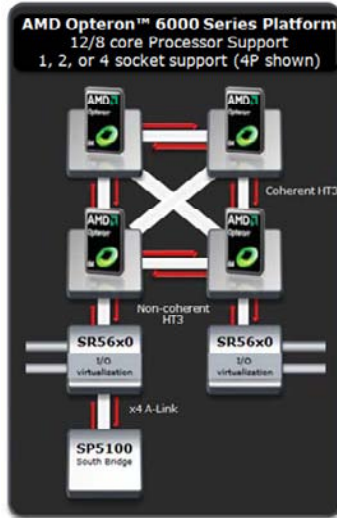


Figure 2.11 AMD 16-bit HyperTransport 3 architecture in which each processor has its own memory. (Image from <http://www.amd.com>.)

2.9 IMPLEMENTATION OF VALU FDTD METHOD

In this subsection, we use the electric field component update as an example to explain how the FDTD code is accelerated by using VALU. For example, the electric field E_z can be expressed as follows [8, 9]:

$$E_z^{n+1}(i, j, k + 1/2) = E_z_Coeff \cdot E_z^n(i, j, k + 1/2) + H_y_Coeff \cdot [H_y^{n+1/2}(i + 1/2, j, k + 1/2) - H_y^{n+1/2}(i - 1/2, j, k + 1/2)] - H_x_Coeff \cdot [H_x^{n+1/2}(i, j + 1/2, k + 1/2) - H_x^{n+1/2}(i, j - 1/2, k + 1/2)] \quad (2.1)$$

where E_z_Coeff , H_x_Coeff , and H_y_Coeff are coefficients in the field update equations [4]. Using the following procedure VALU can be used to compute four E_z at the same time to accelerate the simulation, which is described as follows:

- Load the coefficient of the magnetic field H_y into the SSE registers.
- Load the coefficient of the magnetic field H_x into the SSE registers.
- Convert the float pointer to the SSE 128-bit pointer.
- Calculate the difference of magnetic fields.
- Multiply the difference of magnetic fields and their coefficients.
- Calculate the contribution of magnetic fields to the electric fields.
- Multiply the previous electric fields and their coefficients.
- Calculate the electric fields and write them to memory.

```
for( i = 0; i <= nx; i ++ ) {
    vHi_Coeff = _mm_loadl_ps( &Hi_Coeff ); //load single float value to vector
    for( j = 0; j <= ny; j ++ ) {
        vHj_Coeff = _mm_loadl_ps( &Hj_Coeff ); //load single float value to
                                                //vector
        vEz = ( __m128 * ) Ez[i][j];
        vHy = ( __m128 * ) Hy[i][j];
        vHx = ( __m128 * ) Hx[i][j];
        vHy_minus = ( __m128 * ) Hy[i-1][j];
        vHx_minus = ( __m128 * ) Hx[i][j-1];
        for( k = 0, vk = 0; k < nz; k += 4, vk ++ ) {
            vEk_Coeff = _mm_loadl_ps( &Ek_Coeff );
            xmm0 = _mm_sub_ps( vHx[vk], vHx_minus[vk] );
            xmm0 = _mm_mul_ps( vHj_Coeff, xmm0 );
            xmm1 = _mm_sub_ps( vHy[vk], vHy_minus[vk] );
            xmm1 = _mm_mul_ps( vHi_Coeff, xmm1 );
            xmm0 = _mm_sub_ps( xmm1, xmm0 );
            xmm1 = _mm_mul_ps( vEk[vk], vEk_Coeff );
            vEk[vk] = _mm_add_ps( xmm1, xmm0 );
        } } }
```

Using the VALU and this procedure can significantly accelerate the multiplication operation compared to the traditional FDTD code. In any case, the data continuity in the FDTD code and memory bandwidth are the most important factors for the VALU acceleration. To evaluate the performance of the FDTD code, we define the performance as follows:

$$Performance (Mcells/s) = \frac{(N_x \times N_y \times N_z) \times Number_of_timesteps}{Simulation_time (second)} \quad (2.2)$$

For example, the performance of a regular parallel conformal FDTD code on an Intel Core i7-965 3.2GHz is 87 Mcells/sec, as shown in Figure 2.12. If we apply the optimized synchronization technique on the parallel FDTD code, the performance increases to 124 Mcells/sec. When we implement the VALU acceleration technique on the parallel conformal FDTD code, the performance increases to 191 Mcells/sec. Besides the field update module, further optimization on the material list table, cache usage, dispersive medium, and near-to-far-field transformation will also significantly improve the FDTD performance, as shown in Figure 2.12.

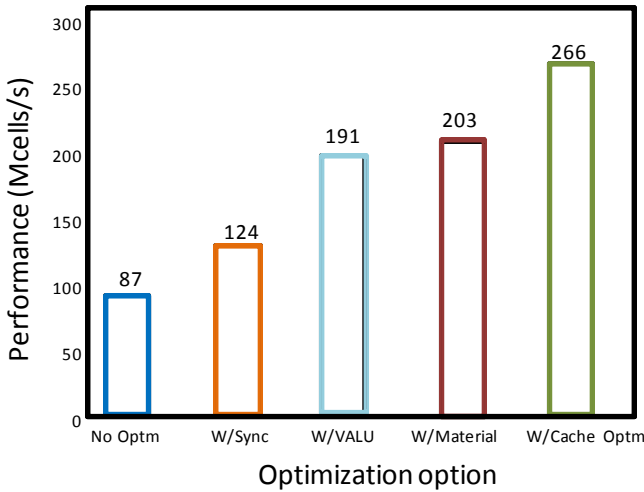


Figure 2.12 Performance improvement of the parallel FDTD code using the optimized VALU acceleration technique.

Now we use the parallel conformal FDTD code to simulate an ideal test case that is a hollow box with the simplest excitation and output; its domain is truncated by using the perfect electric conductor (PEC). We ran the problem with the different sizes using the regular FDTD code (Intel Core i7-965), and the FDTD code with the VALU (Intel Core i7-965 with four VALUs, the total memory bandwidth is 32 GB). The simulation summary is plotted in Figure 2.13. It is observed from Figure 2.13 that the peak performance of the VALU FDTD code is four times faster than the FDTD code on a regular CPU for the ideal test case.

In contrast to the Intel processor, we need to make the slight changes on the parallel FDTD code to fit the AMD architecture. We investigate the performance of the parallel conformal FDTD code on the AMD Opteron 6128 processor and Intel Xeon X5550 processor. For the same ideal test case, the problem size increases from 8 Mcells to 125 Mcells and the performance of parallel conformal

FDTD code on NVIDIA Tesla C1060 (30 SMs (Streaming Multiprocessor) /240 cores and the memory bandwidth is 109 GB/s), two AMD Opteron 6128 2.0 GHz processors (total memory bandwidth is 85 GB/s) with total 16 VALUs, and two Intel Xeon X5550 2.7-GHz processors (the total memory bandwidth is 64 GB/s) with total 8 VALUs. It is observed from Figure 2.14 that the performance of parallel conformal FDTD code on two AMD Opteron 6128 CPUs or on two Intel Xeon X5550 CPUs is just slightly slower than NVIDIA Tesla C1060. Now we use two computers (each of them includes two Intel Xeon X5550 CPUs, and the total memory bandwidth is 128 GB and the total number of VALUs is 16) to run the parallel FDTD code with the VALU acceleration, and its maximum performance can reach more than 840 Mcells/second, which is much faster than the GPU Tesla C1060 (www.nvidia.com, 240 cores) for all the sizes of problems, as shown in Figure 2.14. Two computers are connected through the Ethernet network with the bandwidth of 10 Gbps.

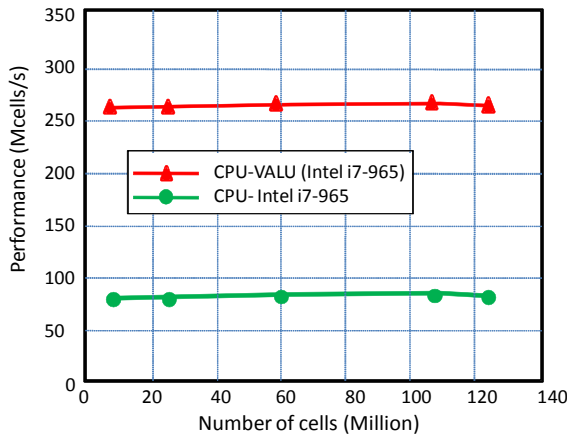


Figure 2.13 FDTD performance on the Intel CPU and Intel CPU plus VALU for the ideal test case.

Due to the discontinuous data in the PML boundary, the PML boundary [31, 32] will reduce 50% performance of the GPU code. However, the PML boundary only reduces 20% performance of the FDTD code on the VALU due to the large available cache in the CPU. For the same reason, the FDTD code on the VALU is much faster than that on the GPU for handling dispersive medium and conformal cells.

For the practical problems, the project model handling, material preprocessing, mesh generation, far-field pattern calculation, and large range of surface current outputs are the major bottleneck of GPU code; hence, they are extremely time-consuming for the GPU code. However, all those options match very well to the CPU architecture and especially for the CPU cluster with the domain decomposition techniques.

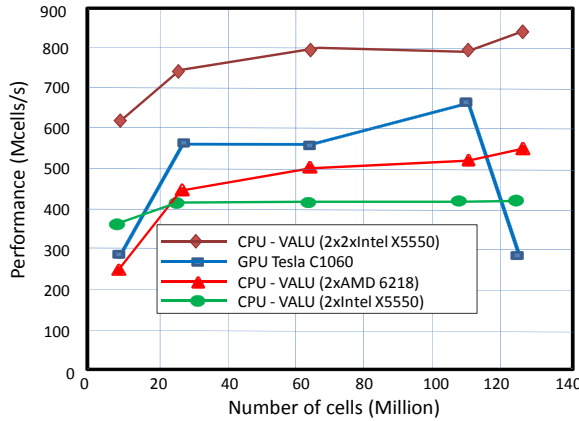


Figure 2.14 FDTD performance on the NVIDIA GPU, AMD CPU, and Intel CPU for the ideal test case.

Before ending this section, we investigate the performance of parallel FDTD code on an intermediate cluster that includes 18 nodes (36 CPUs with 144 cores and 144 VALUs, Intel Xeon X5550 2.7-GHz processor). The 10-Gbps Ethernet is used to connect the computer nodes. The performance versus the problem size and number of CPUs is plotted in Figure 2.15.

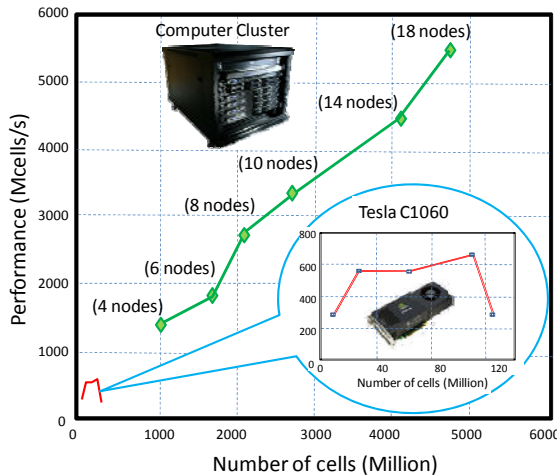


Figure 2.15 Performance of the parallel FDTD method on the computer cluster with 144 VALUs and the comparison with NVIDIA Tesla C1060 GPU.

References

- [1] H. Enoch, *Digital Logic and Microprocessor Design with VHDL*, Thomson, 2006, <http://faculty.lasierra.edu/~ehwang/digitaldesign>.
- [2] W. Stallings, *Computer Organization & Architecture: Designing for Performance*, 7th ed., Upper Saddle River, NJ: Pearson Prentice Hall, 2006, <http://williamstallings.com/COA/COA7e.html>.
- [3] Intel, Intel® Architecture Optimization, *Reference Manual*, 1999 Intel Corporation.
- [4] <http://softpixel.com/~cwright/programming/simd/sse.php>.
- [5] http://en.wikipedia.org/wiki/History_of_computing_hardware.
- [6] <http://www.intel.com>.
- [7] <http://www.amd.com>.
- [8] W. Yu, et al., "New Development of Parallel Conformal FDTD Method in Computational Electromagnetic Engineering," *IEEE Antennas and Propagation Magazine*, Vol. 53, No. 3, 2011.
- [9] W. Yu, et al., *Parallel Finite-Difference Time-Domain Method*, Norwood, MA: Artech House, 2006.
- [10] http://www.nic.uoregon.edu/iwomp2005/iwomp2005_tutorial_openmp_rvdp.pdf.
- [11] <http://cobweb.ecn.purdue.edu/~eigenman/ECE563/Handouts/ECE563-OpenMP.pdf>.
- [12] <http://www.slac.stanford.edu/comp/unix/farm/mpi.html>.
- [13] <http://www.mcs.anl.gov/research/projects/mpi/tutorial/gropp/talk.html>.
- [14] http://en.wikipedia.org/wiki/Streaming_SIMD_Extensions.
- [15] "AMD Plots Single Thread Boost with x86 Extensions," http://www.theregister.co.uk/2007/08/30/amd_sse5/.
- [16] <http://developer.amd.com/sse5.jsp>.
- [17] "AMD64 Architecture Programmer's Manual Volume 6: 128-Bit and 256-Bit XOP, FMA4 and CVT16 Instructions," http://support.amd.com/us/Processor_TechDocs/43479.pdf.
- [18] "Intel Offers Peek at Nehalem and Larrabee," <http://www.extremetech.com/article2/0,1697,2276803,00.asp>.
- [19] "Striking a Balance," <http://forums.amd.com/devblog/blogpost.cfm?threadid=112934&catid=208>.
- [20] http://www.nvidia.com/object/product_tesla_c1060_us.html.
- [21] *NVIDIA CUDA: Compute Device Architecture Programming Guide*, v2.3, NVIDIA Corporation, Santa Clara, CA, 2009.
- [22] <http://developer.amd.com/documentation/articles/pages/opencl-and-the-ati-stream-v2.0-beta.aspx>.
- [23] A. Elsherbeni and V. Demir, *The Finite-Difference Time-Domain Method for Electromagnetics: With MATLAB Simulations*, Raleigh, NC: Scitech Publishing, 2008.

- [24] <http://developer.amd.com/pages/1162007106.aspx>.
- [25] http://en.wikipedia.org/wiki/Non-Uniform_Memory_Access.
- [26] <http://lse.sourceforge.net/numa/faq/>.
- [27] D. Ott, "Optimizing Software Applications for NUMA," <http://software.intel.com/en-us/articles/optimizing-software-applications-for-numa/>.
- [28] Drepper, Ulrich, "What Every Programmer Should Know About Memory," November 2007, <http://www.akkadia.org/drepper/cpumemory.pdf>.
- [29] Intel® 64 and IA-32 Architectures Optimization Reference Manual. Section 8.8, "Affinities and Managing Shared Platform Resources," March 2009.
- [30] C. Lameter, "Local and Remote Memory: Memory in a Linux/NUMA System," <http://www.kernel.org/pub/linux/kernel/people/christoph/pmig/numamemory.pdf>.
- [31] J. Berenger, "A Perfectly Matched Layer Medium for the Absorption of Electromagnetic Waves," *Journal of Computational Physics*, Vol. 114, October 1994, pp. 185–200.
- [32] S. Gedney, "An Anisotropic Perfectly Matched Layer-Absorbing Medium for the Truncation of FDTD Lattices," *IEEE Transactions on Antennas and Propagation*, Vol. 44, No. 12, December 1996, pp. 1630–1639.

Chapter 3

Parallel FDTD Method and Systems

In this chapter, we introduce the parallel FDTD method and parallel processing system including the network system. The brief introduction to the parallel systems helps the readers understand the electromagnetic simulation performance.

3.1 PARALLEL FDTD METHOD

In the parallel processing technique, the original problem is divided into small pieces, and the FDTD code assigns each piece to each computer. Each computer in the cluster only needs to simulate the specified subdomain, as shown in Figure 3.1.

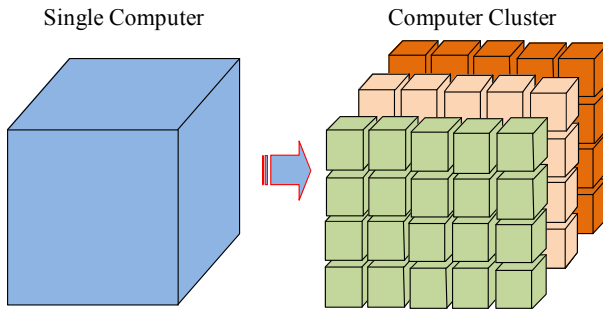


Figure 3.1 Basic idea of parallel processing FDTD simulation on a computer cluster, in which the original job is split into small pieces that are then assigned to each core/processor/node.

A complete FDTD simulation is carried out in two steps, namely, project preprocessing and project simulation. In the first step, the FDTD code generates the material distribution based on the geometry information and the specified mesh. In this procedure, each core in a cluster does not have to wait for others, since it does not need to borrow any information from its neighbors. Due to the complexity of the need to mesh and the fact that the object structure in each subdomain may be different, the processing time needed in each subdomain may vary for different cores. In order to speed up the preprocessing procedure, we do

not have to let the number of subdomains equal to the number of cores and the cores carrying on the simple subdomains can work on more subdomains without the need to wait for the other cores, as illustrated in Figure 3.2. In contrast to the project preprocessing, the FDTD update processing in each subdomain must be synchronized at each time step. Therefore, we will not allow the core to switch from one subdomain to another during the simulation.

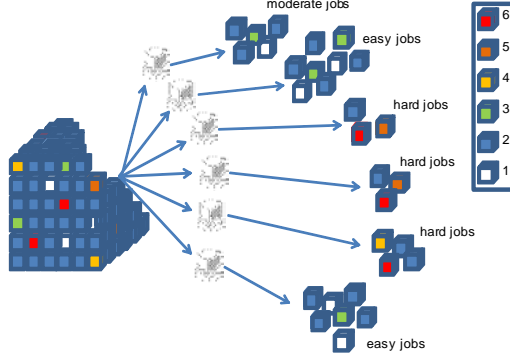


Figure 3.2 Break the original job into small pieces, each core in the cluster will handle one piece at one time. Since all the related information is localized, the project preprocessing including material distribution and mesh generation is more efficient.

The field update in the FDTD method only requires the information around it, namely, it is one of the highly parallelized methods. Let us start from one component of Maxwell's equations [1, 2]:

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon_z} \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma_z E_z \right) \quad (3.1)$$

Using the central difference scheme, we can get the following discretization formulation for (3.1):

$$E_z^{n+1}(i, j, k + 1/2) = \frac{\varepsilon_z - 0.5\sigma_z\Delta t}{\varepsilon_z + 0.5\sigma_z\Delta t} E_z^n(i, j, k + 1/2) + \frac{1}{\varepsilon_z + 0.5\sigma_z\Delta t} \left[\frac{H_y^{n+1/2}(i + 1/2, j, k + 1/2) - H_y^{n+1/2}(i - 1/2, j, k + 1/2)}{0.5[\Delta x(i) + \Delta x(i - 1)]} - \frac{H_x^{n+1/2}(i, j + 1/2, k + 1/2) - H_x^{n+1/2}(i, j - 1/2, k + 1/2)}{0.5[\Delta y(j) + \Delta y(j - 1)]} \right] \quad (3.2)$$

For a serial code, the field solution in the entire FDTD domain can be solved by forcing the fields to satisfy the proper condition on the domain boundary.

However, for the parallel code, the fields on the boundary of each subdomain are unknown, but they are calculated by borrowing some information from the adjacent subdomains. Therefore, we need the high-performance network to transform the information from one computer to another one at each time step. In contrast to other electromagnetic simulation techniques, the parallel FDTD method only requires passing the fields on the interface between the adjacent computers. For example, considering (3.2) in Figure 3.3, the electric field E_z is located on the interface between the processors 1 and 2. Update of this electric field needs two magnetic fields H_{y1} , namely, $H_y^{n+1/2}(i+1/2, j, k+1/2)$, and H_{y2} , namely, $H_y^{n+1/2}(i-1/2, j, k+1/2)$, that are located in the processors 1 and 2, respectively.

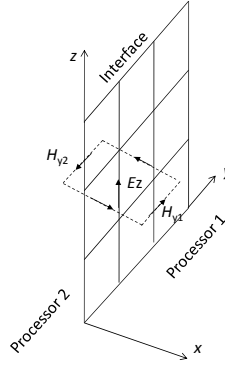


Figure 3.3 Distribution of the electric and magnetic fields near the interface between two processors.

We rewrite (3.2) in the processors 1 and 2, and the electric field on the interface can be expressed as:

$$E_z^{n+1, processor1}(i, j, k+1/2) = \frac{\varepsilon_z - 0.5\sigma_z \Delta t}{\varepsilon_z + 0.5\sigma_z \Delta t} E_z^n(i, j, k+1/2) + \frac{1}{\varepsilon_z + 0.5\sigma_z \Delta t} \left[\frac{H_y^{n+1/2}(i+1/2, j, k+1/2) - H_{y1}^{n+1/2, processor2}}{0.5[\Delta x(i) + \Delta x(i-1)]} - \frac{H_x^{n+1/2}(i, j+1/2, k+1/2) - H_x^{n+1/2}(i, j-1/2, k+1/2)}{0.5[\Delta y(j) + \Delta y(j-1)]} \right] \quad (3.3)$$

$$E_z^{n+1, processor2}(i, j, k+1/2) = \frac{\varepsilon_z - 0.5\sigma_z \Delta t}{\varepsilon_z + 0.5\sigma_z \Delta t} E_z^n(i, j, k+1/2) + \frac{1}{\varepsilon_z + 0.5\sigma_z \Delta t} \left[\frac{H_y^{n+1/2, processor1} - H_y^{n+1/2}(i-1/2, j, k+1/2)}{0.5[\Delta x(i) + \Delta x(i-1)]} - \frac{H_x^{n+1/2}(i, j+1/2, k+1/2) - H_x^{n+1/2}(i, j-1/2, k+1/2)}{0.5[\Delta y(j) + \Delta y(j-1)]} \right] \quad (3.4)$$

The magnetic fields $H_{y1}^{n+1/2,processor\ 2}$ in the processor 2 and $H_y^{n+1/2,processor\ 1}$ in the processor 1 are exchanged through the high performance network at each time step.

3.2 OPENMP FOR MULTICORE PROCESSORS

All processing units are now following a many-core architecture trend. A major challenge to address is the fact that parallel hardware requires parallel programming to take advantage of multiple available resources. OpenMP [3] is an implementation of multithreading, in which the master thread splits a specified number of slave threads and runs the threads concurrently on different processors. During simulation, the operating system will assign the compute resource to the FDTD code depending on the processor status. For example, when you run a parallel FDTD code on a quad core processor, if all four cores are available, the FDTD code will use all cores; otherwise, the FDTD code will share the compute cores with other application codes.

The parallel processing is carried out on a multiple core processor via the use of OpenMP algorithm. The cores inside one processor share the same host name; hence, we do not need to divide the domain into a number of subdomains or to assign the job to an individual core for the simulation.

Like the multiple core processor, the multiple processor computer also uses the OpenMP to speed up the simulation (see Figure 3.4). Today, we can easily place four processors in one computer. AMD company has released the 12-core processor, namely, one 4-CPU computer can include 48 cores and 48 VALUs. Intel has released its 8-core processor, which has a similar performance to the AMD processor. The Intel processor has a higher clock frequency than the AMD; it can process more operations in the same cycle than the AMD and it optimizes the cache hit rate better than the AMD. However, since the AMD has a greater number of VALUs in a similar-level CPU than the Intel, the VALU-based FDTD code on the AMD is more cost-efficient than the Intel.



Figure 3.4 An EM simulation workstation with multiple cores or multiple processors. Today, we can easily place four CPUs with 48 cores and 48 VALUs inside one computer.

Though OpenMP has slightly lower performance than MPI on the same platform, the FDTD code based on OpenMP is much simpler than the MPI.

3.3 MPI TECHNIQUE

Cluster computing, such as that possible with Beowulf class clusters, offers great potential for increasing the amount of computing power and communication resources available to large-scale applications. It is likely that the combined computational power of a cluster of powerful PCs connected to a high-speed network may exceed a stand-alone high-performance supercomputer.

The MPI library [4] is one of the most widely used parallel processing standards, and it is a library description rather than a new programming language, and its functions can be called from a C, C++, or Fortran program. Prior to the development of the MPI between the 1980s and early 1990s, developers of parallel codes had to choose between portability, efficiency, price, and functionality. The situation improved dramatically, however, when in April 1992, a meeting was held at the American Parallel Research Center, where the basic contents of the MPI library were approved. Next, in a meeting of the MPI working group held in Minneapolis in November 1992, the Oak Ridge National Laboratory submitted a report on the parallel MPI for the very first time. This conference also created an MPI forum that included 175 scientists and engineers from 40 research institutes and universities. The first version of the MPI library was introduced in a supercomputer conference in 1993 and the freely downloadable versions of this library became available from the Web site in May 1994.

A PC-based cluster (see Figure 3.5) is usually connected through a gigabit Ethernet switch. For a two-node cluster, one can directly connect two PCs together through a regular network cable, and the performance of the cluster would be good if the number of nodes is less than 4. There are no master or slave nodes in a PC cluster, and one can submit a job from any node in the cluster. The project input parameters can either be read from each node or from the node that is used for the job submission. This type of cluster is much cheaper than the high-performance cluster and each node in the cluster can be used either as an independent PC or as a compute node.

One of the major advantages of the PC cluster is that the computers in the cluster can be used as a compute node for the parallel computing, and each computer in the cluster can also be used as a regular computer if necessary.

The PC cluster connected by using the regular Gigabit Ethernet can usually have four computers at a reasonable parallel efficiency. When the number of computer nodes is more than four, we should use the faster network such as 10 Gigabit Ethernet, InfiniBand network, or Myrinet network to achieve the better performance, as shown in Figure 3.6. Although the bandwidth of the network system may be very important, we care more about latency than the bandwidth because the latency is important in the implementation of the parallel FDTD

algorithm. Since only the fields on the interface will be exchanged between the adjacent neighbors in the parallel FDTD method, the burden on the network is relatively light compared to the methods that require the 3-D data exchanging between the adjacent neighbors. The popular cluster management software (PBS and SGE) [5, 6] requires an extra node as the master node for the cluster management, and this node does not participate in the simulation. The master node is usually installed on a fast and large hard drive to improve the system I/O. Its performance is very good even if the number of nodes is more than hundreds or thousands.



Figure 3.5 A four-port network switch can connect four regular PCs together to form a typical PC cluster. The network cost can be ignored compared to the computer price.

The main features of the high-performance cluster include the high-performance servers or workstations with ECC memory and fast network system.



Figure 3.6 A cluster with the high-performance servers installed with the ECC memory allows us extremely stably to run the projects for a long time [7].

We can also connect the lab or office computers through a high-performance network system to form a powerful cluster, as shown in Figure 3.7. This type of cluster can easily have over 20 computers with more than 100-gigabit memory.



Figure 3.7 High-performance network system allows us to connect the lab/office PCs to form intermediate-sized cluster.

3.4 NETWORK CARD, SWITCH, AND CABLE

Running large-scale parallel applications on a cluster imposes heavy demands on the communication network. Therefore, in the early 1970s, one of the design goals for cluster computing was to limit the amount of communication between hosts. However, due to the features of some applications, a certain degree of communication between hosts may be required. As a result, the performance bottleneck of the network severely limited the potential of cluster computing. Recent developments in the high-speed networks such as 10-Gb Ethernet, InfiniBand, and Myrinet, change the situation somewhat. These high-speed networks offer the minimum 10,000 megabits per second (Mbps). However, the maximum achievable bandwidth at the application level is still far away from the theoretical peak bandwidth. This major roadblock to achieving high-speed cluster communication is caused by the time required for the interaction between software and hardware components.

Compared to the performance improvement on the CPU and memory, the progress on the network system including the network card, network cable, and network switch is not as fast as the CPU and memory. The bandwidth is an important factor for the high-performance computing; however, it is not so important for parallel FDTD software since the transform data is only on the interface of the adjacent subdomains. The most important factor in the parallel FDTD method is the latency that decides the parallel performance of the parallel FDTD method. Latency is the time taken for a sent packet of data to be received at the other end. It includes the time to encode the packet for transmission and transmit it, the time for that data to traverse the network equipment between the nodes, and the time to receive and decode the data.

Gigabit Ethernet [8] is a basic network device that offers a cost-efficient design for simple PC cluster. The required network card is either integrated or independent, which is quipped in the most computers, as shown in Figure 3.8. The required cable can be either the Category 5 or Category 6 that is available in most electronics stores. However, its latency is about 120 μ s to 160 μ s.



Figure 3.8 Typical Ethernet network card and switch used in the small cluster. (Images from <http://en.wikipedia.org>.)

Ethernet networking technology is now virtually ubiquitous; it has evolved from 10Base-T (IEEE 802.3) in 1983, Fast Ethernet (IEEE 802.3u) in 1995, and 1-gigabit Ethernet (802.3z) in 1998 to 10-gigabit Ethernet (802.3ae) [9] in 2002. The 10-gigabit Ethernet standard provides a significant increase in bandwidth while maintaining compatibility with the installed base of 802.3-standard interfaces, to protect existing investments in Ethernet technology. The 10-gigabit Ethernet (see Figure 3.9) has been developed to support a wide range of applications from the enterprise network, through the edge and metro network, and into the wide area network. Its latency is from $2.6 \mu\text{s}$ to $4.0 \mu\text{s}$.



Figure 3.9 The 10-gigabit Ethernet network card and switch used in the intermediate-sized cluster. (Images from <http://en.wikipedia.org>.)

Myrinet, as shown in Figure 3.10, is a high-speed local area networking system designed by Myricom to be used as an interconnect between multiple machines to form computer clusters. Myrinet has much lower protocol overhead than standards such as Ethernet, and therefore provides better throughput, less interference, and lower latency while using the host CPU. The end-to-end latency range ranges from $2.0 \mu\text{s}$ MPI latency to $3.2 \mu\text{s}$ [10].



Figure 3.10 Myrinet network card and switch with low latency used in the large cluster with excellent parallel performance. (Images from <http://en.wikipedia.org>.)

InfiniBand, as shown in Figure 3.11, is a switched fabric communications link used in high-performance computing with high throughput and low latency. The InfiniBand architecture specification defines a connection between processor nodes and high-performance I/O nodes such as storage devices. The end-to-end latency range ranges from $1.07 \mu\text{s}$ MPI latency to $2.6 \mu\text{s}$.



Figure 3.11 InfiniBand network card and switch used in the intermediate-sized or large clusters, and has the fabric and coaxial cards, cables, and switches. (Images from <http://en.wikipedia.org>.)

The cost of the network cable is relatively cheap compared to the expensive computers and high-performance network devices. However, if the network cable becomes a bottleneck, the system performance will be significantly degraded. There are two types of popular network cables, Category 5 and 6 cables, as shown in Figure 3.12. Compared with Category 5, Category 6 features more stringent specifications for crosstalk and system noise.



Figure 3.12 A 1-gigabit Ethernet network cable used to connect the two computers directly or from the computers to switch. (Image from <http://en.wikipedia.org>.)

References

- [1] W. Yu, et al., *Parallel Finite-Difference Time-Domain Method*, Norwood, MA: Artech House, 2006.
- [2] W. Yu, et al., "High Performance Conformal FDTD Techniques," *IEEE Microwave Magazine*, Vol. 11, No. 4, June 2010, pp. 42–55.
- [3] <http://openmp.org/wp/>.
- [4] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, 2nd ed., Cambridge, MA: MIT Press, 1999.
- [5] http://web.njit.edu/all_topics/HPC/basement/index.html.
- [6] <http://www.phy.bme.hu/~cluster/docs/PBS.html>.
- [7] GEMS – A Full 3-D High Performance EM Simulation Software and System, State College, PA.
- [8] P. Farrelly and H. Ongz, "Communication Performance over a Gigabit Ethernet Network," *19th IEEE International Performance, Computing, and Communications Conference – IPCCC*, 2000, pp. 181–189.
- [9] J. Hurwitz and W. Feng, "End-to-End Performance of 10-Gigabit Ethernet on Commodity Systems," *IEEE Micro*, Vol. 24, No. 1, January-February 2004, pp. 10–22.
- [10] <http://www.myricom.com/myrinet/overview/>.

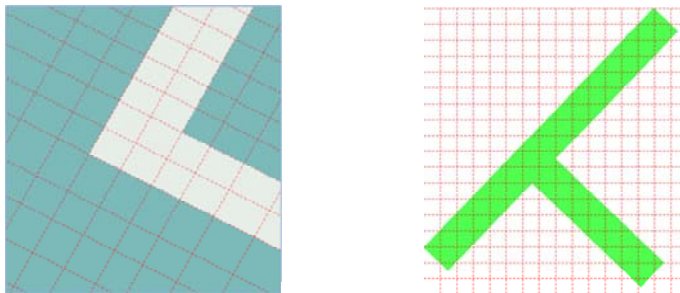
Chapter 4

Electromagnetic Simulation Techniques

In this chapter, we will introduce some key techniques in the FDTD simulation such as the mesh generation and boundary condition selection. Several typical examples such as antenna, antenna array, microwave filter, and signal integrity are employed to explain how to use the FDTD method to solve the practical electromagnetic problems.

4.1 MESH GENERATION TECHNIQUES

The mesh generation technique plays a key role in the FDTD simulation [1–3]. A proper mesh distribution can significantly accelerate the simulation process and can also reduce memory requirements. Even though we have a good electromagnetic tool at our disposal, as well as a powerful cluster on hand to carry out the simulation, it is still necessary to use the proper initial cell size to reduce the simulation time and the memory requirement without sacrificing accuracy. For example, a small gap or a narrow strip line, which is regarded as a fine structure, requires at least two cells to describe the field variation within it, as shown in Figure 4.1. This is true regardless of whether the gap is parallel or antiparallel to the FDTD coordinates.



(a) Mesh for a narrow gap (two cells inside) (b) Mesh for a slanted narrow strip

Figure 4.1 Mesh design requirement for (a) a narrow slot and (b) a narrow strip, in which two cells must be applied.

It is not uncommon to have a fine structure in the problem model, which only serves as a bridge that connects two parts of the geometry. In this case, we do not need to use a small cell size to describe the field variation in the connecting structure. However, we must ensure the continuity of the geometry when modeling this type of structure. Figure 4.2(a) shows the model of a cell phone, which includes some fine structures, for instance, a thin connector that has a very small thickness and is slightly slanted, as shown in Figure 4.2(b). We need not use a small cell size to model its thickness, because it just functions as a connection. Since there may be many fine structures nearby, the adaptive mesh generation may not capture this fine feature. To circumvent this problem, we add a “key” point at one corner of the thin structure, so as to guarantee that the adaptive mesh model will capture the fine feature and ensure that the structure is not discontinuous, as shown in Figure 4.2(c). Choosing the key point will force the mesh to pass through this point, as shown in Figure 4.3. However, we will still need to ensure that the mesh inside the fine structure is continuous.

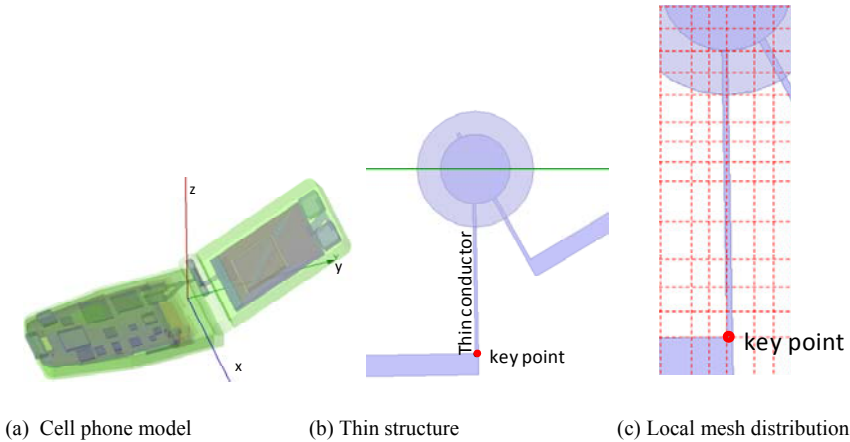


Figure 4.2 Mesh design scheme for (a) a cell phone model, (b) a thin slanted structure, and (c) the local mesh distribution for the thin structure.



Figure 4.3 Using a key point for a specially shaped structure can significantly reduce the number of unknowns.

If a thin structure is parallel to the FDTD coordinates, the mesh generation scheme can capture it without having to use a small cell size. If the fine feature is oriented as shown in Figure 4.4(a), we cannot use just one or two key points to

describe this thin structure. If the conductance of the metal is not infinite, we are using the conductance of copper to model it, we should ensure that the intersection of the copper and mesh edges form a continuous path, as shown in Figure 4.4(b). If we are able to ignore the loss inside the structure, we can model it as a PEC, and the PEC shell model will guarantee that the structure is continuous. To generate a PEC shell, we select the surfaces of the thin structure and reduplicate one while specifying them as PECs. The PEC path is shown in Figure 4.4(c).

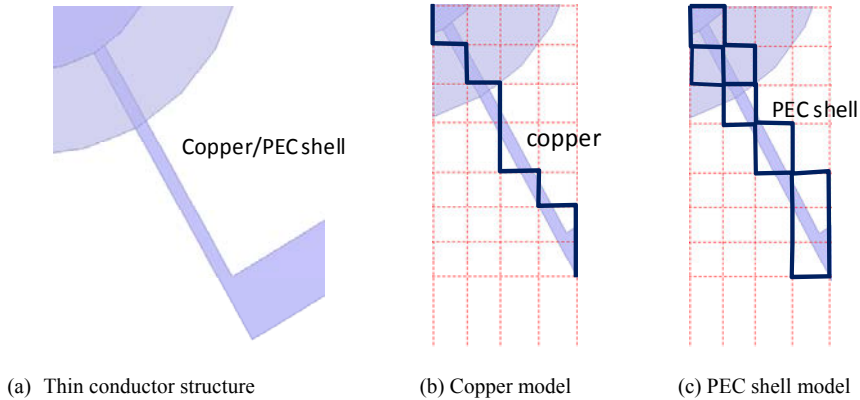


Figure 4.4 Modeling of a thin connecting conductor structure. (a) The thin slanted structure may be PEC or copper. (b) If the thin slanted structure is copper, the conformal scheme shown in the black line will ensure that the thin structure is continuous using a relatively coarse mesh. (c) If the thin slanted structure is PEC, the parallel conformal FDTD code will convert it to the PEC shell and then model it using the conformal technique.

Next, we consider a geometry that includes thin planar structures as shown in Figure 4.5(a). In this case we need not use the PEC ground thickness as a reference for the mesh generation scheme, provided we ignore the skin depth, and the conformal technique can handle this thin PEC structure. For example, if the thickness of the thin PEC plane is 0.01 mm, we can choose the minimum cell size to be 0.5 mm, as shown in Figure 4.5(b), and the simulation result will still be accurate. The mesh distribution for the thin copper structure is shown in Figure 4.5(c).

We should point out that the PEC and finite-conductor conformal techniques utilize different strategies to model the structures. If an edge intersects with a conductor in the finite-conductor model, this edge will also be a finite-conductor. However, the PEC conformal technique will only include the contribution of the PEC part. For example, if the two thin plates are modeled by using the finite-conductor model, and we model the dielectric between the two conductors by using only two cells, the dielectric layer would be replaced by the conductor. To avoid this scenario, we need to use the PEC to model the two thin plates, or else use more than two cells inside the dielectric. However, if we increase the number of cells

inside the dielectric, this will not only degrade the efficiency of the simulation, but will require more memory as well.

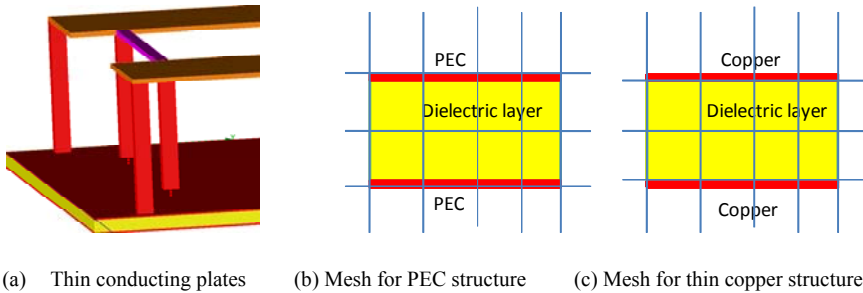


Figure 4.5 Mesh schemes for (a) a thin conductor structure, (b) the PEC materials, and (c) the copper material are treated in different ways in the conformal techniques.

4.2 BASIC SIMULATION PROCEDURE

We should follow the procedure below in the FDTD simulations [3, 4] to achieve a good result for the practical problem solving.

1. Specify a project unit. The project unit is determined by the problem size. For example, if the geometry dimensions are measured in millimeters, millimeter is a proper unit. In any case, we should avoid using a number like 0.000001 to describe a dimension that may introduce an approximation error into the problem geometry. Most commercial EM software tools require specifying the project unit before generating a new project.
2. Draw the project model, or, import the CAD model into the EM solver. If the problem geometry has been in the CAD file, we can import it into the EM software directly. Otherwise, you need to draw it in the EM software. When importing a CAD file, we should check the file format and the version number of the modeling software built in the EM software.
3. Specify the material and assign a proper name and color to each object. Material information is usually provided by the users according to the actual project. Most CAD files do not contain the material information. In the CAD file, the parts with different types of materials cannot be united as a single object since we need to assign them the different materials in the EM software.
4. Draw and specify the excitations and outputs. Most EM software requires drawing an object in the domain first and then assigning a property (excitation or output). For example, we can draw a line and then assign a line as a lumped port, wave port, field output and voltage excitation, or output.
5. Specify the excitation pulse. For an open port, the pure Gaussian pulse will introduce the DC component into the time-domain signal, and in turn

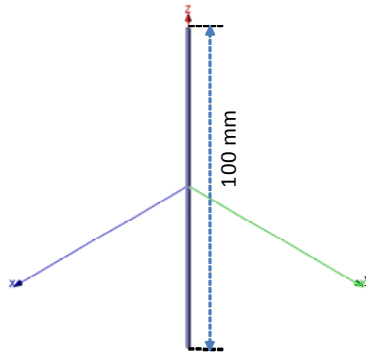
- it may introduce the error into the frequency-domain results. In any case, we cannot use the pure Gaussian pulse in the plane wave source, mode excitation such as TEM, TE, and TM modes, and aperture field excitation.
6. Specify the domain size and boundary condition. In some EM software packages, the domain size is an option for the users, and it can be adjusted in some special applications. For example, the users are allowed to specify the unit size in the periodic structures. Generally speaking, the simulation of the finite or infinite structure is controlled by the white space between the structure and absorbing boundary. Any structure touched by the absorbing boundary will be extended to the infinity, namely, the edge effect will be ignored.
 7. Design mesh distribution. Although the FDTD method allows us to use cell size as large as one tenth of the wavelength corresponding to the highest frequency of interest, in fact, the cell size is determined by the dimension of the fine structure in the practical problem. It is a safe rule to use two cells in the fine structure. The larger ratio between the adjacent cells may introduce the error into the result; however, we need to balance the accuracy, simulation time, and memory usage.
 8. Specify the parallel setting for the cluster simulation. For the parallel processing solver, we can split the domain into small pieces equal to the number of available cores in the cluster. The different distribution of the subdomains may affect the simulation performance.
 9. Generate simulation project file and specify the convergence criterion. We need to tell the EM solver when the simulation is terminated, for example, based on the convergent criterion or the width of the time window.
 10. Submit the job to cluster for simulation. Submit the project to the cluster for simulation for the parallel processing.
 11. Open the display window to visualize the simulation results. Visualize the simulation results in the display window, or export the results to the text file to do the data postprocessing in other software.

4.3 DIPOLE ANTENNA

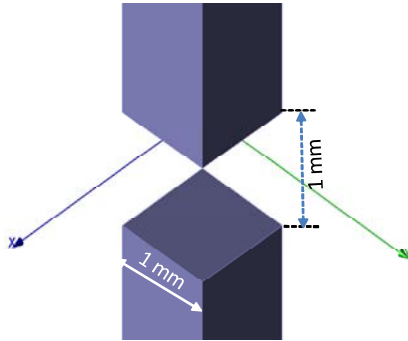
The dipole [4] is one of the simplest antennas, which is often used as the test case to check the accuracy electromagnetic software. In this section, we use the FDTD code to simulate the dipole antenna and describe the most important considerations for the simulation. The length of dipole is 100 mm with a small gap (1 mm) at the center, and the size of rectangular cross section is 1 mm \times 1 mm, as shown in Figure 4.6. The dipole is located in free space, and there is no interference with any other objects. Therefore, the absorbing boundary should be applied to truncate the computational domain. The absorbing boundary in the FDTD simulation just functions as the absorbing wall in the microwave chamber, which is used to erase the reflection from the environment. Today, the most popular absorbing boundary

is the PML that includes several artificial material layers and it perfectly matches to the free space for all the frequencies and at all incident angles.

In this case, the absorbing boundary is used to truncate all six walls around the dipole. The dipole is a finite size, and hence, the absorbing boundary cannot touch the dipole; namely, we need a white space between the absorbing boundary and the dipole. The smaller the white space used and the less memory required, the faster the simulation is. If the absorbing boundary is too close to the dipole, some nonphysical reflection may be caused by the absorbing boundary, and, in turn, the simulation result may be inaccurate. The popular absorbing boundary, PML, requires the number of layers to be six.



(a) Configuration of the dipole antenna



(b) Feed structure of the dipole antenna

Figure 4.6 (a) Dipole configuration and (b) the dimensions of the dipole antenna.

You can also put an absolute distance between the dipole and the absorbing boundary as the white space. However, the number of cells inside white space will vary with the local cell size. It is not a good choice for the practical applications

since we need to know the number of cells to estimate the accuracy, simulation time, and memory usage.

Since the shape of the dipole is a long cylinder, the computational domain will have a similar shape if we use the same size of white space around the dipole. For an irregular domain (the size in one direction is much larger than the other two), the simulation is stable but may not be accurate, especially for the far-field results. The reason is that the nonphysical reflection from the absorbing boundary is involved in the electric and magnetic currents on the surface of Huygens' box.

The excitation of a dipole antenna can be a simple voltage located in the small feed gap that points from the top of the lower arm to the bottom of the upper arm. In order to speed up the convergence, we need to place a lumped element (say, the resistance is 50 ohms) along the excitation path. We select the resistance to be 50 ohms because the return loss will be normalized to 50 ohms.

The output parameters in this example is the return loss, impedance, and far-field pattern. Both return loss and impedance are in a specified frequency range (say, from 1 to 5 GHz), however, the far-field patterns are generated only at the frequencies of interest. It is worthwhile to mention that the number of far-field patterns will significantly affect the performance of the FDTD simulation. Usually, we only generate the far-field patterns at the necessary frequencies. Unlike the far-field pattern, the return loss and impedance are near field parameters, which are calculated from the port voltage and current in the time domain through discrete Fourier transform, and, hence, their output bandwidth does not increase the simulation burden.

Mesh distribution plays an important role in the FDTD simulation, and will decide the result accuracy and simulation performance. In this example, the fine structure is the cross section of the dipole. The thickness of the dipole is not critical if it is relatively small compared to its length. Though one cell is sufficient for the accurate result, we like to use two cells here since the excitation can easily be set to be symmetric. The mesh distribution in the vertical direction is independent, which can be uniform or nonuniform, however, the ratio of cell size must be in the vertical and horizontal directions smaller than 20 to get a good simulation result. The nonuniform mesh in the vertical direction may slightly influence the simulation result; however, it will be acceptable in most cases. The number of cells in the vertical direction inside the feed gap is between two and four. The feed gap of dipole cannot be too large; otherwise, the higher mode in the port voltage and current will introduce error into the return loss and impedance. If possible, we usually select the size of the feed gap in the vertical direction to be two cells regardless of the cell size.

Generally speaking, the simulation result is independent from the type of excitation pulse. However, it may affect the convergence and the accuracy as well. Pure Gaussian pulse has the largest DC component in its spectrum. However, if there is no resistance used in the excitation source, the pure charges left on the two dipole arms that are caused by the pure Gaussian pulse will introduce the DC component into the time signature in both the port voltage and current. This DC component will not allow us to directly use the DFT or FFT to generate the correct

return loss and impedance. For the same reason, the far field is not accurate either. If a resistance is used along the excitation voltage, either a pure Gaussian or a differential Gaussian pulse will generate the accurate result. One of the best choices to select the 3-dB beamwidth in both the pure and differential Gaussian pulses is equal to the highest frequency of interest (say, 6 GHz). The return loss and impedance of the dipole antenna are plotted in Figures 4.7 and 4.8. The 2-D directivity pattern and 3-D far-field pattern are plotted in Figures 4.9 and 4.10.

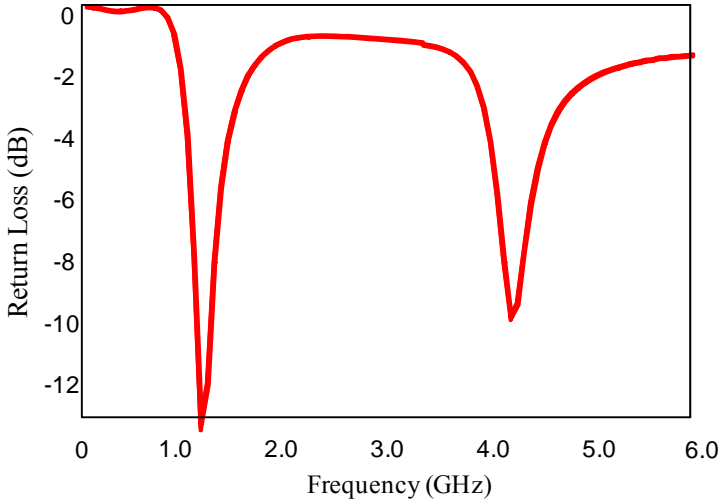


Figure 4.7 Return loss of the dipole antenna from the low frequency to 6 GHz.

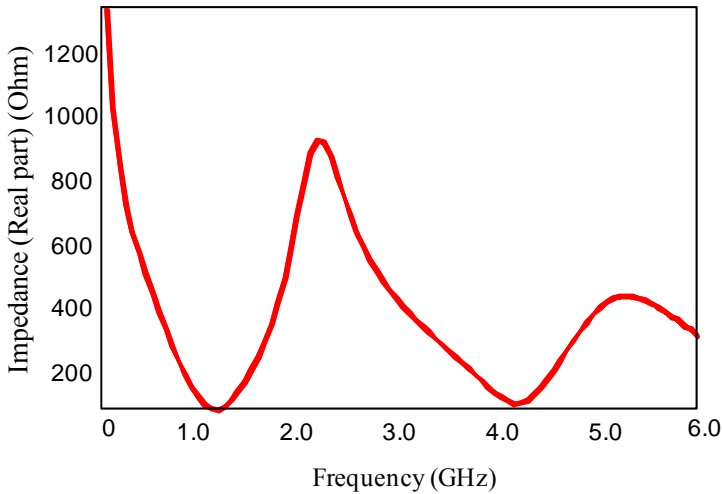


Figure 4.8 Impedance of the dipole antenna from the low frequency to 6 GHz.

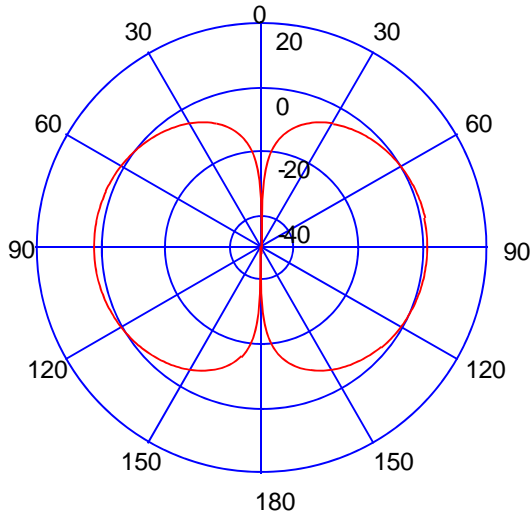


Figure 4.9 Directivity pattern of the dipole antenna in $\phi=90^\circ$ plane at 1.38 GHz.

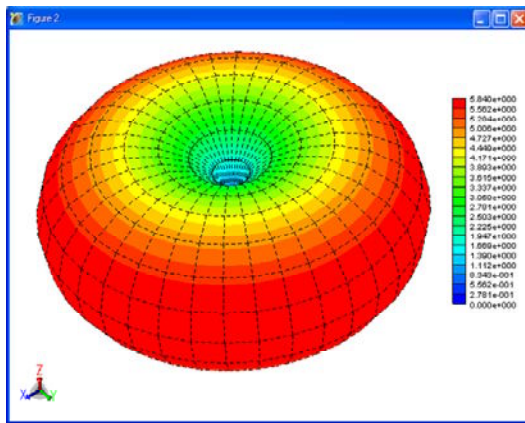


Figure 4.10 3-D far-field pattern of the dipole antenna at 1.38 GHz.

4.4 VIVALDI ANTENNA SIMULATION

A Vivaldi antenna [5] is shown in Figure 4.11, which is often used as a test case to validate the EM software. The antenna includes three pieces of Vivaldi-shaped metals and two layers of dielectrics. Due to the complex shape, this type of antenna is usually designed in the modeling software and imported into the EM simulation software. Before importing it into the EM simulation software, we need to know if the file format matches to the required format. If not, we can use other software to convert the format to the required one. In addition, the pieces with the different

materials should not be united in the original CAD file; otherwise, we cannot specify the material to different structures. If the model is correctly imported into the EM software, we assign the proper material and color to each structure.

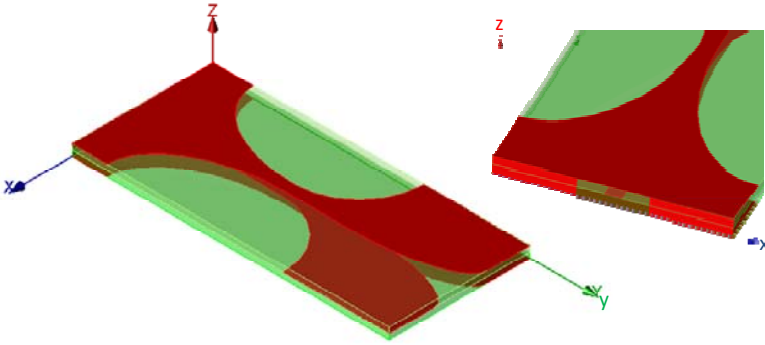


Figure 4.11 Configuration of the Vivaldi antenna and the feed structure.

The first thing we need to do in the simulation is to find a way to excite the Vivaldi antenna. Since the model does not have a feed structure, we must add one for it. It appears that there are an inner conductor and an outer conductor at the feed port; and we can extend the inner conductor, dielectric, and outer conductor to a proper length, as shown in Figure 4.12. The extend length is usually 10 to 20 cells to get accurate results. We should have an idea about the cell size when we extend the length of the feed structure. For example, it is equal to 10 times the width of the inner conductor if the cell size is about half the width of the inner conductor.

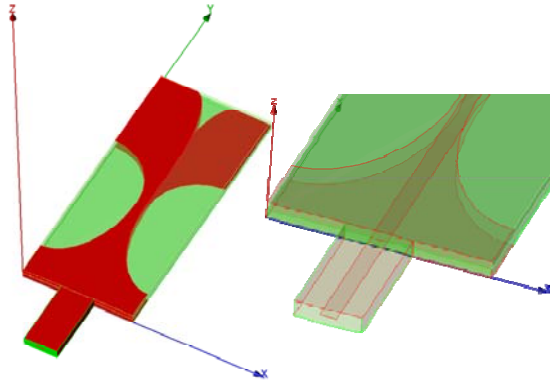


Figure 4.12 Extended feed structure of the Vivaldi antenna.

Next, we will describe the simulation procedure to generate a good result. The output parameters in the problem are the return loss and far-field pattern. Therefore, the antenna structure including the feed structure must be finite to apply a closed Huygens' surface to calculate the far-field pattern.

For the antenna problem, the metal pieces can be either the perfect electric conductor (PEC) or good conductor (say, copper). If the metal structure has no thickness, it can be only assigned to be PEC since the actual structure should have a finite thickness. If the metal structure has a finite thickness, it can be assigned to be either the good conductor or PEC.

For a finite structure problem, absorbing boundary is applied to truncate the six walls of the computational domain. The white space between the antenna and absorbing boundary is equal to six cells. The domain size in the vertical direction is much smaller than those in the horizontal directions, which is also an irregular domain and may result in an unstable solution. In case the solution is unstable, we need to increase the size of the white space in the vertical direction to get a stable solution.

Generally speaking, we use two to three cells in the substrate and superstrate and two cells across the feed line (inner conductor). However, the nonuniform mesh should capture the key points to describe the field variation around these points.

We need to terminate the end of the feed structure using the match load inside the domain since any structures of the antenna must be enclosed inside the Huygens' surface. The PML is one of the best candidates to terminate the feed structure without nonphysical reflection. However, we need to use the PML to create a virtual boundary that serves as the match load to terminate the excitation port inside the domain. The virtual boundary connects to the main domain though a field distribution on the excitation port.

Since this feed structure whose cross section is a rectangular coax supports the TEM mode, we can extract the mode pattern supported by the coax through the numerical approach, as shown in Figures 4.13 and 4.14. The patterns will be applied to excite the Vivaldi antenna.

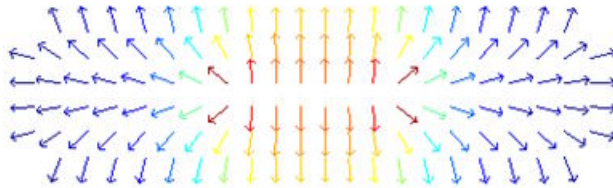


Figure 4.13 Excitation pattern of the electric field distribution inside the feed structure.

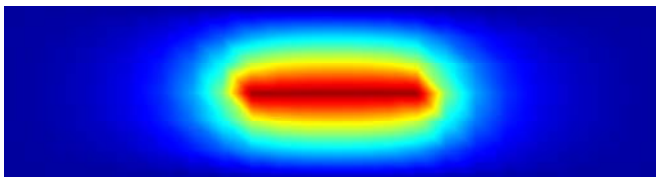


Figure 4.14 Excitation pattern of the electric potential distribution inside the feed structure.

The FDTD simulation is to extract the system properties, which is not related to the excitation pulse. However, the excitation pulse will affect the simulation convergence and accuracy as well. Since the match load in this example is realized by the PML layers, which is unlike the internal resistance of the excitation source, the pure Gaussian pulse will introduce the DC component into the time signature in the time-domain results. In turn, it will cause the truncation error in either DFT or FFT. Therefore, in this example, we should use the differential Gaussian pulse and the 3-dB beamwidth frequency is taken to be 10 GHz, which is equal to the highest frequency of interest.

The return loss is also for the TEM mode from 1 GHz to 10 GHz. We need to measure the field distribution on the cross section of the feed coax, which should be two cells from the excitation source. To calculate the return loss, we need to extract the mode voltage and current from the measured field distribution to the extracted TEM mode pattern. The return loss is calculated by using the mode voltage and current. The bandwidth of the return loss will not increase the burden on the computer. However, the accuracy at the higher frequencies is determined by the maximum cell size. For example, the maximum cell size should be smaller than $\lambda/10$ of the highest frequency of interest to get an accurate result. The minimum cell determined by the fine structures in the problem geometry will affect the size of the time step in the simulation; in turn, it will significantly degrade the simulation performance and increase the simulation time. Therefore, in the practical applications, we first find the minimum structure of interest and then select the minimum cell size to be one-half the dimension of the fine structure. The return loss of the Vivaldi antenna is plotted in Figure 4.15, which demonstrates the wideband property.

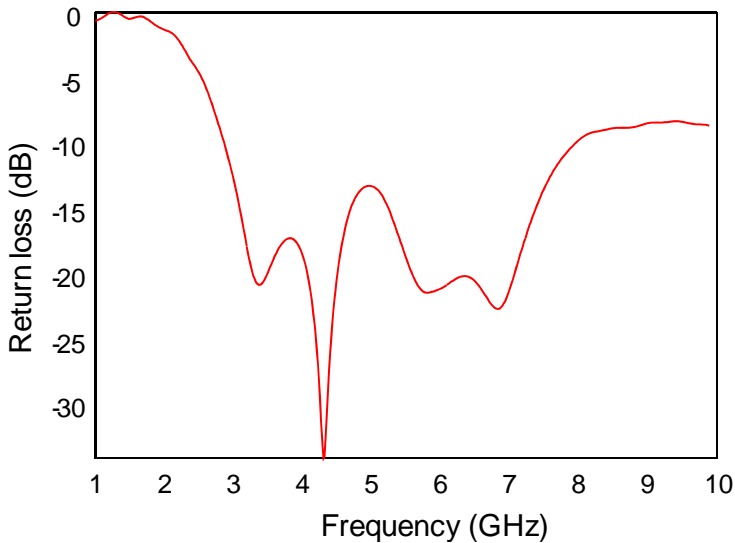


Figure 4.15 Return loss of the Vivaldi antenna from 1 to 6 GHz.

To calculate the far-field pattern, we need to define a closed surface, Huygens' surface, and to calculate the electric and magnetic current distribution on it. The far-field pattern will be calculated by using the equivalent electric and magnetic currents. During the calculation of the equivalent currents, the spatial and time-domain downsampling techniques will be applied to reduce the simulation time and memory requirement. For example, the size of the time step in the FDTD simulation is much smaller than the time step for the DFT from the time-domain currents to the frequency-domain currents. Therefore, we can take the DFT on the much coarser time interval than the FDTD time step. Similarly, we do not need to sample the electric and magnetic currents on the Huygens' surface on the FDTD grids because the near-to-far-field transformation requires much coarser grid than it. Using the downsampling techniques above, we can dramatically reduce the time on the near-to-far-field transformation for the large antenna problems.

The far-field patterns at 6 GHz in the X0Z and Y0Z planes are plotted in Figure 4.16. The symmetric feature along the z -direction is obvious in both planes from Figure 4.16. The far-field patterns at 6 GHz in the X0Y plane is plotted in Figure 4.17, which is not symmetric due to the asymmetric Vivaldi structure.

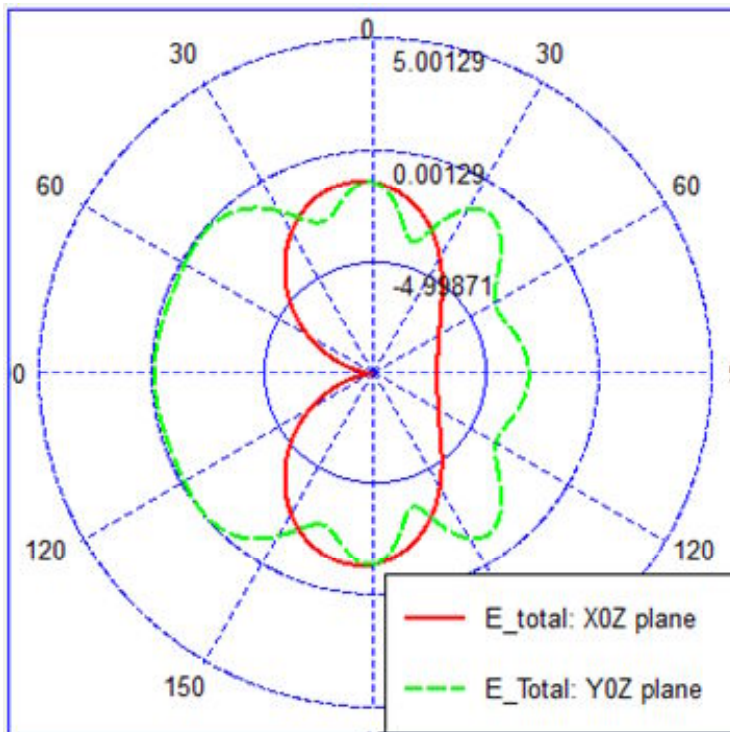


Figure 4.16 Far-field pattern of the Vivaldi antenna at 6 GHz in the X0Z and Y0Z planes.

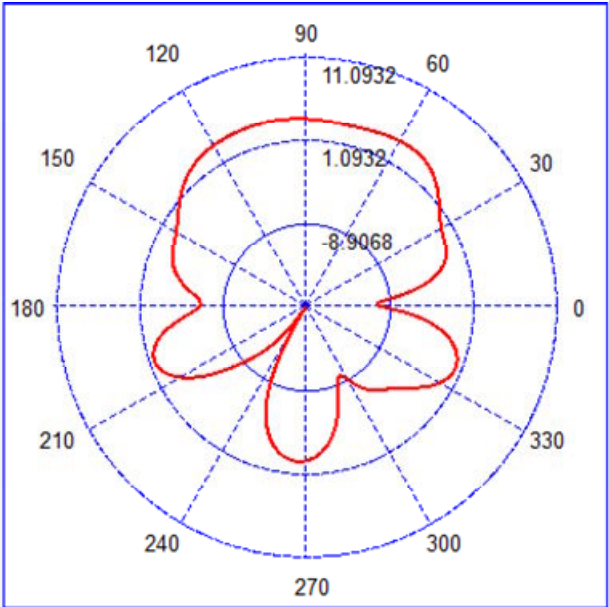


Figure 4.17 Far-field pattern of the Vivaldi antenna at 6 GHz in the X0Y plane.

4.5 BANDED MICROWAVE CONNECTOR

Connectors are the most frequently used microwave components in the various microwave devices [6]. In this part, we introduce the simulation techniques for a microwave connector (see Figure 4.18) using the FDTD method.

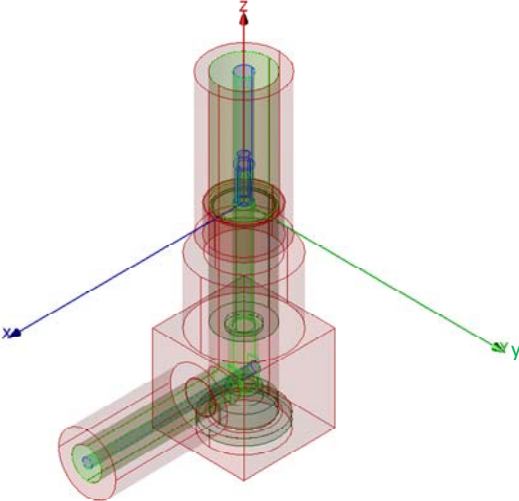


Figure 4.18 Configuration of the microwave coaxial connector with a 90° angle.

For the complex structure problems, the model is in the CAD file and is usually created from the other commercial modeling software. We need to import it into the EM simulation software and assign the material and color to each piece. We follow common sense to go through the simulation procedure next.

In this example, the output parameters of interest are the time-domain reflectometry (TDR) and the field distribution inside the connector. For the TDR calculation, the excitation pulse must be a narrow pure Gaussian. The width of the pure Gaussian pulse should be narrow enough that it allows us to separate the incident pulse from the reflected signature measured at the excitation port.

This is not an open space problem since we are only interested in the TDR and the field distribution inside the connector. The size of the computational domain equals the range occupied by the connector. We use the absorbing boundary to terminate the two open ports to extend them to infinity, and the other four walls can be terminated by using the PEC that is the simplest boundary. The white space is zero; namely, both absorbing and PEC boundaries should touch the connector.

To generate the mesh distribution, we need to check the size of fine structures of interest in the structure; for example, the dimensions of fine structures marked in Figure 4.19(a) are 0.1 mm, and hence, the minimum cell size is selected to be 0.05 mm. The local mesh distribution is shown in Figure 4.19(b), and the key points of the fine structure are on the FDTD grids. For the TDR calculation, we do not have to wait until the measured signal converges, but we can stop the simulation when the Gaussian pulse completely passes the connector, as shown in Figure 4.20.

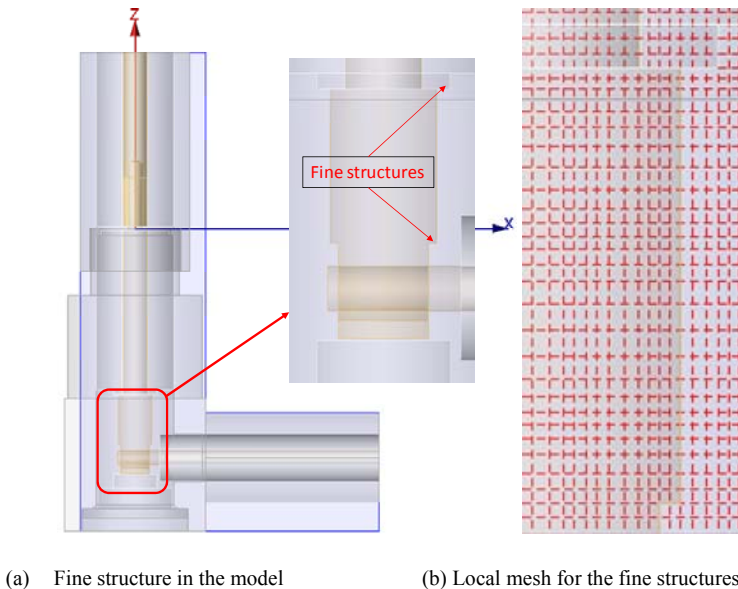


Figure 4.19 (a) Fine structures inside the connector and (b) the corresponding mesh distribution to describe the field variation in the fine structure area.

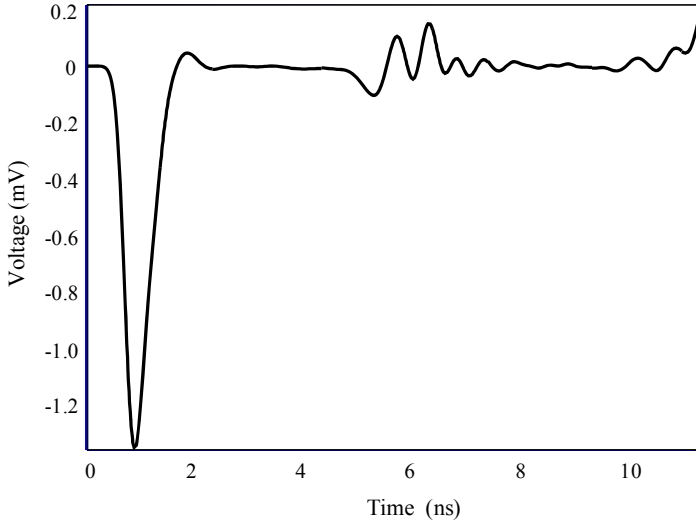


Figure 4.20 Time signature measured at the excitation port of the microwave connector. The incident pulse should be separated from the reflected signal for the TDR calculation.

Supposing that the top port serves the excitation port, we need to extract the mode supported by the cross section. We can generate the mode pattern, TEM mode, supported by the coaxial structure by the analytic approach. However, if the shape of the inner or outer conductors is not perfectly circular, the error will be introduced into the simulation. Therefore, we can use the numerical method to extract the mode pattern if we know the material distribution in the cross section. The mode pattern (TEM) [1] is shown in Figure 4.21.

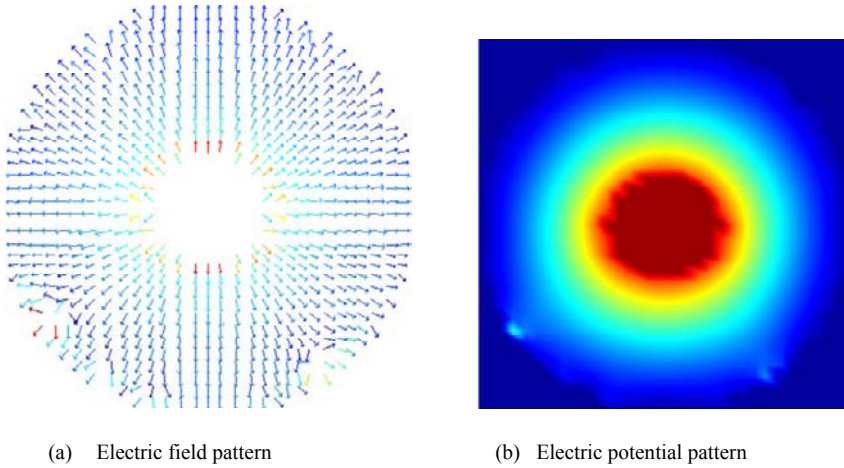


Figure 4.21 (a) Electric field pattern and (b) magnetic field pattern supported by the coaxial structure.

If we consider the 3-dB beamwidth frequency of the pure Gaussian pulse to be 60 GHz, the TDR is plotted in Figure 4.22.

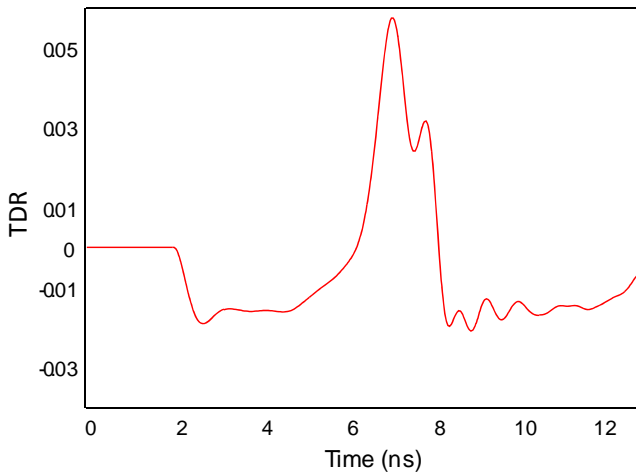


Figure 4.22 TDR of the microwave coaxial connector. The flat part indicates the uniform structure and there is no reflection when the pulse propagates through it.

To view the field propagation procedure inside the connector, we need to define two rectangles inside the connector and specify the field components we would like to output, as shown in Figure 4.23. Since the output file will be large if the rectangles are large, we limit the output region inside the connector. The field distribution at the specified time is shown in Figure 4.24.

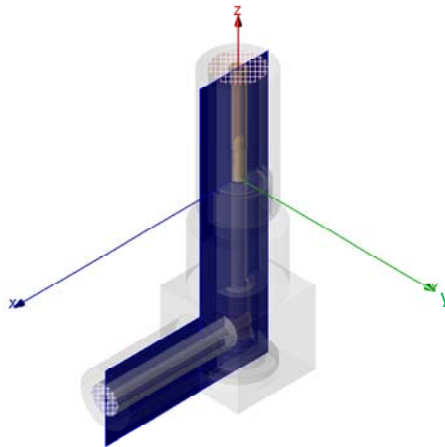


Figure 4.23 Defined rectangular area inside the connector for the field output, which includes two rectangles.

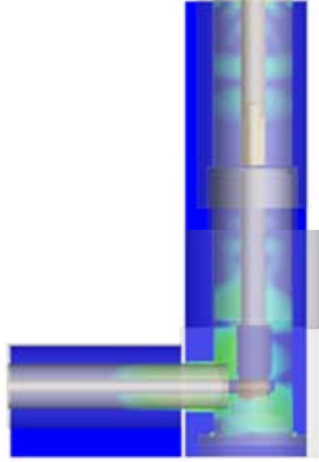


Figure 4.24 Normalized electric field distribution of the electric field at the specified time inside the connector.

4.6 PARALLEL LINES

This is a simple model that is taken from the real chip model [2], as shown in Figure 4.25. Two parallel lines are embedded inside four layers of anisotropic media, as shown in Figure 4.26. The medium has the different permittivity in the vertical direction from that in the horizontal directions.

The four layers of dielectric are finite and backed by a thin metal ground. If the PEC ground has no thickness, we should draw the dielectric layers first and then the PEC ground; otherwise, the PEC ground will be erased by the touched dielectric layers.

The structure is finite and the problem is an open problem. The absorbing boundary is used to truncate the six walls. The size of the white space is six cells around the structure.

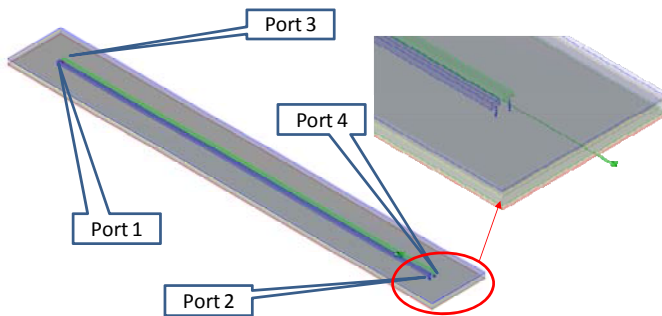


Figure 4.25 Configuration of the parallel signal lines and the port specification for the S-matrix calculation.

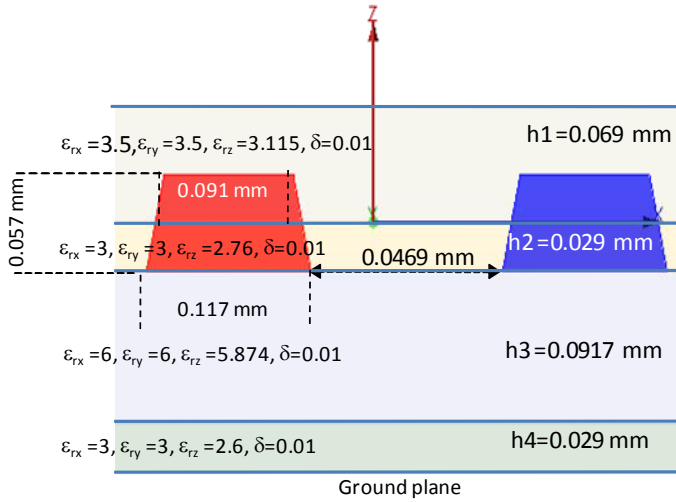


Figure 4.26 Configuration of the parallel signal lines and dielectric layers, and the dimensions and dielectric parameters of the signal lines and dielectric layers.

The parallel lines include the slanted surfaces, and it is impossible for us to use FDTD grids to capture all the edges with using the extremely small cells. In this example, the bottoms of the parallel lines are important since the distance of bottoms will affect the coupling between two parallel lines. There are two important factors to generate the mesh distribution: (1) there are at least two cells between the parallel lines, and (2) the FDTD grids should capture the bottom corner of the parallel lines. The proper mesh distribution around the parallel lines is shown in Figure 4.27.

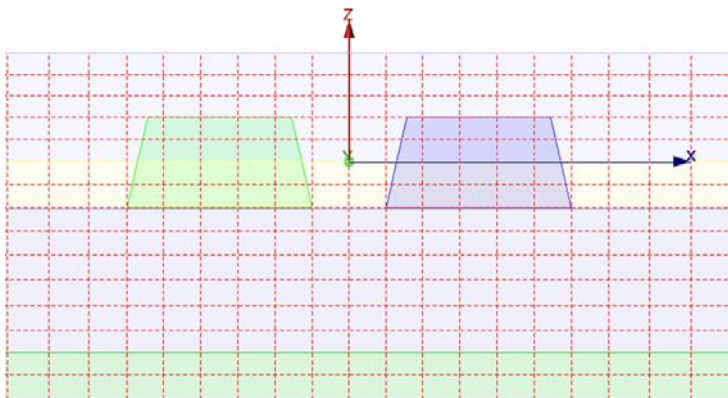


Figure 4.27 Local mesh distribution to capture the key points on the signal lines and the interfaces of the dielectric layers.

To calculate the S-parameters, we need to terminate the four ports using the match loads at the ends of the parallel lines. The match load can be a surface whose width and height should be the bottom width of the signal line and the height from the ground to the signal line, respectively. Also, we can simply use a line from the ground to the signal line and assign it to a resistance (say, 50 ohms), as shown in Figure 4.28.

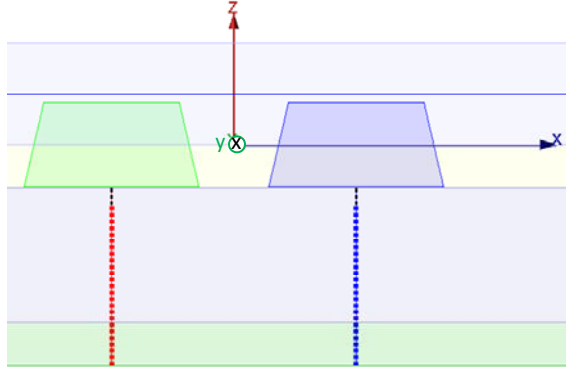


Figure 4.28 A typical port definition with a match load and voltage excitation.

One of the ports is excited with the pure Gaussian pulse or differential Gaussian pulse. The port voltage is measured at the same location as the excitation. The current is calculated from the magnetic field loop around the excitation source and measured at the middle of the ground and signal line. Using the same method, we can define the output port voltages and currents at three output ports. The S-parameters are plotted in Figure 4.29.

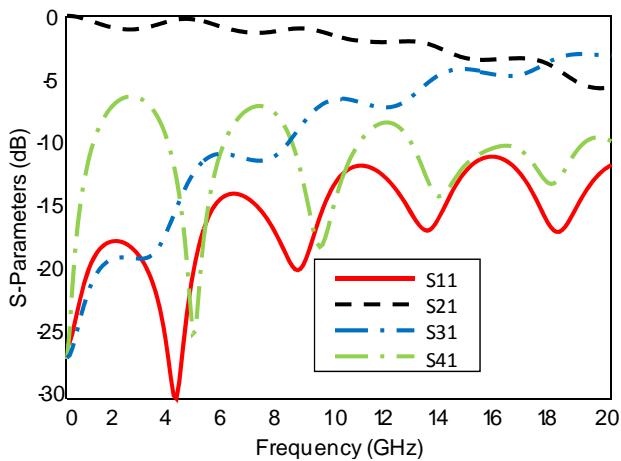


Figure 4.29 S-parameters measured at each port when the port 1 is excited.

4.7 TWO-PORT ANTENNA

This antenna includes two ports [personal communication with Niels Schutten, 2009], as shown in Figure 4.30. The thin PECs are located at both sides of the dielectric slab. The dielectric slab has the finite size, and two ports are located at one side of the dielectric slab.

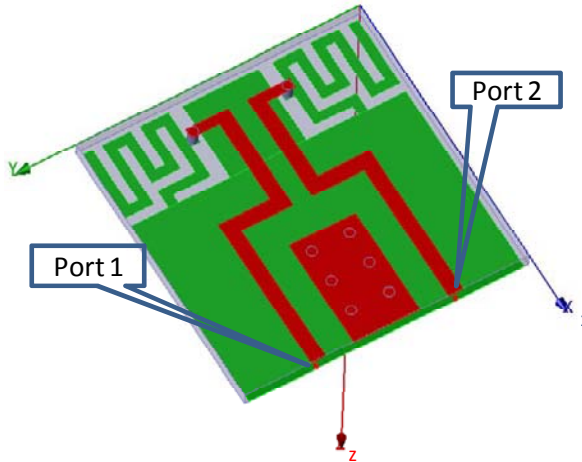


Figure 4.30 Configuration of the two-port antenna and the port definition.

If the distance between the ports is far away and there is no other structure closed to the port, we can define a mode port, as shown in Figure 4.31. The height of the mode port is about four times larger than the thickness of the dielectric slab, and its width is in a proper value; for example, three to four times the width of the signal line.

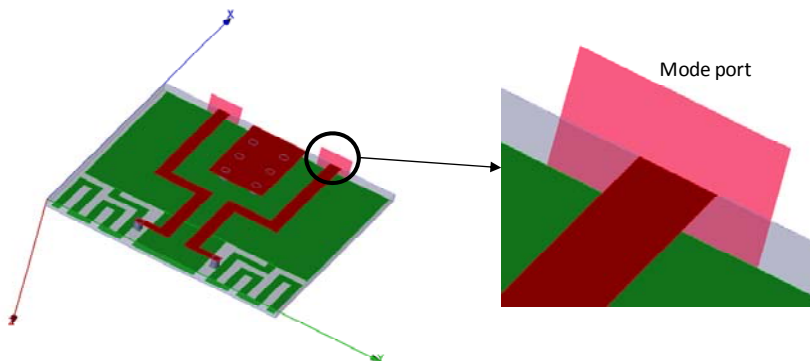


Figure 4.31 Configuration of the mode excitation port definition, whose dimension should be large enough to capture the major power of the TEM mode.

The domain size in the vertical direction is much larger than the thickness of the dielectric slab. The result is usually better than that by using the voltage excitation. If we use the frequency-domain method such as the finite-difference frequency-domain (FDFD) method that is based on the same mesh, we should extract the mode pattern after the mesh generation to avoid the spatial interpolation. The boundary in the FDFD method in this case is the absorbing boundary. The mode pattern is shown in Figure 4.32.

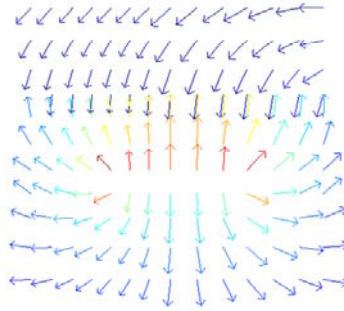


Figure 4.32 Electric field pattern of the excitation mode and we can see the location of the dielectric layer, microstrip line, and ground plane in the mode pattern.

Since there are two ports in this antenna, to calculate the S-parameters, we need to run the FDTD code twice by switching the excitation and output ports. We can run it one time and switch the port internally; specifically, we can generate the mesh and material distributions one time. The S-parameters are plotted in Figure 4.33.

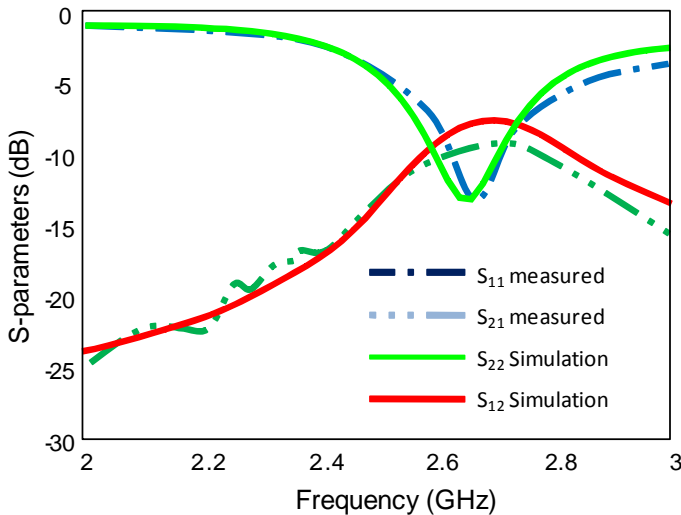


Figure 4.33 S-parameters of the two-port antenna when the two ports are excited separately.

Though most structures in this antenna are in a rectangular shape, we show you how to generate the mesh for the good results. The fine structure in the antenna is marked in Figure 4.34. We need to use two cells in the fine gap. In the FDTD mesh generation, we need to pay the special attention to the small gap formed by the PEC objects, especially formed by the thin PEC objects. Any approximation in the gap will short two PEC objects and the results will be incorrect.

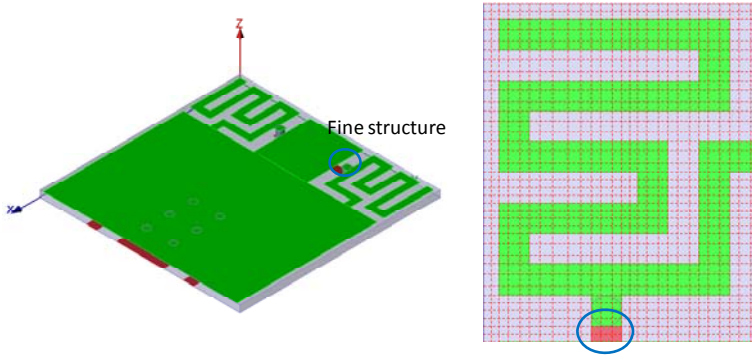


Figure 4.34 (a) Fine structure inside the antenna and (b) local mesh distribution to describe the field variation inside the fine structure.

4.8 SLOT COUPLING

Two slots are cut on an infinitely large PEC sheet, as shown in Figure 4.35. The geometry of this example is provided by Professor Junho Yeo of Daegu University, Korea. In this example, the plane wave illuminates on the slots and we need to calculate the transmitted power through the slots. If the PEC sheet is finite, this problem is relatively easy. However, it is difficult to apply the plane wave source on the infinitely large object since it is in conflict with the definition of the plane wave. We can use the Gaussian beam inside the PML to reduce the truncation error of the plane wave source.

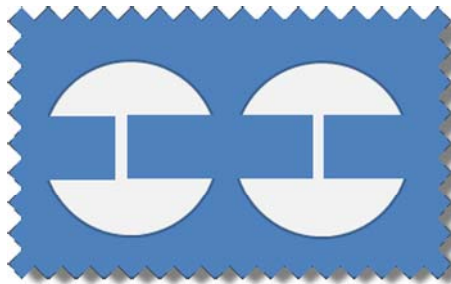


Figure 4.35 Pattern of the slot array on an infinitely large PEC sheet.

However, this problem can be solved by solving its dual problem, namely, switch the PEC and slot part, as shown in Figure 4.36. We need to switch the excitation polarization of electric and magnetic fields. If the PEC sheet has a finite thickness, we cannot use the dual method any more.

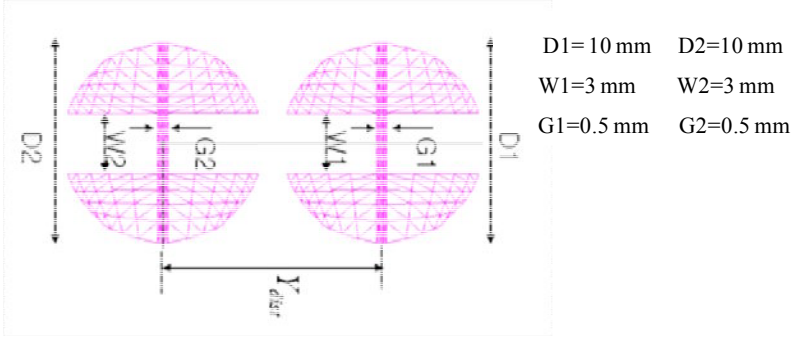


Figure 4.36 Dual problem of the slot array that is solved by using the MoM method. Its solution is the same as the original problem.

The PEC sheet extends to infinity in the horizontal directions. The absorbing boundary is used to truncate the PEC sheet without the white space (the domain size is four to five times larger than the slot dimension) between the PEC sheet and absorbing boundary. In this way, the absorbing boundary will extend the PEC sheet to infinity. The vertical direction of domain is also truncated by using the absorbing boundary and a proper size of white space is applied (say, 20 cells).

The fine structure is the small gap inside the slot. Following the general rule, there must be two cells inside the small gap to get a good result. However, in this problem, the coupling is relatively low and the accuracy of field inside the small gap is critical. The numerical experiments have demonstrated that we need at least four cells in the small gap to get accurate results, as shown in Figure 4.37.

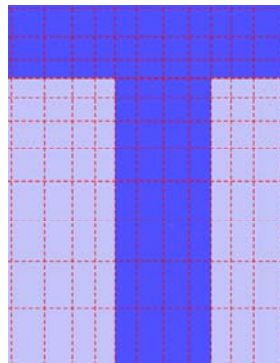


Figure 4.37 Local mesh distribution around the narrow gap of the slot array.

The plane wave source located between the top boundary and slot is incident from the z -direction with the E_x polarization, as shown in Figure 4.38.

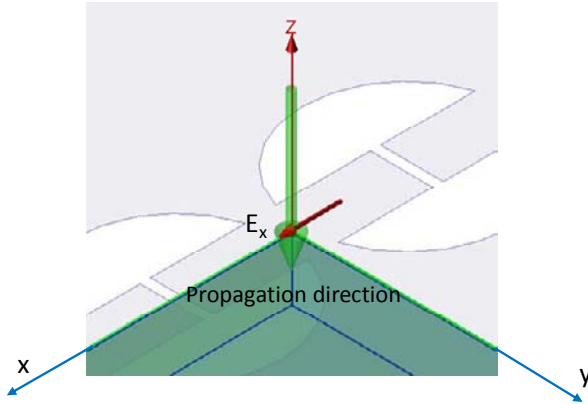


Figure 4.38 Propagation direction and polarization of the plane wave incident on the slot array structure.

The output is the transmitted power through the slots when the plane wave is incident in the normal direction. A box without the top surface below the PEC sheet captures the transmitted power [2]. The variation of transmitted power with the distance (11 mm, 20 mm, and 29 mm) of the slots is plotted in Figure 4.39 together with those obtained by using the MoM code. The excellent agreement between the FDTD and MoM is evident from Figure 4.39.

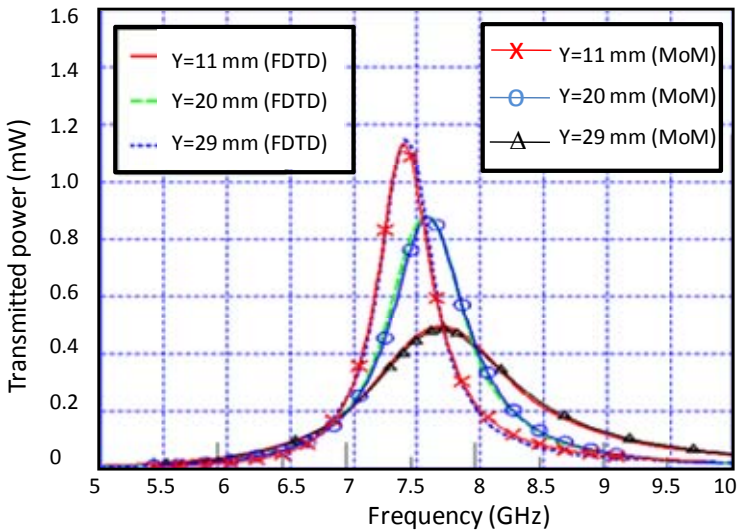


Figure 4.39 Transmitted power through the slot array using the MoM and FDTD methods.

Due to the symmetric structure, we may only need to simulate one quarter of the structure using the PEC boundary in the minimum x -direction and the PMC boundary in the minimum y -direction. We can also use an alternative way to validate the FDTD simulation results. For example, we change the shape of slot to be a square with 4-mm side length, as shown in the inset of Figure 4.40. For the same plane wave source and polarization, the variation of the transmitted power with the frequencies is plotted in Figure 4.40. As we can see from Figure 4.40, the transmitted power obtained by using the FDTD method approaches 1.6 mW when the frequency increases, which is exactly what we expect.

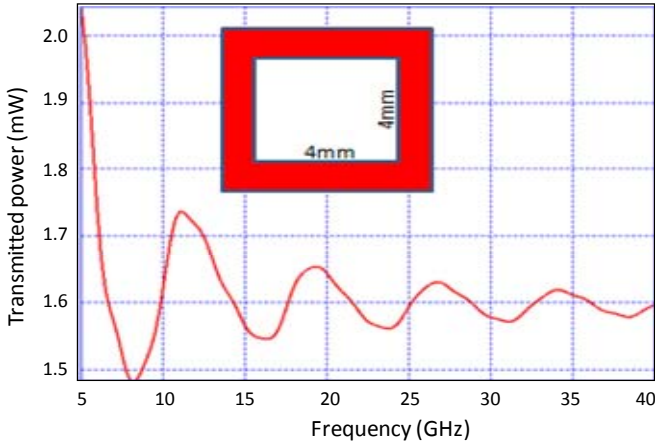


Figure 4.40 Transmitted power through a square slot using the FDTD method, which should approach 1.6 when the frequency increases.

It is worthwhile to mention that the excitation pulse in the plane wave source case cannot be the pure Gaussian pulse; otherwise, the DC component will introduce the error into the simulation results. Once again, the 3-dB beamwidth frequency of the differential Gaussian pulse is 10 GHz, which is the highest frequency of interest.

4.9 MICROWAVE FILTER

This example is a microwave filter [6], as shown in Figure 4.41. The feed waveguide is not a regular rectangle. The model is in the STEPS file format and can be imported into the FDTD code. In some cases, the problem modeling is even more difficult than the simulation. For example, this model is a solid in its original format. To generate a hollow waveguide, we select all the surfaces of the model and duplicate the surfaces and then delete the original model. The actual waveguide model does not include the two end surfaces, and hence, we deselect the two end surfaces, and assign the PEC material to the waveguide. The two end surfaces are directly defined as the input and output ports.

This is not an open space problem and the two ports are extended to infinity in the propagation direction. We use the absorbing boundary to truncate the two ports and the PEC boundary to truncate the other four walls. Since the waveguide wall has no thickness, any reasonable cell size can generate a closed waveguide surface. If the waveguide wall has a finite thickness, we have to select the cell size to be smaller than the half dimension of waveguide thickness to ensure the waveguide not leaking.

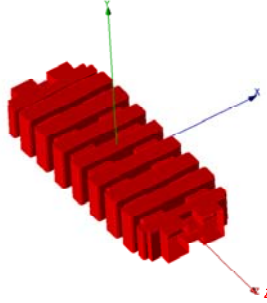


Figure 4.41 Configuration of the waveguide filter with the infinitely thin PEC wall.

We select the two end surfaces and use the PEC boundary to truncate the 2-D domain in the FDFD method. The first mode is the transverse electric (TE) mode, as shown in Figures 4.42 and 4.43 whose cutoff frequency is 2.37 GHz. The reflection and transmission coefficients are plotted in Figure 4.44.

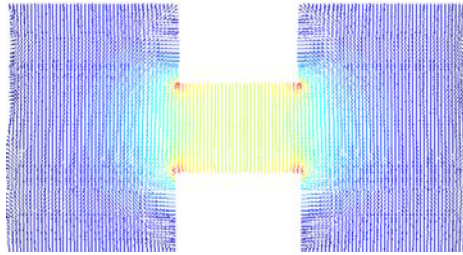


Figure 4.42 Normalized electric field pattern of the TE mode at 2.37 GHz at the excitation port.

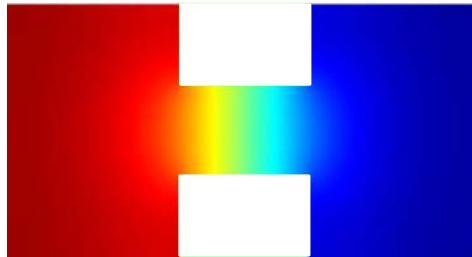


Figure 4.43 Normalized magnetic field pattern of the TE mode at 2.37 GHz at the excitation port.

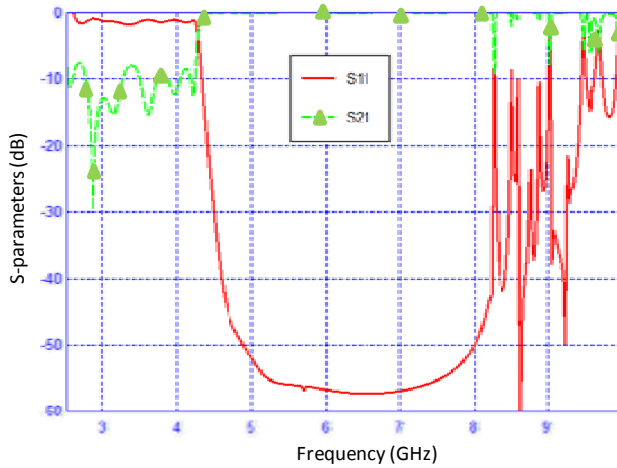


Figure 4.44 Reflection and transmission coefficients of the microwave filter from 2.6 to 10 GHz.

For a high Q system, the solution converges relatively slowly. Once again, the 3-dB beamwidth frequency of the differential Gaussian pulse is 10 GHz, which is the highest frequency of interest.

4.10 OPTIMIZATION AND PARAMETER SCAN

This example is a patch antenna array [7], as shown in Figure 4.45. The substrate and ground plane are finite in the horizontal directions. The four patches sit on the top surface of the substrate. There is a free space gap between the substrate and the ground PEC plane. In this example, the output parameters are the return loss and far-field pattern, and their variation with the height of free space gap. The antenna is fed by a coaxial cable in the original design, as shown in Figure 4.46.

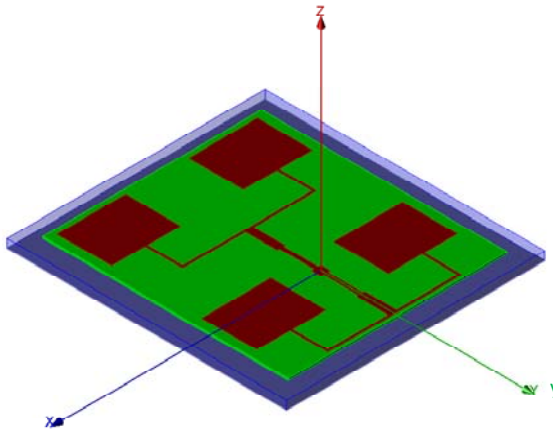


Figure 4.45 Configuration of the patch antenna array fed by the microstrip line network.

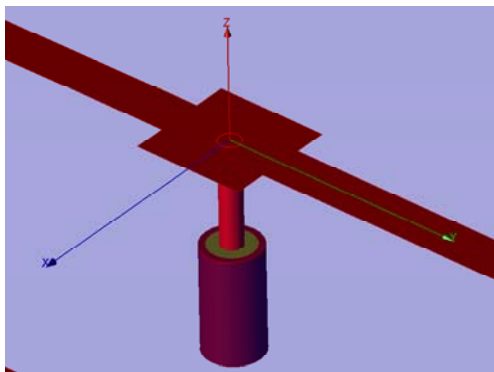


Figure 4.46 Antenna array fed by a coaxial cable used in the experiment.

If we directly simulate the antenna with the coaxial cable, the cell size will be small and the simulation time will be much longer. Hence, we replace the coaxial cable with a simplified probe, as shown in Figure 4.47. A large number of numerical experiments have demonstrated that the simplified model can generate the very good results for both the return loss and far-field pattern. Using the simplified probe model, the simulation is much faster with the much lower memory requirement.

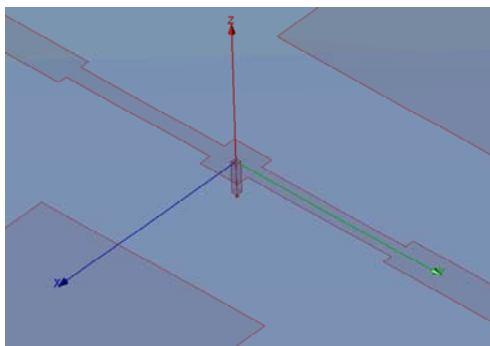


Figure 4.47 Antenna array fed by the simplified probe to avoid meshing the coaxial cable.

If we properly select the thickness of probe and feed gap between the probe and ground, as shown in Figure 4.48, the simulation results will be accurate using lesser computer resources. The best choice is that the cross section of probe is 2×2 cells and two to four cells are between the probe and ground, as shown in Figure 4.49. In this example, we need to investigate variation of the output parameters with the distance change between the dielectric and ground. The probe length and ground position will vary accordingly but the feed gap remains the same in each case.

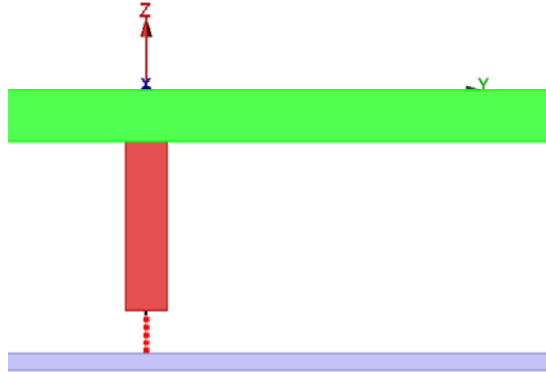


Figure 4.48 A probe touching the feed structure and has a small gap (say, two cells) from the ground plane. A voltage excitation is located in the gap between the probe and ground plane.

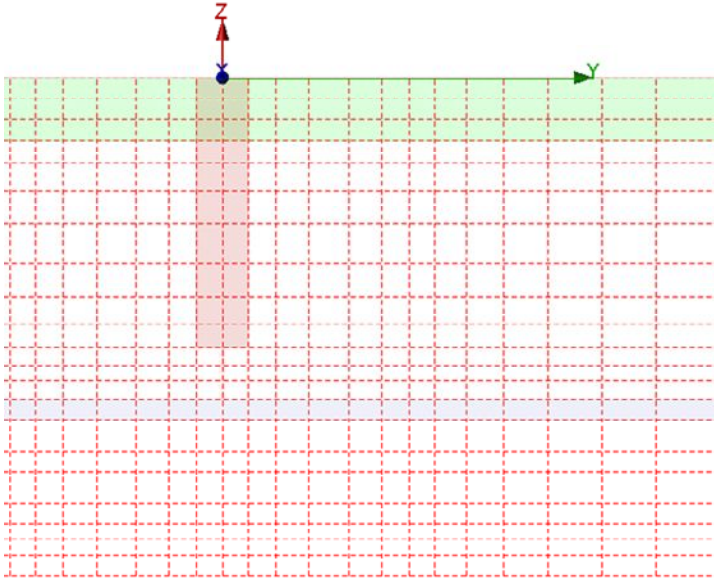


Figure 4.49 Local mesh distribution between the probe and ground plane that varies with the change of the gap dimension.

We like to vary the mesh and material distributions only in the region where the structures vary in each simulation. Mesh and material distributions in the rest area remain the same in each simulation. It is the partial reason why the cells inside the variation region must match to those in its neighboring regions.

To investigate the variation of the output parameters with the specified dimensions, we usually define a parametric variable (say, H) by which the mesh and material vary each time, as shown in Figure 4.50.

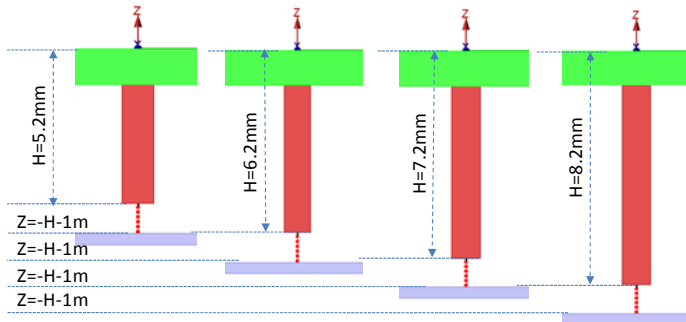


Figure 4.50 Variation of the parametric variable H that controls the probe length, namely, gap size between the patch and ground plane.

The FDTD grids capture all the key points on the patch, feed probe, and feed network, as shown in Figure 4.51. There are three cells across the fine feed line. The 3-D far-field pattern at 2.3 GHz is plotted in Figure 4.52 together with the problem model. The S-parameters at different sizes of the free space gap are plotted in Figure 4.53. Though we do not show the measurement data here, the simulation results match very well with the measurement data.

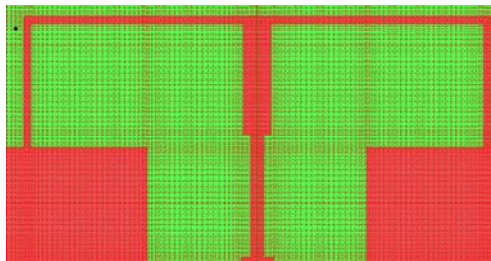


Figure 4.51 Local mesh distribution along the feed network structure. The mesh should always capture the key points on the patches and feed network structure.

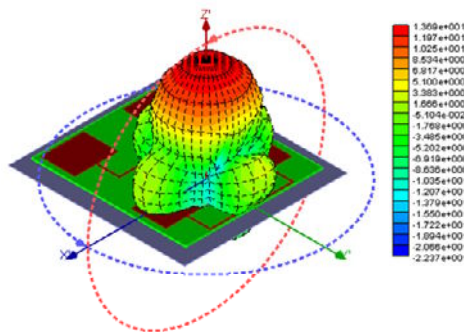


Figure 4.52 The 3-D far-field pattern of the patch antenna array at 2.3 GHz together with the antenna array model.

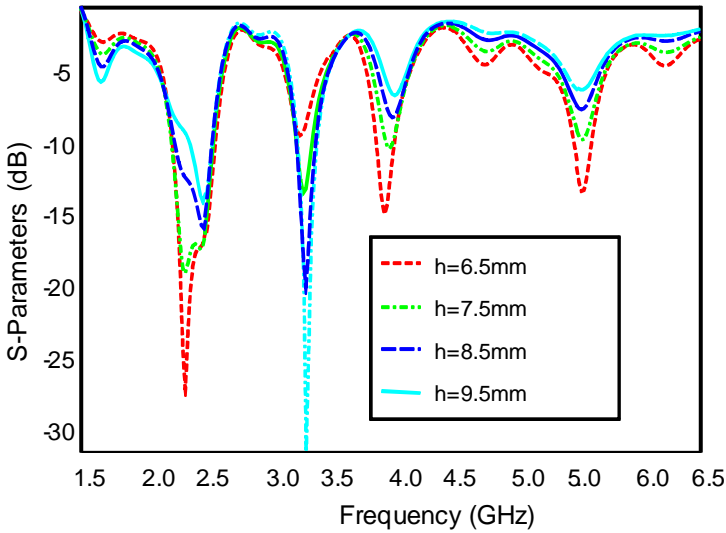


Figure 4.53 Variation of the S-parameters with the distance between the patches and ground plane.

4.11 PERIODIC STRUCTURE SIMULATION

A periodic structure [8, 9] has a configuration shown in Figure 4.54. Its element includes four layers of dielectrics and two open rings, as shown in Figure 4.55. The dimension of the open ring is indicated in Figure 4.56. The distance between two open rings is 0.5 mm, as shown in Figure 4.57. We need to calculate the reflection and transmission coefficients when a plane wave illuminates from the z -direction with E_y polarization, as shown in Figure 4.58.



Figure 4.54 Configuration of the periodic structure in the two directions. Each element contains a pair of open PEC rings.

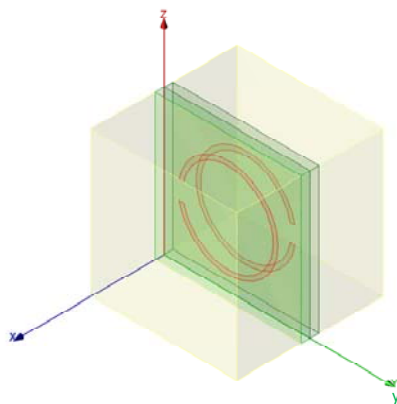


Figure 4.55 Element pattern of the periodic structure including a pair of open PEC rings.

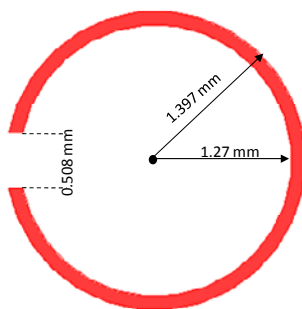


Figure 4.56 Open ring dimensions buried inside the periodic structure.

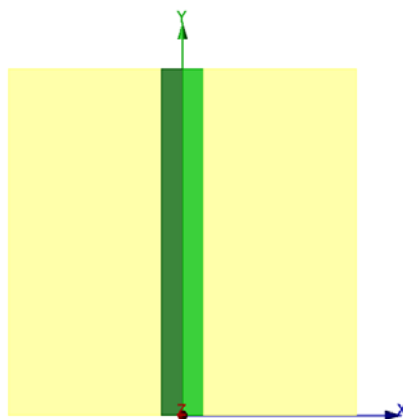


Figure 4.57 Element configuration and dimensions of the periodic element and the parameters of the dielectric layer.

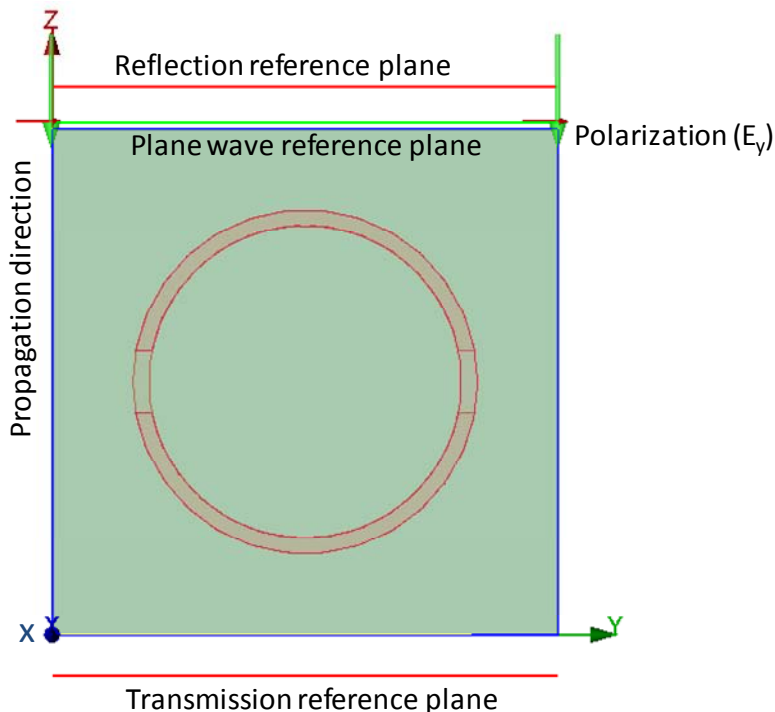


Figure 4.58 Relative position of the reference planes of the plane wave, reflection and transmission measurements.

Since the periodic structure is in the x - and y -directions, we only need to simulate one element in the X - Y plane and the PBC (periodic boundary) is applied to truncate the four directions in the X - Y plane. The domain size should be equal to the element size, and there is no white space in the four directions in the X - Y plane. The absorbing boundary is applied to truncate the two vertical directions (say, along the z -direction). There are six cells between the absorbing boundary and the periodic structure. The plane wave reference is located between the absorbing boundary and the periodic structure. The reflection coefficient reference plane is located above the plane wave reference plane, and the transmission coefficient reference plane is located below the periodic structure.

In this example, the width of the ring is 1.27 mm. Following the general rule, we select the cell size to be 0.0635 mm, namely, two cells across the ring width. The resonant frequency occurs about 10.2 GHz. When the cell size decreases to 0.05 mm, the resonant frequency moves to 10.5 GHz. However, when the cell size decreases to 0.025 mm (see Figure 4.59), the resonant frequency is 11 GHz, as shown in Figure 4.60. However, when further decreasing the cell size, the resonant frequency does not change any more.

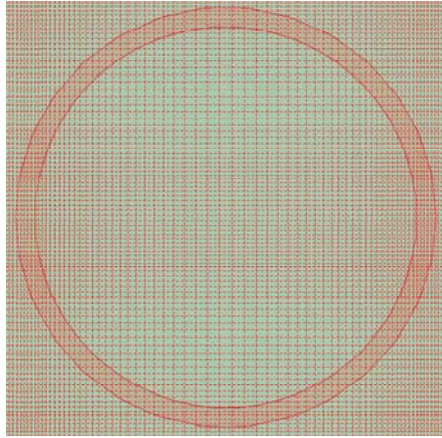


Figure 4.59 Local mesh distribution in the open ring area and the fine mesh is used in the open gap.

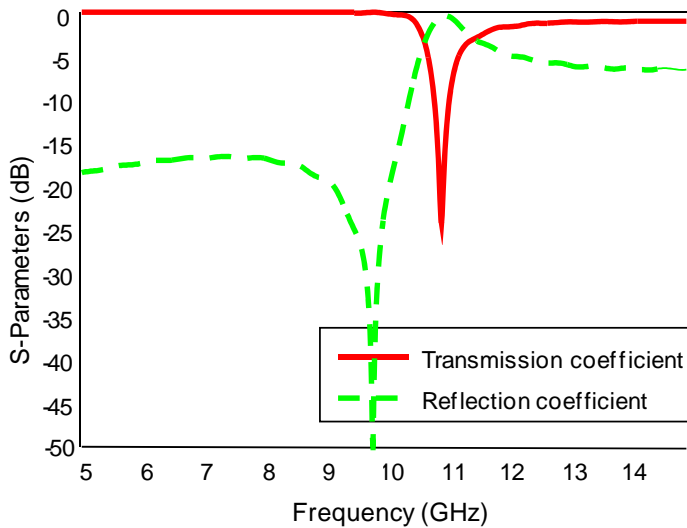


Figure 4.60 Variation of the reflection and transmission coefficients with the frequencies.

4.12 GROUND PENETRATING RADAR MODEL

A ground penetrating radar (GPR) [10–12] model is shown in Figure 4.61. A pair of antennas are used as the transmitter and receiver, respectively. The bow-tie antenna is installed inside the PEC box without the bottom surface. The four corners of the bow-tie antenna connect to the PEC box through four 200 resistor-ohms, as shown in Figure 4.62.

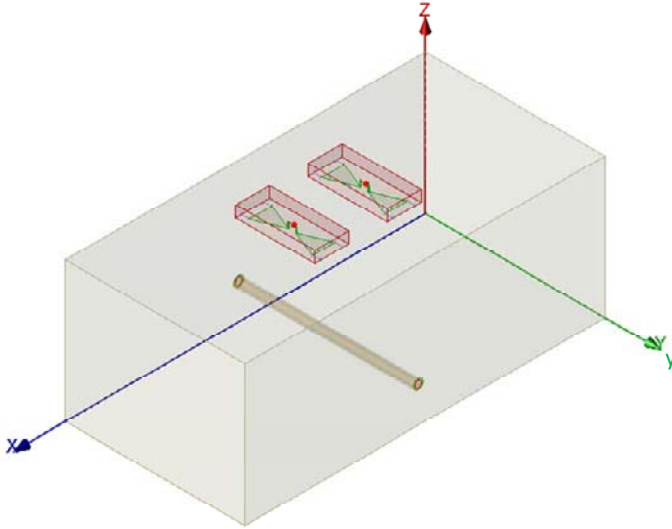


Figure 4.61 GPR model above the ground soil with the buried water pipe.

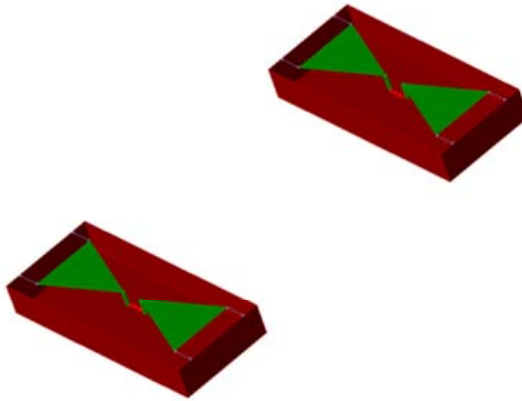


Figure 4.62 Wideband bow-tie antennas that are mounted inside the PEC box.

The ground soil is extended to infinity in the horizontal directions and in the $-z$ -direction as well. The absorbing boundary is applied to truncate the x -, y -, and $-z$ -directions and there is no white space between the absorbing boundary and ground soil in the x -, y -, and $-z$ -directions. The absorbing boundary is used to truncate the z -direction and the white space with six cells in between the absorbing boundary and bow-tie antenna. This example is different from other examples in this chapter because the soil is a dispersive medium, as shown in Figure 4.63. Therefore, the absorbing boundary is able to handle the dispersive medium. In this example, we need to extract the reflected signal by the buried pipe that includes the plastic shell and fill-in water, as shown in Figure 4.63.

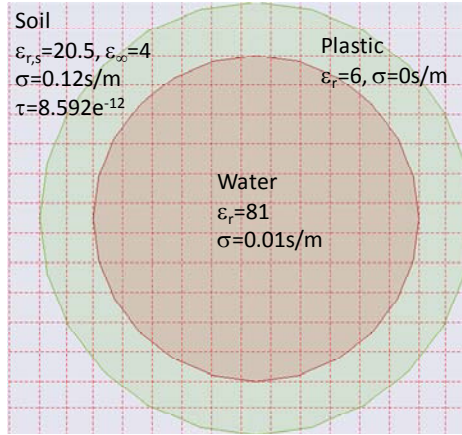


Figure 4.63 Buried pipe structure and material parameters of the soil, plastic pipe, and water inside the plastic pipe.

The length and thickness of the feed structure between two elements of the bow-tie antenna are not important for this problem; however, their selections will affect the simulation time and memory usage. To be smart, we select the thickness of the feed structure to be rectangular and its cross section to be 2×2 cells regardless of the cell size.

To generate the field distribution on a specified plane, we need to draw a surface on which we like to generate the field, as shown in Figure 4.64. For this example, the interesting output is the scattered field from the buried pipe. To this end, we need to simulate the problem two times; in the first time, we remove the buried pipe from the domain and measure the time-domain signal at the receiver. In the second simulation, we simulate the original problem and measure the time-domain signal at the receiver. The subtract of the two signals is the scattered signal from the buried pipe, as shown in Figure 4.65.

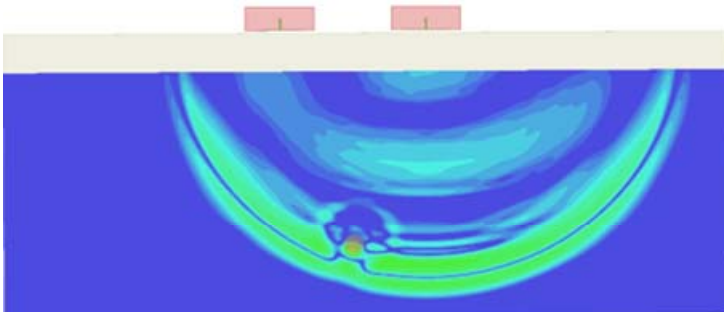


Figure 4.64 Normalized electric field distribution in one cross section at a specified moment.

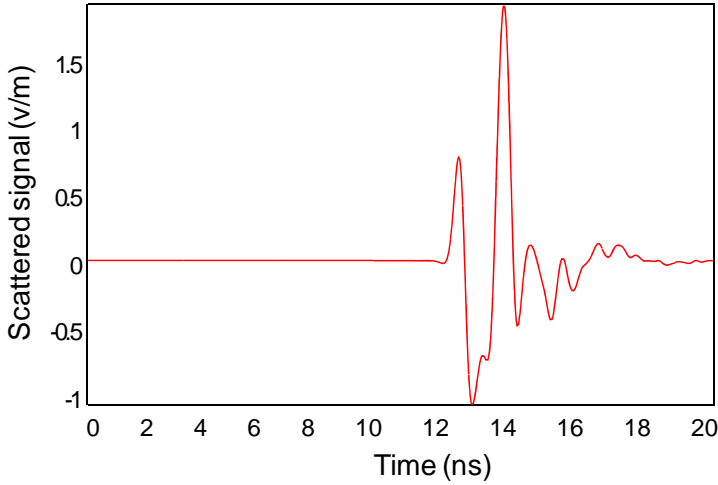


Figure 4.65 Scattered time-domain signature at the receiving antenna when another is excited.

4.13 MICROWAVE CONNECTOR

We finally apply the parallel conformal FDTD to simulate a microwave connector [6], as shown in Figure 4.66(a). The connector includes a cylindrical dielectric structure with six small holes. The coaxial feed is excited by using a TEM mode, as shown in Figure 4.66(b). This is a transformation structure, and the two ports are extended to infinity. We apply the absorbing boundary to truncate the two ports without any white space between the connector and absorbing boundary. The PEC boundary is used to truncate the other four walls of the computational domain without any white space.

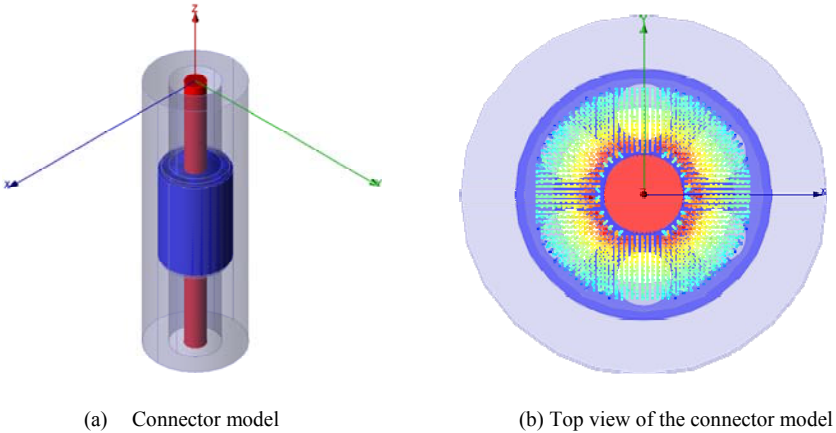


Figure 4.66 (a) Microwave connector configuration and (b) the dielectric fill.

We notice that the connector has fine features (0.329 mm) between the small holes and the inner conductor. A small gap in the model, 0.01095 mm, is located between the inner conductor and the filled dielectric, which does not affect the result. We also notice that the geometry with the minimum dimension is not parallel to either the x - or y -axis, and we can obtain accurate results if we use a minimum cell size in the range of 0.1 mm to 0.1645 mm in the x - and y -directions. Since there is a small gap (0.2 mm) in the z -direction, we use 0.2 mm as the minimum cell size in that direction, as shown in Figure 4.67.

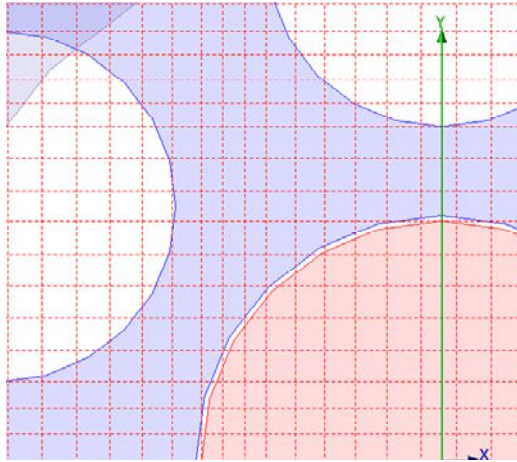


Figure 4.67 Local mesh distribution to describe the fine structures.

If the output parameters include the TDR or time impedance, we must use the narrow pure Gaussian pulse as the excitation pulse to ensure no overlapping of the incident pulse and reflected signal.

If we are only interested in the time-domain reflectometry (TDR) and time-domain impedance of a connector, we can stop the simulation when the excitation pulse goes through the connector. However, to get the accurate frequency response, we need to wait until the simulation is convergent.

The inner conductor of the connector is gold, whereas the outer conductor is PEC. The reflection coefficient for this conductor is computed using the FDTD code and the results are plotted in Figure 4.68. Even when we run the simulation up to a frequency of 50 GHz, we find that the return loss is lower than -27 dB over the entire frequency range.

The incident pulse we have chosen is a pure Gaussian pulse with a 3-dB beamwidth frequency of 65 GHz, which corresponds to a rise time of 0.538 ps ($=0.35/65\text{e}9$), and a time-step resolution of 0.27 ps. The TDR [13, 14] response is simulated by using the time-domain signal, and the result is shown in Figure 4.69. The time-domain impedance, computed by using the reflected and incident signals, is shown in Figure 4.70.

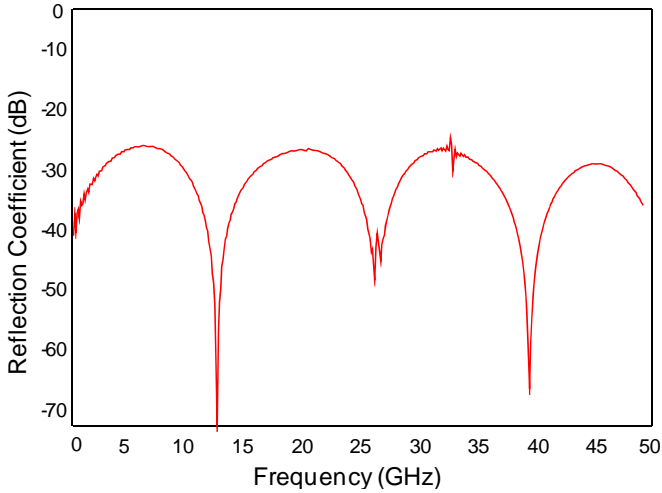


Figure 4.68 Reflection coefficient of the microwave connector when one side is excited.

There are different ways that we can calculate the TDR response. One possibility is to use a narrow Gaussian pulse as the incident wave, measure the total time-domain signal at the excitation port, and then separate the incident and reflected signals from the total time-domain signal. The TDR response can then be calculated by using the incident and reflected time-domain signals. If we use this method, we do not have to wait until the signal converges to derive the TDR response. The TDR can also be calculated by using the frequency-domain reflection coefficient via Fourier transformation.

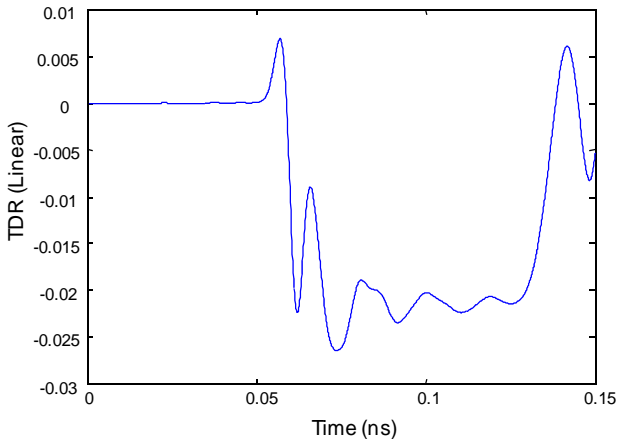


Figure 4.69 Time-domain reflectometry when a narrow pure Gaussian pulse propagates through the connector.

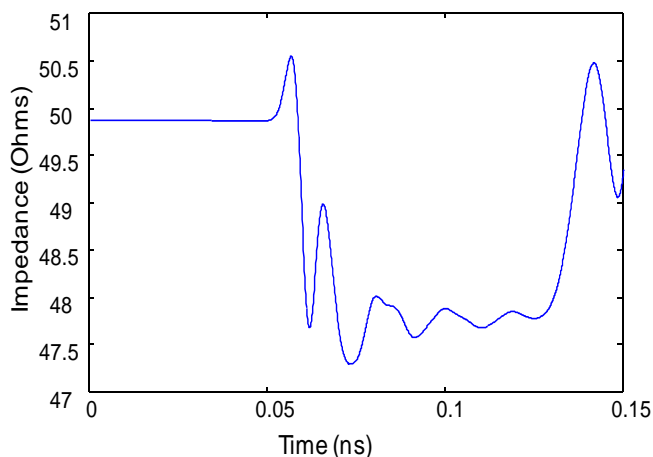


Figure 4.70 Time-domain impedance when a narrow pure Gaussian pulse propagates through the connector.

References

- [1] W. Yu, et al., *Parallel Finite-Difference Time-Domain Method*, Norwood, MA: Artech House, 2006.
- [2] W. Yu, et al., "High Performance Conformal FDTD Techniques," *IEEE Microwave Magazine*, Vol. 11, No. 4, June 2010, pp. 42–55.
- [3] T. Su, et al., "A New Conformal Mesh Generating Technique for Conformal Finite-Difference Time-Domain (CFDTD) Method," *IEEE Antenna and Propagation Magazine*, Vol. 46, No. 1, January 2004, pp. 37–49.
- [4] C. Balanis, *Antenna Theory: Analysis and Design*, 3rd ed., New York: Wiley-Interscience, 2005.
- [5] "The 2000 CAD Benchmark Unveiled," *Microwave Engineering Europe*, November 2000.
- [6] D. M. Pozar, *Microwave Engineering*, 2nd ed., New York: John Wiley & Sons, 1998.
- [7] W. Yu, et al., "New Development of Parallel Conformal FDTD Method in Computational Electromagnetic Engineering," *IEEE Antennas and Propagation Magazine*, Vol. 53, No. 3, 2011.
- [8] B. Munk, *Frequency Selective Surfaces: Theory and Design*, New York: Wiley-Interscience, 2000.
- [9] P. Maagt, R. Gonzalo, and J. Vardzoglou, "Review of Electromagnetic Bandgap Technology and Applications," <http://antennas.unavarra.es/Publicaciones/Images/Pub79.pdf>.
- [10] G. Bourgeois and G. Smith, "A Full Three Dimensional Simulation of a Ground Penetrating Radar: FDTD Theory Compared with Experiment," *IEEE Transactions on Geosciences and Remote Sensing*, Vol. 34, No. 1, 1996, pp. 36–44.
- [11] J. Daniels, D. Gunton, and H. Scott, "Introduction to Subsurface Radar," *Proc. IEE*, Vol. 135, No. 4, 1988, pp. 278–320, [Antennas.unavarra.es/publicaciones/images/pub79.pdf](http://antennas.unavarra.es/publicaciones/images/pub79.pdf).

- [12] L. Peter and J. Young, "Application of Subsurface Transient Radar," *Time-Domain Measurements in Electromagnetics*, New York: Van Nostrand Reinhold, 1986.
- [13] Tektronix Application note, http://www2.tek.com/cmsreplive/tirep/3846/55W_14601_2_2008.09.04.12.32.15_3846_EN.pdf.
- [14] Tektronix Application note, http://www2.tek.com/cmsreplive/tirep/4873/85W_19888_0_LowRes_2010.04.15.17.32.48_4873_EN.pdf.

Chapter 5

EM Simulation Software Benchmarks

In this chapter, we use four of the most popular commercial software packages to simulate some typical problems; all can be simulated using a regular PC installed with 2-GB memory. In order to present a complete simulation idea, we analyze and discuss the most important steps for each example based on our best knowledge. We give GEMS simulation more detailed explanation because we are familiar with GEMS. To be fair to other software packages, note that they do not use any hardware acceleration including the GPU and VALU.

5.1 BASIC STEPS IN EM SIMULATION

In this section, we introduce the basic steps the users should follow when they use the popular commercial software packages for EM problem solving. The procedure summarized here may not be optimized, and hence it is only used to demonstrate the project design and procedure in different software packages. Once again, the interpolation for each software package is based on our best knowledge; please refer to the manufacturer's manual or Web site for any further questions and detailed information. Next, we explain the main features in each software package.

5.1.1 HFSS

High Frequency Simulation Solver (HFSS) [1] (<http://www.ansoft.com>) is developed based on the finite element method (FEM). The basic simulation steps are described as follows:

1. Select a project unit.
2. Generate the project model or import the project model through the CAD files, and then specify the material type to each part.
3. Specify the boundary condition.
4. Generate port objects, and then assign the port property.
5. Provide the frequency band of interest and frequency step size.
6. Validate project to check the project settings.
7. Simulate the project.
8. Process and generate results.

The main features include:

1. Three-dimensional modeling is developed based on ACIS that supports the most popular CAD file formats.
2. Parametric design and optimization.
3. Adaptive mesh generation.

5.1.2 CST

Computer Simulation Tool (CST) [2] (<http://www.cst.com>) was developed based on the finite integration technique (FIT). The basic simulation steps are described as follows:

1. Select a project unit.
2. Select the background properties.
3. Generate the project model or import the project model through the CAD files, and then specify the material type to each part.
4. Set the frequency band of interest.
5. Specify the boundary condition.
6. Create excitation port.
7. Set output parameters.
8. Specify the number of cells per wavelength, and check the mesh distribution and see if the fine structures are properly captured.
9. Select solver and specify the convergence criterion.
10. Simulate the project.
11. View the simulation results.

The main features include:

1. Three-dimensional modeling is developed based on ACIS.
2. Parametric design and optimization.
3. Adaptive mesh generation.

5.1.3 FEKO

FEKO [3] (<http://www.feko.info>) was developed based on the method of moments (MoM). The basic simulation steps are described as follows:

1. Select a project unit.
2. Generate the project model or import the project model through the CAD files, and then specify the material type to each part.
3. Create a port that cannot be located on the dielectric surface.
4. Assign the port property.
5. Specify the frequency band of interest.
6. Set up the output parameters.
7. Design the mesh distribution.
8. Simulate the project.

9. View the simulation results.

The main features include:

1. Three-dimensional modeling is developed based on PARASOLID.
2. Parametric design and optimization.
3. Adaptive mesh generation.

5.1.4 GEMS

GEMS [4] (<http://www.2comu.com>) was developed based on parallel conformal FDTD method, which can be used to simulate the extremely large problems using the computer cluster at a relatively high parallel efficiency. Since FDTD is a time-domain technique, we can obtain a broadband frequency response in one GEMS simulation. The conformal technique allows us to simulate the curved surfaces using a relatively large cell size. In GEMS simulation, we need to provide the highest frequency of interest, minimum cell size that usually equals half of the fine structure of interest and the ratio between adjacent cells, number of processors for the parallel version. The basic simulation steps are described as follows:

1. Select a project unit.
2. Create a new GEMS project.
3. Generate the project model in GEMS interface or import the project model through the CAD files in the SAT, STEP, IGES, ProE, Catia, and DXF, and then specify the material type to each part.
4. Set the excitation source, and output options.
5. Specify the highest frequency.
6. Specify the boundary condition.
7. Provide the minimum cell sizes.
8. Setup the output parameters for the far-field patterns.
9. Design the processor distribution for the parallel processing.
10. Specify the convergence criterion and check the project settings.
11. Simulate the project on a PC through GEMS solver or on a cluster through GEMS WBI.
12. Open GEMS display window to view the simulation results and derive the results through the GEMS data postprocessing center.

The main features include:

1. Three-dimensional modeling is developed based on ACIS that supports the most popular CAD file formats including DXF.
2. Parametric design and optimization.
3. Adaptive mesh generation.
4. Support 32-bit and 64-bit Windows and Linux platforms.
5. Support MPI and distributed computing techniques.
6. Support IBM BlueGene platform.

7. Support Web job submission, job status checking, and result view.
8. Provide the combination of parallel software and hardware system.

5.2 HARDWARE PLATFORMS

In order to be fair to the other four software packages, when we compare the simulation time, memory usage, and the accuracy of HFSS, CST, FEKO, and GEMS, all the numerical experiments are carried out on the same hardware platform, and its configuration is listed here:

Lenovo ThinkPad T400

CPU: Intel dual core 2.4 GHz (L2 Cache: 3 MB)

Memory: 2 GB (PC3-850 DDR3 1067 MHz)

Operating system: 32-bit Microsoft Windows XP



Since the basic version of GEMS supports the multicore processor without any extra cost, we also use the quad core processor to give readers an idea of the performance of GEMS on the GEMS workstation. The configuration of the GEMS workstation is listed here:

EM simulation workstation

CPU: Intel quad core 2.4 GHz (L2 Cache: 6 MB)

Memory: 8 GB (DDR2-667 800 MHz)

Operating system: GEMS Linux/Windows



Since the GEMS workstation is specially optimized for the GEMS simulation software, the excellent performance of the GEMS workstation is evident compared to the other software and systems.

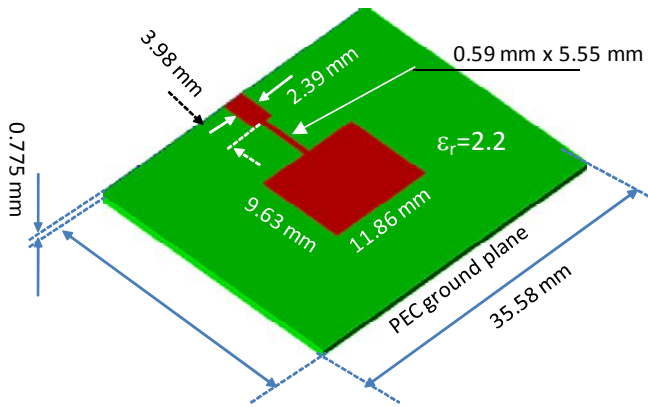
5.3 PATCH ANTENNA

The patch antenna includes a rectangular patch, one-quarter match line, and a feed line, as shown in Figure 5.1. The dimensions of the antenna and material are indicated in Table 5.1. The output parameters will be the return loss. Since the far-field pattern is not sensitive to the approximation in the EM simulation, we only compare the return loss in this example.

Although both the ground plane and substrate are finite, this antenna is still a relatively simple problem. The basic parameter settings in different software packages and performance are summarized in Table 5.2.

Table 5.1 Dimensions and Material of Components in the Antenna


Components	Dimensions (mm)	Material
Substrate	28.89×35.58×0.775	$\epsilon_r=2.2$, $\sigma=0$
Patch	9.63×11.86	PEC
Matched line	0.59×5.65	PEC
Feed line	2.39×3.98	PEC
Ground plane	28.89×35.58	PEC

**Figure 5.1** Patch antenna configuration and the dimensions of patch, feed structure, and substrate.

Once again, all the numerical experiments are carried out based on the same hardware platform. During the simulation, we have noticed that the two cores of the dual core processor were used for the simulation in all four software packages. Unlike CST, HFSS and FEKO, GEMS can realize the communication between the compute cores through either OpenMP or MPI in a single processor. OpenMP is the default option in GEMS, and the operating system will organize the jobs between the cores automatically regardless of the number of cores in the computer. During the simulation, we can observe two applications of “GEMS_Solver.exe.” One of “GEMS_Solver.exe” uses all cores to solve the electric and magnetic fields and second one is used for the job management to reach a good code performance. In contrast, MPI option in GEMS will automatically break the whole domain into two pieces of subdomains for a dual core processor and four pieces of subdomains for a quad core processor, and each core only simulates its own subdomain. Except for the simple geometry problems, the MPI option will significantly improve the simulation performance. When we use the MPI option, during the simulation we will observe that the number of applications of “GEMS_Solver.exe” is equal to 3 for the dual processor and 5 for the quad core processor. Two “GEMS_Solver.exe”


in the dual core processor are used for field solving, and one for the job management. Four “GEMS_Solver.exe” in the quad core processor are used for field solving, and one for the job management. Besides “GEMS_Solver.exe,” we can see one more application during GEMS simulation, “Solver.exe,” which is the process for the GEMS solver window. Unlike HFSS, CST, and FEKO, GEMS does not need to keep the designer open during the GEMS simulation, therefore the memory requirement of the GEMS designer should not be counted into the GEMS simulation. It is worthwhile to mention that the graphical interface of HFSS, CST, and FEKO may use a large amount of memory for the complex geometries.

Table 5.2 Simulation Summary Using Different Software Packages

					
	HFSS	CST	GEMS	FEKO	
Cell size	N/A	$\lambda/42$; mesh line ratio limit: 10	$\Delta x = 0.44$ mm; $\Delta y = 0.44$ mm; $\Delta z = 0.4$ mm; ratio: 1	Edge length = 2; segment length = 0.775; wire radius = 0.01	
Number of unknowns	11,872 tetrahedrons	168,300 cells (110×90×17)	178,542 cells (109×91×18)	5,937 basis functions	
Boundary condition	Radiation (61×54×26)	Open (add space)	Open boundary (add space)	N/A	
Simulation time	332 sec.	154 sec.	89 sec.	1,155 sec.	
Environment memory (MB)	156	158 (designer: 75, modeler: 83)	23 (controller: 11, solver: 12)	104 (cadfeko)	
Simulation memory (MB)	207	Matrix: 100, solver: 76	28.6	270.9	
Total memory usage	363 MB	>258 MB	54 MB	374.9 MB	
Time steps	N/A	4,900	2,698	N/A	
Convergence criterion	$\Delta s = 0.011$	−30 dB	−30 dB	N/A	

For the same problem and number of unknowns, we use the GEMS workstation to simulate it, and the performance is demonstrated in Table 5.3.

Table 5.3 GEMS Performance on the Workstation

	Configuration	Operating System	Problem Size	Simulation Time
Workstation	Intel Q6600 quad core 2.4 GHz	Linux	178,542 cells (109×91×18)	21 sec.

GEMS can detect the number of cores in the current computer and break the problem domain into the same number of subdomains and then apply the MPI on it automatically. We can see more “GEMS_Solver.exe” applications in the Task Manager window. Each GEMS MPI option will require about 10 MB of memory excluding the memory for the field solving.

Besides the GEMS solver engine, GEMS environment memory requirement totals about 25 MB, including about 11 Mb for the GEMS controller and 14 MB for GEMS simulation window.

Although HFSS allows the user to close the model during the simulation, it does not allow the user to close the designer window. HFSS designer without model loading has a memory usage of about 101 MB.

Both CST and FEKO do not allow the user to close the designer window and project model during the simulation. Hence, the memory usage of their designer depends on the problem size and complexity. For this simple case, CST designer and modeler require 135 MB memory, and FEKO designer requires 104 MB memory.

The mesh design in FEKO strongly depends on the user’s experience. Here, to avoid involving the detailed mesh design, we adjust the global mesh parameters to reduce the number of unknowns to compare with other three software packages. Further reducing the number of cells will result in an inaccurate result. Since this structure is rather simple, we do not need to get involved in the mesh design in other three software packages. We adjusted the mesh in GEMS to match the number of cells used in CST because CST and GEMS use the similar method. The simulation results using different software packages are plotted in Figure 5.2. Though we do not show the far-field pattern here, they match very well from different software packages.

It is a little difficult for us to compare the time spent on the far-field pattern calculation. The reason is that the far-field pattern is calculated during the simulation in GEMS, CST, and FEKO, but it is calculated in the data postprocessing in HFSS.

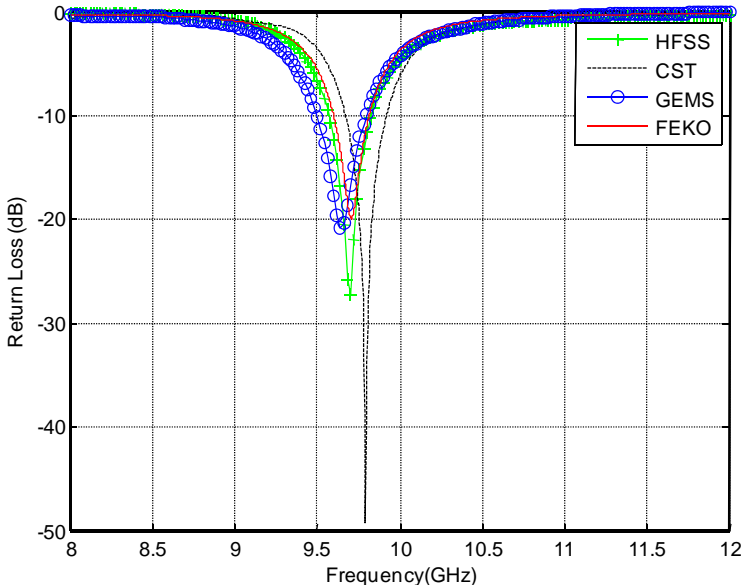


Figure 5.2 Return loss of the patch antenna using different software packages.

It is a little difficult for us to compare the time spent on the far-field pattern calculation. The reason is that the far-field pattern is calculated during the simulation in GEMS, CST, and FEKO, but it is calculated in the data postprocessing in HFSS.

The modeling procedure in different software packages:

1. HFSS:

- (a) Import the SAT model.
- (b) Specify the material type to ground, antenna, and substrate.
- (c) The users decide the object overlapping relationship.

2. CST:

- (a) Import the SAT model.
- (b) Specify the material type to ground, antenna, and substrate.
- (c) The users decide the object overlapping relationship.

3. FEKO:

- (a) Import the SAT model.
- (b) Specify the material type to ground, antenna, and substrate.
- (c) The users decide the object overlapping relationship.

4. GEMS:

- (a) Import the SAT model.
- (b) Specify the material type to ground, antenna, and substrate.
- (c) Follow the order of nature; namely, the latter inputted objects will erase the previous ones in the overlapping region. Input of the infinitely thin patch and ground must follow the substrate.

The mesh design procedure in different software packages:

1. HFSS:

- (a) The user provides the frequency range of interest.
- (b) HFSS generates adaptive mesh.

2. CST:

- (a) The user provides the frequency and number of cells per wavelength and the maximum mesh ratio.
- (b) CST generates adaptive mesh, but the user needs to check the mesh distribution.

3. FEKO:

- (a) The user provides the mesh size including edge length, segment length, and wire radius.
- (b) FEKO generates adaptive mesh.

4. GEMS:

- (a) The user provides the minimum structure of interest and a mesh ratio.
- (b) GEMS generates adaptive mesh.

The excitation setting in different software packages:

1. HFSS:

- (a) Create an excitation port in two steps.
- (b) Specify the port property.

2. CST:

- (a) Create a port.
- (b) Assign the port property.

3. FEKO:

- (a) Dig a hole in the substrate, where the excitation source is located.
- (b) Create a port.
- (c) Assign the port property.

4. GEMS:

- (a) Draw a line from the ground to the signal line.
- (b) Assign this line as an excitation port.

The output parameters in different software packages:

1. HFSS:

- (a) Process the output parameters in the data postprocessing.
- (b) View the output parameters.

2. CST:

- (a) Preset the output parameters.
- (b) View the output parameters.

3. FEKO:

- (a) Preset the output parameters.
- (b) View the output parameters.

4. GEMS:

- (a) Preset the output parameters.
- (b) View the output parameters.
- (c) Use the formula library to generate the user-defined parameters in the data postprocessing.

We have a chance to compare the performance of GPU and CPU for this simple case, namely, the case does not require the far-field pattern, dispersive medium, and 3-D field or current distributions, which will be not efficient for GPU. The output parameter is the return loss only. It means that the GPU only carries out the field update and absorbing boundary (PML). The GPU data is provided by Professor Atef Elsherbeni of the University of Mississippi (atef@olemiss.edu). The problem domain is discretized into 105,924 ($97 \times 80 \times 14$) cells for both GPU and CPU (GEMS) codes. The simulation information is summarized in Table 5.4, in which the simulation time is for 2,700 time steps.

Table 5.4 Performance Comparison of GPU and CPU

Hardware Platform	Simulation Time (Sec.)	
	Windows XP System	GEMS Linux System
Intel Core 2 dual 2.4 GHz	61	
Intel Q6700 2.66 GHz	32	
Intel Q6600 2.4 GHz		21
Two Intel Q6600 2.4 GHz		12
NVIDIA 7500M	21	
NVIDIA 8800 GTS	10	
NVIDIA 280 GTX	6.4	

Summary: The ground plane and substrate of the patch antenna are finite and four software packages can simulate it accurately. The software performance strongly depends on the user's experience, and the readers should simulate it using the different software packages and draw their own conclusions.

5.4 VIVALDI ANTENNA

The Vivaldi antenna [5], as shown in Figure 5.3, is a good benchmark for the commercial EM simulation software packages. There are some publications on this topic, but these benchmarks only show the simulation result and brief summary without the detailed simulation procedure, therefore, the readers cannot repeat the results by themselves. In this part, we will present detailed simulation information for the different software packages. Due to the similarity of GEMS and CST, we first use CST to simulate the Vivaldi antenna and then use GEMS for the same number of cells. FEKO and HFSS are based on the frequency-domain techniques, so we try to use their default input parameters to simulate the Vivaldi antenna.

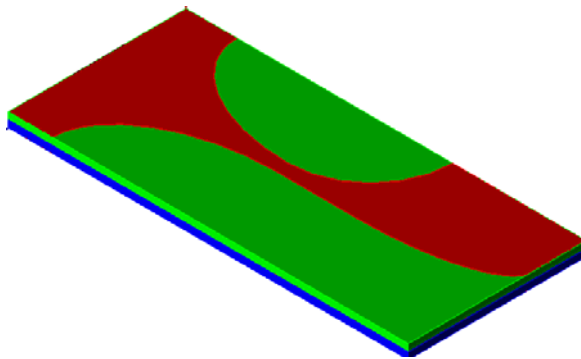


Figure 5.3 Configuration of the Vivaldi antenna that includes three Vivaldi structures.

The dimensions of the Vivaldi antenna are indicated in Figure 5.4. We can simulate half of the antenna taking the benefit of the symmetric structure about the central element. However, if the superstrate and substrate have a different dielectric constant, we cannot use the symmetric property. Hence, here we simulate the whole structure directly. Since four software packages provide the features to handle the symmetric structure, therefore the simulation time will be reduced to the half if we simulate the half structure. Readers can generate the Vivaldi structure based on the information in Figure 5.4, or can find the SAT model for this Vivaldi antenna online.

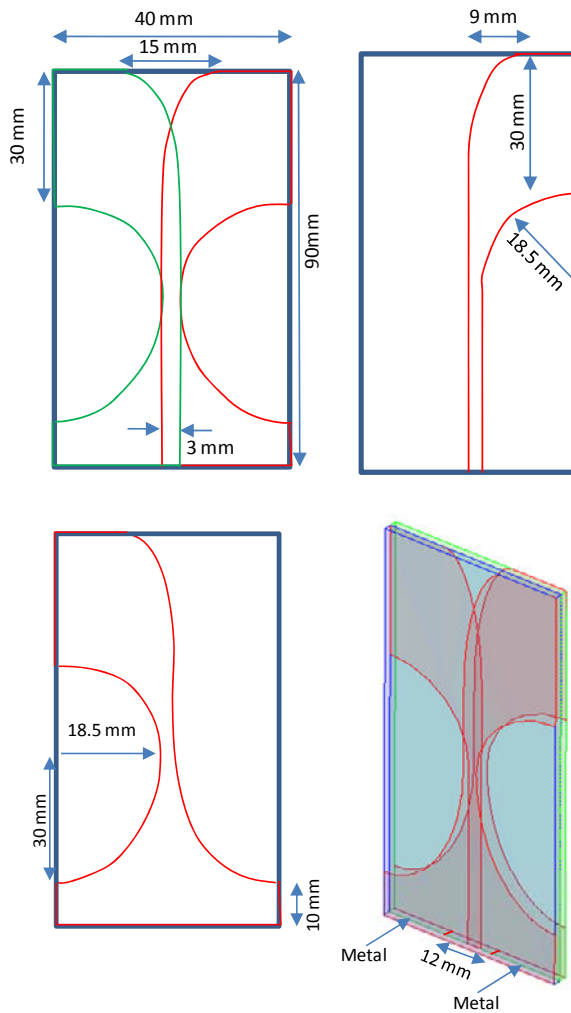


Figure 5.4 Structure and dimensions of the Vivaldi antenna.

This Vivaldi antenna has a special feed structure, as shown in Figure 5.5, so we need to find a way to excite the antenna in different software packages.

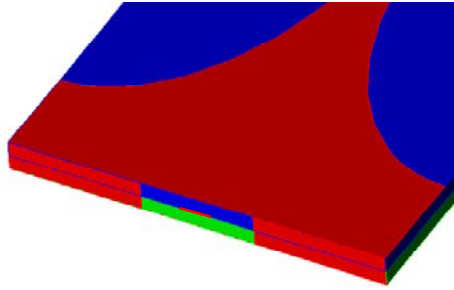


Figure 5.5 Feed port configuration of the Vivaldi antenna.

When we calculate the return loss or S-parameter, the excitation port must be terminated by a matched load. There are several ways to terminate the port of Vivaldi antenna: (1) extend the feed structure to a coaxial cable, (2) let the feed port touch the absorbing boundary, and (3) add a terminator at the feed port.

1. Extend the feed structure to a coaxial cable, as shown in Figure 5.6. Four software packages provide this technique to extend the feed structure to a coaxial cable. However, it is a relatively complicated procedure.
 - (a) Draw a block that has the same cross section as the feed port.
 - (b) Select the four surfaces to form a hollow guide, or directly draw four surfaces that are used as the walls of guide.
 - (c) Draw a dielectric block (superstrate) to extend the upper dielectric into the guide.

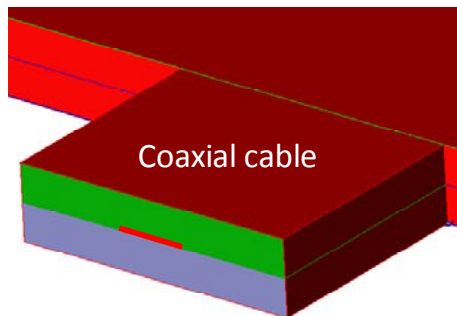


Figure 5.6 Extended rectangular coaxial cable port of the Vivaldi antenna.

- (d) Draw a dielectric block (substrate) to extend the lower dielectric into the guide.
- (e) Draw a strip line to extend the strip into the guide.
- (f) Specify the guide, strip line and dielectric parameters.

We can use the TEM mode to excite the feed guide port. In addition, the guide structure will increase the burden of the simulation.

2. Let the feed port touch the absorbing boundary, as shown in Figure 5.7. We can let the feed port touch the absorbing boundary of the computational domain to extend the port structure to infinity. In this case, we do not need to do any extra work on the model, and directly let the antenna port touch the absorbing boundary. Then, we can use a regular wave port to excite the port to calculate the return loss. There is a big problem when we need to calculate the far-field pattern because we cannot have information in the feed side. Though we did not try this model, we believe both GEMS and CST have this feature.

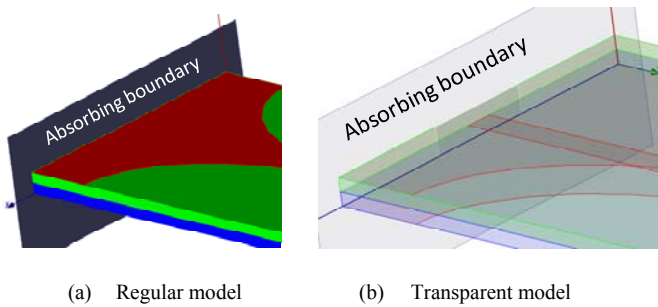


Figure 5.7 Excitation port touching the boundary (a) regular model and (b) transparent model.

3. Add a terminator at the feed port, as shown in Figure 5.8.

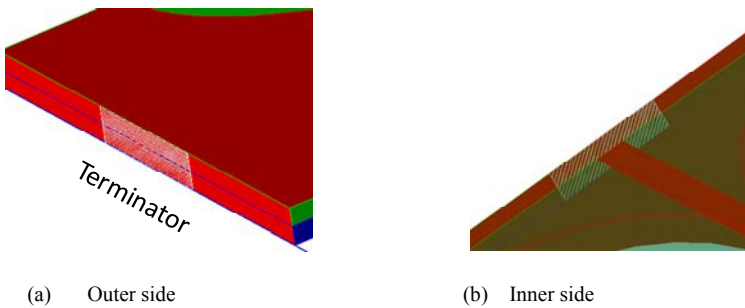


Figure 5.8 Excitation structure of the Vivaldi antenna terminated (a) outer side and (b) inner side.

It is a relatively easy job to add this matched load.

- (a) Draw rectangle at the feed port.
- (b) Assign it to be a matched load.

In this feed model, we can use a regular wave port to excite the port, as shown in Figure 5.9. A six-layer CPML is applied to truncate all six walls of the computational domain in GEMS simulation.

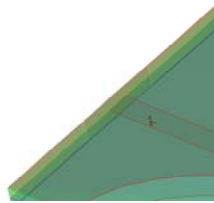


Figure 5.9 One of the port excitation ways used in the GEMS software.

The simulation results are summarized in Figure 5.10. To get the matched result from HFSS, we tried to use different feed ports, and different boundary conditions, and different sizes of the domain, however, we could not match the result. The results from HFSS we did get are plotted in Figure 5.11. We observed from Figure 5.11 that the results from different settings are consistent. We tried to use FEKO to simulate this antenna; however, we could not get a result that is close to those from GEMS, CST, and HFSS.

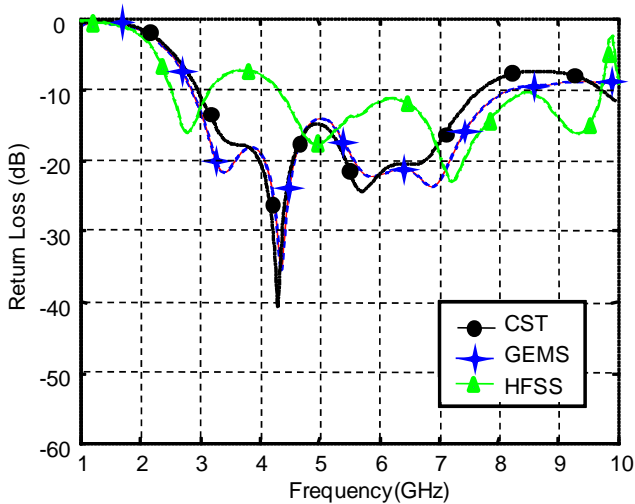


Figure 5.10 Return loss of the Vivaldi antenna using different software packages.

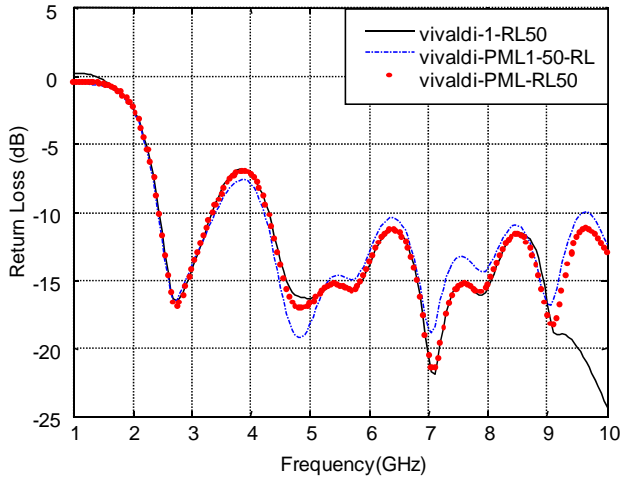


Figure 5.11 Return loss obtained by using HFSS for the different boundary conditions.

This Vivaldi antenna was simulated by using HFSS and published in Microwave Engineering Online “The 2000 CAD Benchmark Unveiled,” as shown in Figure 5.12. However, we would not repeat this result using different feed options in HFSS.

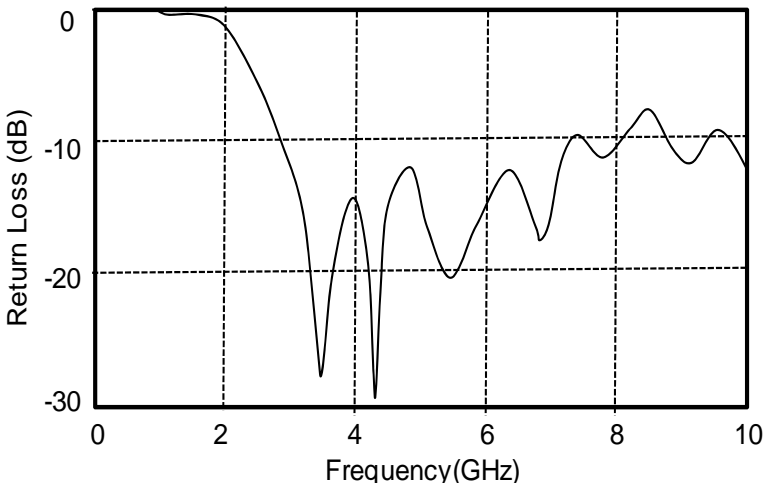


Figure 5.12 Return loss in the literature obtained by using HFSS.

The project settings and performance in each software package are given in the Table 5.6. Different software provides the memory usage and simulation time in different ways. Here, we show how we got the numbers in Table 5.6.

GEMS:**1. Memory usage:**

- (a) GEMS solver (GEMS engine);
- (b) GEMS Controller (for MPI or OpenMP);
- (c) Solver (GEMS solver window).

The memory usage above has the same life span. The first one may vary during simulation. The memory for the second and third processes is much smaller than the first and is almost constant.

2. Simulation time:

- (a) Preprocessing: mesh and material generation;
- (b) GEMS engine for the field solving;
- (c) Far-field pattern calculation (optional).

The simulation time for each process is shown inside the simulation window. The time in the time bar may be longer than the summation of each process inside the window because the MPI process launching time may not be included.

3. Cell size setting policy:

- (a) Input half of the minimum dimension of problem structure.
- (b) Inputs the mesh ratio.
- (c) It is not necessary to check the mesh distribution.

CST:**1. Memory usage:**

- (a) Matrix_cal (varies during simulation);
- (b) Solver (varies during simulation);
- (c) Modeler (exist during the simulation);
- (d) CST Design Environment (exists during the simulation).

The memory for (a) and (b) may not reach the peak value at the same time.

2. Simulation time:

- (a) Matrix calculation;
 - (b) Solver;
 - (c) Others.
3. Cell size setting policy:
- (a) Input the number of mesh lines per wave length;
 - (b) Inputs the mesh line ratio;
 - (c) It is necessary to check the mesh distribution to avoid the ill-conditioned mesh.

HFSS:


1. Memory usage:
- (a) HFSS.exe (environment);
 - (b) HFSScomengine (solver);
- The memory reading is for the solver only.
2. Simulation time:
- The simulation is reported in the summary file.
3. Cell size setting policy:
- Adaptive mesh.

FEKO:

1. Memory usage:
- (a) Peak memory (read from the simulation summary file);
 - (b) Cad_feko (environment).
- We are not sure if the peak memory includes the environment memory.
2. Simulation time:
- Read from the simulation summary file.
3. Cell size setting policy:


- (a) Edge length;
- (b) Wire segment length;
- (c) Wire radius.

Table 5.6 Simulation Summary of the Vivaldi Antenna Simulation

	HFSS	CST	GEMS
Cell Size	N/A	0–10 GHz: $\lambda/21$, mesh line ratio limit: 10	$\Delta x=0.85$ mm; $\Delta y=0.85$ mm; $\Delta z=1$ mm; Ratio: 1
Number of unknowns	65,183 tetrahedrons	114,240 cells (60×119×16)	119,040 cells (62×120×16)
Boundary condition	Radiation (190×170×150)	Open (add space)	Open boundary (add space)
Simulation time	5 h, 8 min., 55 sec.	100 sec.	59 sec.
Simulation memory (MB)	1,320	Matrix: 113.8 Solver: 56.9	24.7
Environment memory	174.7 MB	139 MB	25.6 MB
Total memory usage	1490 MB	>252.8 MB	50.3 MB
Time steps	N/A	1,650	1,798
Convergence criterion	0.006	–30 dB	–30 dB

For the same problem and the same number of unknowns, we use GEMS workstation to simulate it and the simulation is summarized in Table 5.7.

Table 5.7 GEMS Performance on the Workstation

	Configuration	Operating System	Problem Size	Simulation Time
Workstation	Intel Q6600 quad core 2.4 GHz	Linux	119,040 cells (62×120×16)	24 sec.

Summary: Due to the special feed structure of the Vivaldi antenna, it is important to find a correct way to excite the Vivaldi antenna in different software packages. The software performance strongly depends on the user's experience, and the readers should use their own methods to excite the Vivaldi antenna in the different software packages and draw their own conclusions.

5.5 SCATTERING OF DIELECTRIC SPHERE

In this section, we present one example for the 3-D radar cross section (RCS) calculation. A dielectric sphere (see Figure 5.13, its radius=30 mm, relative dielectric constant = 4) is illuminated by a plane wave along $\theta=0^\circ$. The output parameter is the bistatic RCS at the frequency = 9.368 GHz (wavelength in free space = 32 mm, and 16 mm in dielectric). The cell size is selected to be 0.75 mm.

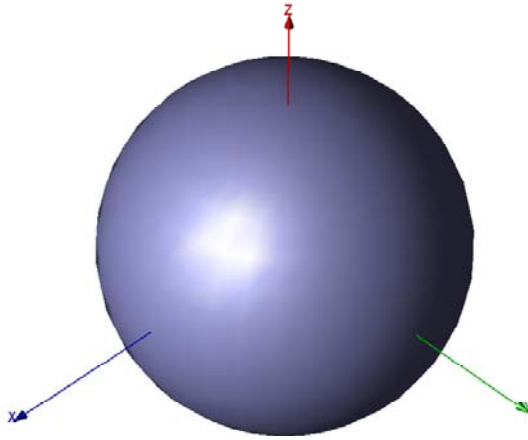


Figure 5.13 A dielectric sphere with the dielectric constant = 4.

We picked the dielectric sphere as a benchmark because it is simple and has the MIE solution [6] so that we can check if the simulation result is accurate. Since there is no fine structure inside the computational domain, we can select the cell size in the time-domain method based on the highest frequency of interest. However, we need to take the dielectric constant 4 into account. We select the input parameters as follows in different software packages:

HFSS:

1. Cell sizes: $\Delta x = \Delta y = \Delta z = 0.75$ mm.
2. Boundary: radiation boundary, one-half wavelength white space.
3. Excitation: Plane wave source.
4. Convergence criterion: 0.042.

5. Outputs: RCS (process in the postprocessing).

CST:

1. Min. mesh step: 0.71; Max. mesh step: 1.5.
2. Boundary: Open (add space).
3. Number of cells: $52 \times 52 \times 52$.
4. Excitation: Plane wave source.
5. Convergence criterion: -40 dB.
6. Outputs: RCS (process in the postprocessing).

FEKO:

1. Edge length = 1.5 mm, which is $\lambda/10$ inside the sphere at the 9.368 GHz.
2. Excitation: Plane wave source.
3. Output: RCS (direct output parameter).

GEMS:

1. Cell sizes: $\Delta x = \Delta y = \Delta z = 0.75$ mm.
2. Boundary: 6-layer CPML.
3. Number of cells: $52 \times 52 \times 52$ (not include PML layers).
4. Excitation: Plane wave source.
5. Output: RCS (direct output parameter).

The 3-D far-field pattern calculated using GEMS is plotted in Figure 5.14. Though we cannot compare the difference of 3-D far-field pattern, from their shape and level we know they are in good agreement from the different software packages. GEMS and CST can generate both RCS (dB) and RCS (dBsm); however, we cannot find a way to generate the RCS (dB) in both FEKO and HFSS when we do the data postprocessing.

The simulation results are plotted in Figures 5.15 and 5.16. We have observed from Figures 5.15 and 5.16 that the bistatic RCS of the dielectric sphere using different software is almost the same except for the HFSS results. We tried to change the domain size in HFSS, and though the result is improved when the domain gets larger, its result is still not close to that obtained by using the other three software packages.

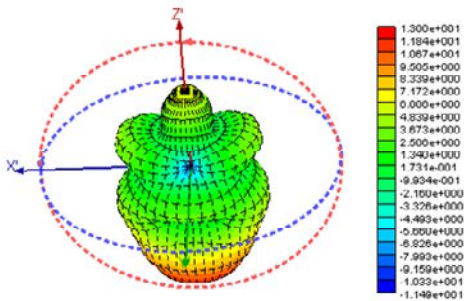


Figure 5.14 The 3-D RCS pattern of the dielectric sphere at $f=9.368$ GHz.

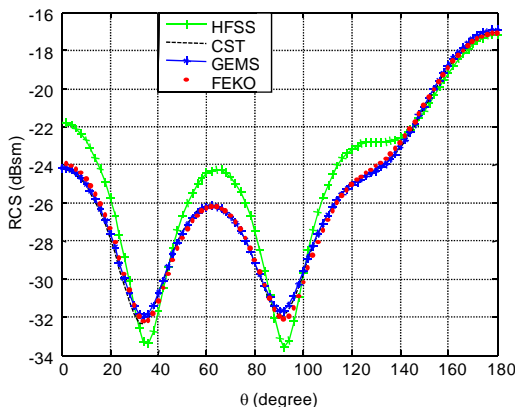


Figure 5.15 Bistatic RCS versus spatial angle in $\phi=0^\circ$ plane at 9.368 GHz.

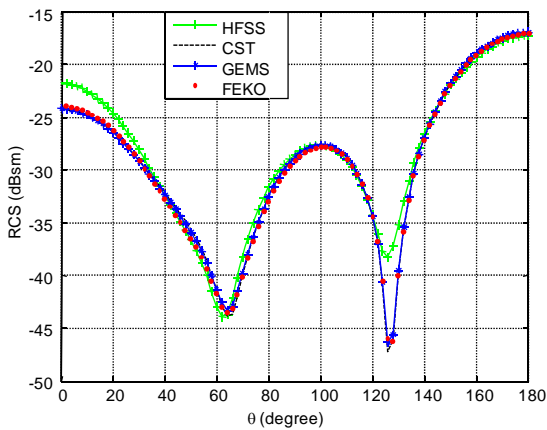


Figure 5.16 Bistatic RCS versus spatial angle in $\phi=90^\circ$ plane at 9.368 GHz.

HFSS does not require the users to select the mesh size for a given problem. We picked up the mesh size in FEKO, CST, and GEMS based on our best understanding. For example, in CST and GEMS, the cell size is selected to be 0.75 mm (about 20 cells per wavelength). For the sake of comparison, we intentionally use the same number of cells in CST and GEMS. For the dielectric cavity problem the field inside the cavity attenuates very slowly. For the antenna problems, the far-field pattern is not sensitive to the near-field and current distribution inside the domain. However, in this example, -30 dB convergence criterion may not be sufficient to reach the convergence result in both CST and GEMS. We may need to decrease the criterion to achieve a good result, as shown in Figures 5.17 and 5.18.

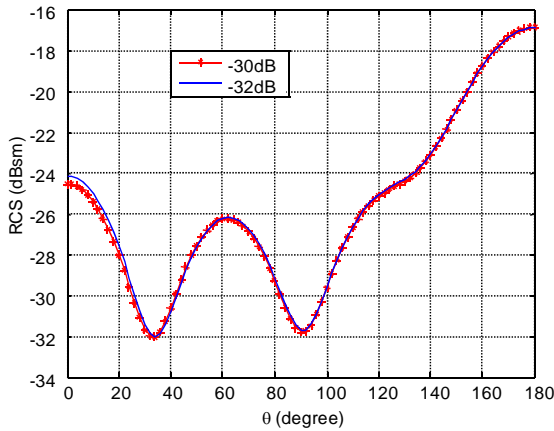


Figure 5.17 Bistatic RCS variation with convergence criterion in $\phi=0^\circ$ plane at 9.368 GHz.

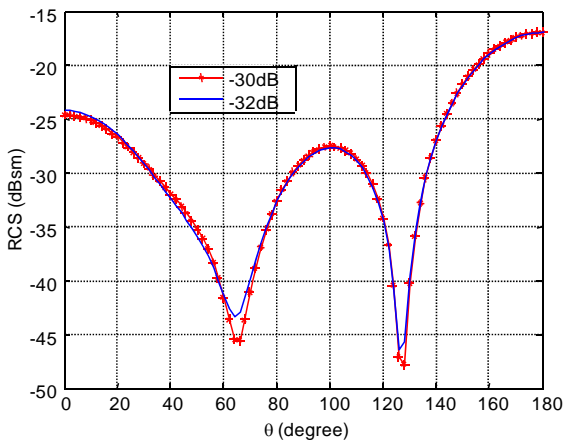


Figure 5.18 Bistatic RCS variation with convergence criterion in $\phi=90^\circ$ plane at 9.368 GHz.

In the frequency-domain method such as MoM and FEM, the convergence criterion is forced on each frequency, therefore, the simulation result will reach the convergent value when the convergence criterion is satisfied. In contrast, the convergence criterion is on a wide frequency band, the convergence criterion is a statistical concept. Even if the convergence criterion is reached, it does not mean that the result is convergent for each individual frequency. In fact, it may depend on the shape of the excitation pulse.

The same thing may happen in CST, the RCS difference between a -30 dB and a -40 dB convergence criterion is shown in Figures 5.19 and 5.20.

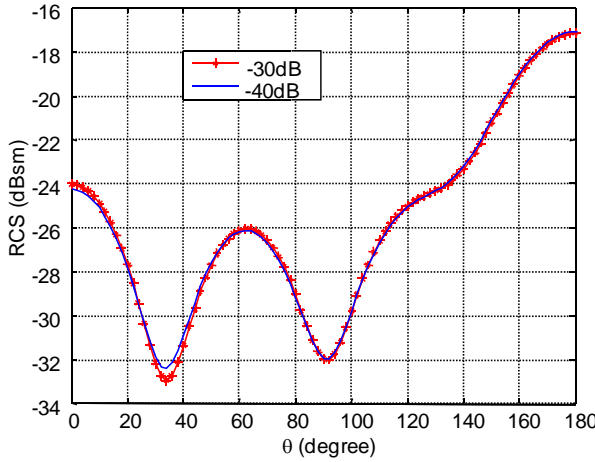


Figure 5.19 Bistatic RCS variation with convergence criterion in $\varphi=0^\circ$ plane at 9.368 GHz.

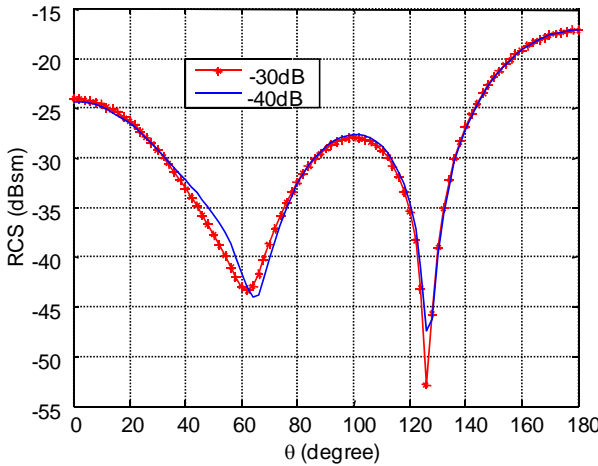


Figure 5.20 Bistatic RCS variation with convergence criterion in $\varphi=90^\circ$ plane at 9.368 GHz.

Each software package will generate a simulation report that includes the simulation information such as simulation time, memory usage, preprocessing time, and computer status. It is hard to compare the resource usage for different methods. Here, we show the typical report of each software package. Besides the simulation information, GEMS also lists the computer name and available memory in the project report. The simulation information is summarized in Table 5.8.

GEMS:

The memory usage of GEMS includes the memory for GEMS solver engine, GEMS controller, and GEMS solver window. When GEMS is running, we can read the memory usage for each module from the Windows Task Manager. The memory usage for the GEMS solver window does not vary with the problem size. The memory usage for the GEMS controller may change a little bit with the problem options such as problem size excitation and output parameters. However, the memory usage for the GEMS solver engine strongly depends on the problem size.

Simulation time of GEMS includes the following parts: (1) GEMS solver engine that includes the field update and far-field pattern calculation; (2) the mesh and material generations, namely, project preprocessing; and (3) the domain decomposition, launching MPI, and so forth. The time spent on this part can usually be ignored. Though we may use the GEMS PC version, GEMS still allows us to use the multiple cores through MPI. For the complex problems, GEMS performance on a multiple core processor can be improved significantly using MPI.

The approach to calculate the far-field pattern is different in the different methods. In MoM, FEM, and FIT, the mesh is generated on the surface of objects; therefore, the electric and magnetic surface currents are located on the object surface. Once we get the surface current distributions inside the domain, we can calculate the far field easily. In the time-domain solvers, we cannot keep all the history of the surface current distribution in each time step, and then calculate the far-field pattern using either DFT or FFT. Instead, we need to take DFT at each time step to count the contribution of the surface current at the each time step. In contrast, GEMS based on the FDTD method will not calculate the surface current distribution on the object surface, but GEMS calculates the equivalent current distributions on the surface of the Huygens' box that encloses the entire antenna structure. The far-field pattern is calculated from this surface current. We can store the frequency-domain surface current on the Huygens' surface and calculate the far field in the data postprocessing, or calculate the far-field pattern following the FDTD update, in which the far field will have direct output parameters. In GEMS, both 2-D and 3-D far fields will have direct output parameters.

GEMS solver memory usage summary:
Available memory in compute nodes:
Host name: wen-laptop , available memory 1146.801 MB
Memory needed in compute nodes:
Host name: wen-laptop
Total memory needed in all processors: 33.8 MB
GEMS solver simulation time summary:
Time for project preprocessing: << 0: 0: 1 >>
Time for initialization: << 0: 0: 0 >>
Time for field update: << 0: 0: 29 >>
Time for frequency-domain output: << 0: 0: 0 >>
Time for far-field calculation: << 0: 0: 5 >>

Total running time: << 0: 0: 35 >>

CST:

Number of mesh cells: 140,608
Excitation duration: 2.369699e-001 ns
Calculation time for excitation: 9 sec.
Number of calculated pulse widths: 10.5292
Steady state accuracy limit: -40 dB
Simulated number of time steps: 1,854
Maximum number of time steps: 176,082
Time step width:
 without subcycles: 1.345790e-003 ns
 used: 1.345790e-003 ns
Number of threads used: 2
Matrix calculation time: 13 sec.
Solver time: 41 sec.
Total time: 54 sec.
Total simulation time: 54 sec.

HFSS:

Task	Real Time	CPU Time	Memory	Information
mesh3d_init	00:00:01	00:00:00	38.5 M	1,537 tetrahedra
wave_1_seed_FT	00:00:01	00:00:01	39.1 M	3,734 tetrahedra
Adaptive Pass 1				
Frequency: 9.368 GHz				
adapt_part1	00:00:02	00:00:01	19.4 M	3,734 tetrahedra
Solver CSS	00:00:09	00:00:09	90.0 M	24,140 matrix
adapt_part2	00:00:01	00:00:00	15.4 M	3,734 tetrahedra

Adaptive Pass 2

Frequency: 9.368 GHz

mesh3d_adapt_FT	00:00:02	00:00:01	39.5	M	4,485	tetrahedra
adapt_part1	00:00:02	00:00:02	20.1	M	4,485	tetrahedra
Solver CSS	00:00:14	00:00:14	117.0	M	28,966	matrix
adapt_part2	00:00:01	00:00:00	17.0	M	4,485	tetrahedra

Adaptive Pass 3

Frequency: 9.368 GHz

mesh3d_adapt_FT	00:00:02	00:00:01	39.7	M	5,385	tetrahedra
adapt_part1	00:00:02	00:00:02	22.3	M	5,385	tetrahedra
Solver CSS	00:00:21	00:00:21	153.0	M	3,4738	matrix
adapt_part2	00:00:01	00:00:01	18.9	M	5,385	tetrahedra

Adaptive Passes converged

Elapsed time 00:01:07

Total 00:00:59 00:00:53


FEKO:

SUMMARY OF REQUIRED TIMES IN SECONDS

	CPU-time	runtime
Reading and constructing the geometry	0.109	0.109
Checking the geometry	0.062	0.063
Initialization of the Greens function	0.000	0.000
Calcul. of coupling for PO/Fock	0.000	0.000
Calcul. of matrix elements	289.110	289.140
Calcul. of right-hand side vector	0.093	0.094
Preconditioning system of linear eqns.	19.047	19.046
Solution of the system of linear eqns.	464.766	464.766
Determination of surface currents	0.000	0.000
Calcul. of impedances/powers/losses	0.015	0.016
Calcul. of averaged SAR values	0.000	0.000
Calcul. of power ideal receiving ant.	0.000	0.000
Calcul. of cable coupling	0.000	0.000
Calcul. of electric near field	0.000	0.000
Calcul. of magnetic near field	0.000	0.000
Calcul. of far field	17.829	17.828
Other	0.594	0.563
Total times:	791.625	791.625
(Total times in hours:	0.220	0.220)

Peak memory usage during the whole solution: 961.734 MByte


Table 5.8 Simulation Summary Using Different Software Packages

	HFSS	CST	GEMS	FEKO
Cell size	N/A	0–15 GHz: $\lambda/14$, mesh ratio limit:10	$\Delta x=0.75$, $\Delta y=0.75$, $\Delta z=0.75$, Ratio: 1	Edge Length: 1.5 mm, $(\lambda/10)$
Number of unknowns	5,385 tetrahedrons	140,608 cells (52×52×52)	140,608 cells (52×52×52)	11,214 basis functions
Boundary condition	Radiation (46×46×46)	Open (add space)	Open boundary (add space)	N/A
Simulation time	59 sec.	54 sec.	44 sec.	792 sec.
Simulation memory	153 MB	82.8 MB	26.4 MB	961.7 MB
Environment memory	160.2 MB	147.1 MB	25.3 MB	104.3 MB
Total memory usage	313.2 MB	>191.9 MB	51.7 MB	1066 MB
Time steps	N/A	1,854	1,649	N/A
Convergence criterion	0.042	–40 dB	–32 dB	N/A

The peak memory of “Matrix calc” and “solver” in CST may not be reached at the same time, therefore, we use the sign “>” to indicate a memory usage range. We select –40 dB in this example, because we cannot select a number between –30 dB and –40 dB in CST.

For the same problem and the same number of unknowns, we use GEMS workstation to simulate it and the performance is evident from Table 5.9.

Table 5.9 GEMS Performance for Dielectric Sphere on the Workstation

	Configuration	Operating System	Problem Size	Simulation Time
Workstation	Intel Q6600 quad core 2.4 GHz	Linux	140,608 cells (52×52×52)	11 sec.

Summary: Since the RCS of a dielectric sphere can be expressed as MIE series, we can compare the simulation result with the analytic solution. Once again, the software performance strongly depends on the user's experience, and the readers should simulate it using the different software packages and draw their own conclusions.

5.6 CELL PHONE ANTENNA

In this section, we present one cell phone antenna, whose configuration [7] is shown in Figures 5.21 and 5.22. The antenna structure includes a PEC ground (it is treated as PEC since it has no thickness in the original design), dielectric (relative permittivity = 4.4), antenna and two feed ports. The antenna model in the SAT format was exported from FEKO. When this SAT model was imported to HFSS, HFSS showed the error message in the dielectric object, and the simulation was terminated. When this SAT model was imported to GEMS, GEMS returned an error message from an ACIS function and the simulation was terminated.

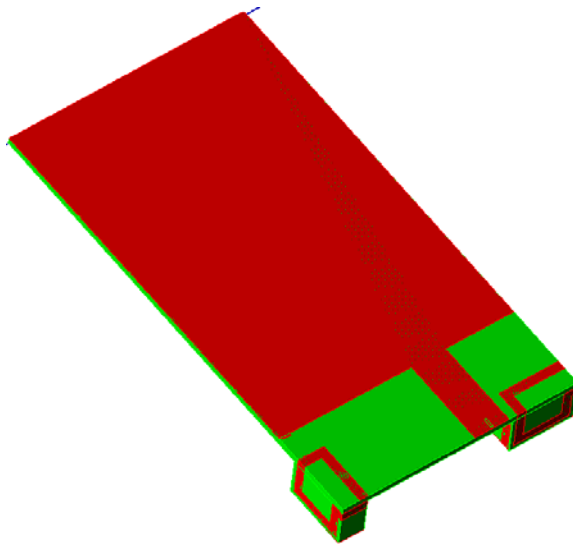


Figure 5.21 MIMO antenna configuration with two antennas at the corners.

Since the model is simple, we manually redrew the dielectric model. HFSS, CST, and GEMS can simulate and generate the correct results.

The small bridge structure of feed strip line and the hole in Figure 5.23 are very inconvenient to all the software packages because it forces us to use a small cell size, and in turn, the time step in the time-domain method will be dramatically reduced. A similar problem will happen using the frequency-domain method.

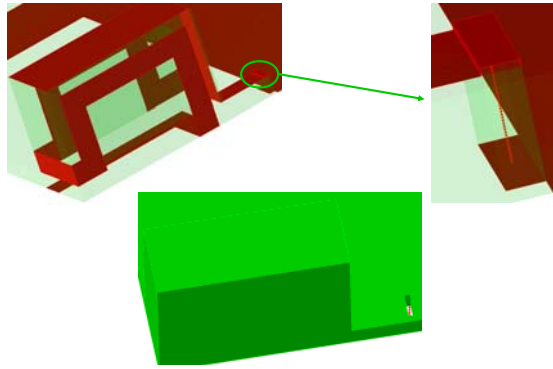


Figure 5.22 Excitation port structure of the MIMO antenna.

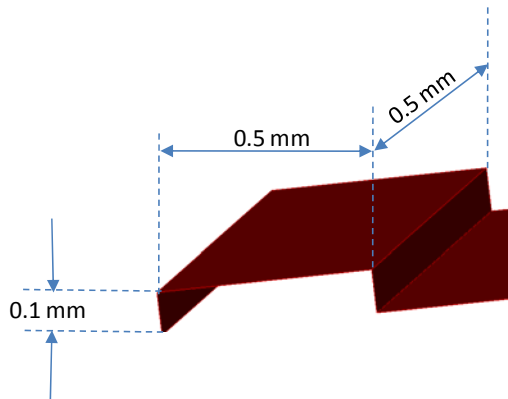


Figure 5.23 Excitation port structure of the MIMO antenna, which is created in FEKO software.

We first use GEMS to simulate this problem. In GEMS, we select the cell size based on the following rules: (1) find the dimension of the minimum structure of interest, for example, the minimum dimensions are 0.5, 0.5, and 0.1 mm in the x -, y - and z -directions; and (2) generally speaking, we need to put two cells in the minimum structure to describe the field variation inside it. In this problem, the dimension in the z -direction is relatively small, and we do not need to describe the field variation in the z -direction. Basically, we can select the cell sizes to be 0.25 mm, 0.25 mm, and 0.1 mm, in the x -, y -, and z -directions, respectively. If the region except this feed structure does not contain many fine structures, we can select the cell size to be up to 60% of the minimum structure. For example, the cell sizes can be 0.3 mm and 0.3 mm in the x - and y -directions, respectively. Both the memory usage and simulation time will reduce by 12% when the cell sizes are reduced from 0.3 mm to 0.25 mm in the x - and y -directions. However, we cannot observe any changes in the S -parameters and far-field patterns.

In this example, we do not need to draw any objects in the EM simulation software since all parts have been included in the SAT model. What we need to do is to specify the material type for each part of SAT model, and set up the excitation and output parameters. The fine structure in this problem is the feed part, and no matter what method is selected, we should ensure the excitation path touches the center point of the square part of the feed structure. This problem can be simulated by using a single PC with 2-GB memory. In the four EM packages, we can import the SAT model into the graphical interface. Next, we introduce the basic procedure when we simulate it using four software packages.

The basic steps using GEMS to simulate this antenna are given here:

1. Import the SAT model into GEMS interface, and specify the material type to the antenna (PEC), ground plane (PEC), and substrate (relative permittivity is 4.4).
2. Draw two lines from the ground to the feed line at the two feed ports and then assign them to be the lumped ports.
3. Specify the highest frequency = 6 GHz in the differential Gaussian pulse.
4. Set the initial cell sizes ($\Delta x = \Delta y = 0.3$ mm, $\Delta z = 0.1$ mm, ratio=1.05) to generate the adaptive mesh.
5. Specify the convergence criterion = -30 dB.
6. Simulate the project on the laptop through GEMS solver.

The basic steps using HFSS to simulate this antenna are given here:

1. Import the SAT model into HFSS interface, and specify the material type to the antenna (PEC), ground plane (PEC), and substrate (relative permittivity is 4.4).
2. Draw two rectangular integration lines from the ground to the feed line at the two feed ports as the lumped ports.
3. Specify the frequency band of interest from 0.5 to 6 GHz, and the frequency step is 0.1 GHz.
4. Set the convergence criterion = 0.02.
5. Simulate the project on the laptop (only one core is used for the simulation).
6. Process the results.

The basic steps using CST to simulate this antenna are given here:

1. Import the SAT model into CST interface, and specify the material type to the antenna (PEC), ground plane (PEC), and substrate (relative permittivity is 4.4).

2. Draw two lines from the ground to the feed line at the two feed ports and then assign them to be the lumped excitation ports.
3. If we specify the highest frequency = 6 GHz and 20 cells per wavelength, the mesh cannot capture the fine feed structure, and hence, the simulation result is not correct. To capture the fine feed structure, we need to specify the highest frequency = 20 GHz.
4. Set the convergence criterion = -30 dB.
5. Simulate the project on the laptop (two cores are used for the simulation).

The basic steps using FEKO to simulate this antenna are given here:

1. Import the SAT model into the FEKO interface, and specify the material type to the antenna (PEC), ground plane (PEC), and substrate (relative permittivity is 4.4).
2. Draw two lines from the ground to the feed line at the two feed ports and then assign them to be the lumped ports.
3. Specify the frequency band of interest from 0.5 to 6 GHz, and the frequency step is 0.1 GHz.
4. We tried FEKO; however, we could not get the correct results.

The simulation results including S_{11} , S_{21} , S_{12} , and S_{22} are plotted in Figures 5.24 to 5.26. We can see from Figures 5.24 and 5.25 that there is a strange behavior at 1.8 GHz in the CST result and strange behavior between 4.5 to 6 GHz in the HFSS result. Other than that, the results generated by three software packages are good and acceptable.

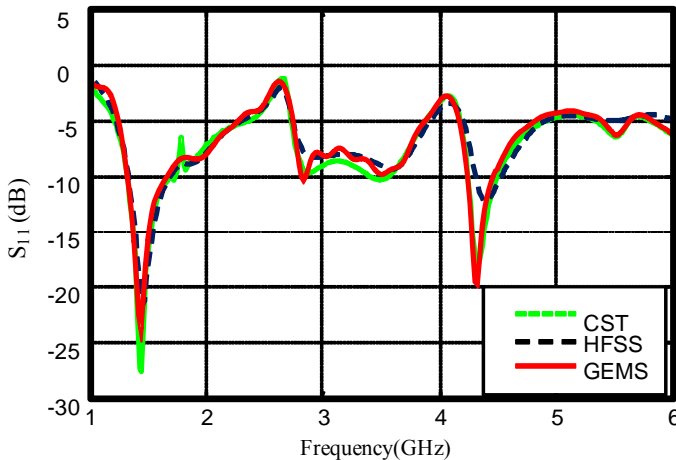


Figure 5.24 S_{11} of the MIMO antenna using different software packages.

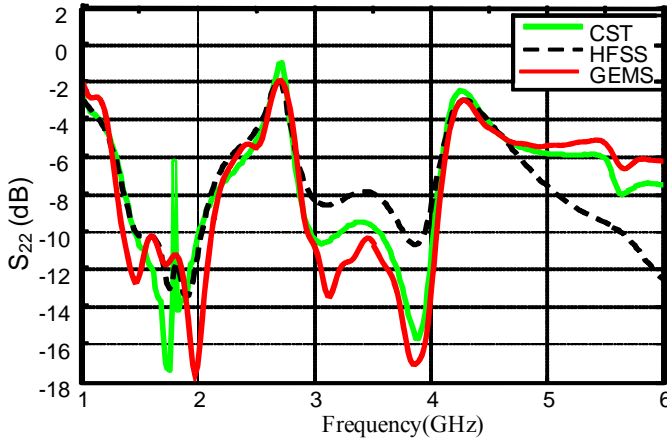


Figure 5.25 S_{22} of the MIMO antenna using different software packages.

The result of CST between 1.6 to 1.9 GHz in Figure 5.25 jumps to about -6 dB and it is not obviously physical. We tried to increase or decrease the cell size and the behavior remained. The S_{22} difference in Figure 5.25 between the different packages may be caused by the feed model that is treated in different ways in different packages.

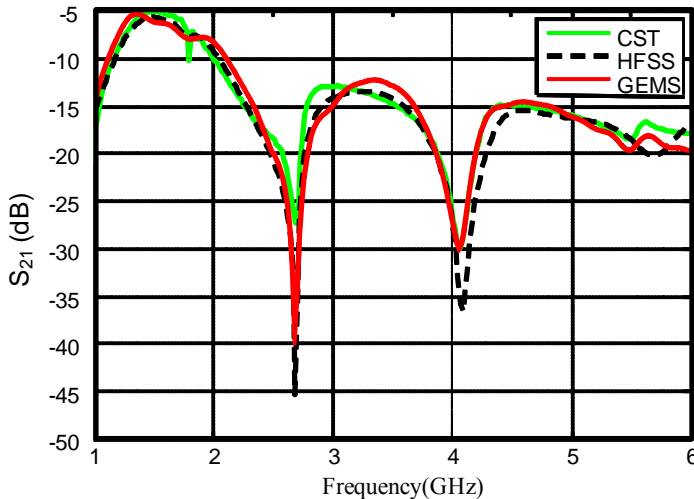



Figure 5.26 S_{12} and S_{21} of the MIMO antenna using different software packages.


The simulation information is summarized in Table 5.10. We always use the same hardware platform when we compare the performance of different software packages. We use the method described before to record the memory usage and simulation time.

Table 5.10 Simulation Summary for Cell Phone Antenna

	HFSS	CST	GEMS
Cell Size	N/A	$\lambda/21$, mesh line ratio limit: 10	$\Delta x=0.3$, Ratio: 1.05 $\Delta y=0.3$, Ratio: 1.02 $\Delta z=0.1$, Ratio: 1.2
Number of unknowns	82,400 tetrahedrons	2,339,901 cells (171×317×43)	2,481,864 cells (168×187×79)
Boundary condition	Radiation (355×400×320 mm)	Open (add space)	Open boundary (add space)
Simulation time	479 min.	294 min.	90 min.
Simulation memory (MB)	1,280 (peak value) (Solver)	Calc.: 381.4 Solver: 226.1	125 (GEMS_solver)
Environment memory (MB)	162 (hfss.exe: 106.3, hfsscomengine.exe: 56.5)	158.1 (modeler: 83.6, CST Design: 74.5)	25.6 (Controller: 11.2) Solver: 14.4
Memory usage (MB)	1,442	>539.5	150.6
Time steps	N/A	52,560	13,012
Convergence criterion	$\Delta s=0.008$	−30 dB	−30 dB

For the same problem and the same number of unknowns, we use GEMS workstation to simulate it and the simulation information is summarized in Table 5.11.

Table 5.11 GEMS Performance on the Workstation

	Configuration	Operating System	Problem Size	Simulation Time
Workstation	Intel Q6600 quad core 2.4 GHz	Linux	2,481,864 cells (168×187×79)	46 min., 6 sec.

When the Cartesian mesh is used in CST, the finite integration technique becomes the FDTD method, and hence we can compare CST and GEMS in memory usage, simulation time, and accuracy. Here, we use this example to investigate the capacity of different software packages to handle the ill-conditioned model. The antenna model in the SAT format was exported from FEKO. When it is imported to HFSS, HFSS shows the error message in the substrate and the simulation stops. When the same model is imported to CST, CST does not show any error message and can simulate it, but its result is obviously wrong. When the same model is imported to GEMS and if the cell size is selected to be 0.1 mm in the x -, y -, and z -directions, the simulation stops due to the ill-conditioned dielectric model. However, if the cell size is selected to be other values (not 0.1 mm, 0.1 mm, and 0.1 mm) such as 0.3 mm, 0.3 mm, and 0.1 mm in the x -, y -, and z -directions, respectively, for the same ill-conditioned SAT model, GEMS can simulate and generate the correct result. The simulation result is summarized in Figure 5.27, the HFSS result is used as a reference, and we can observe that besides the strange behavior in CST at low frequency (<1.3 GHz), the S_{11} and S_{22} have a big deviation from HFSS and GEMS.

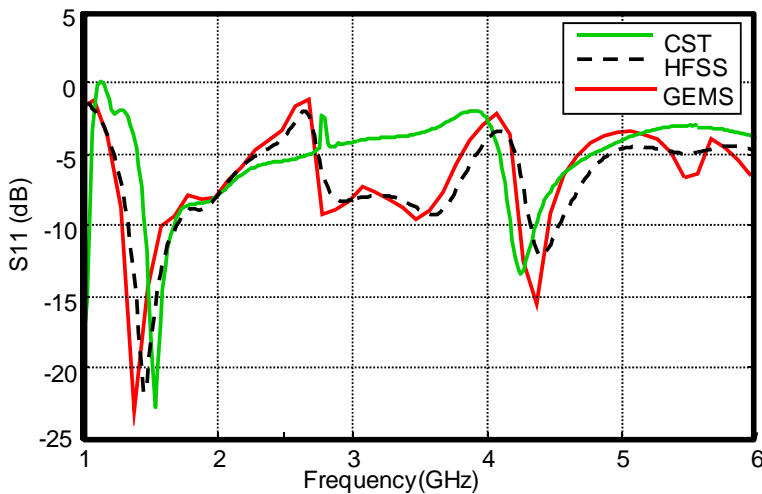


Figure 5.27 S_{11} parameter of the MIMO antenna using different software packages.

We cannot simply compare GEMS and HFSS in terms of the memory usage and simulation time since the methods used in the two software packages are different. If the number of unknowns in HFSS reduces, can we still get a good result? To this end, we reduce the number of unknowns in HFSS to 25,000 tetrahedrons, and the number of cells in GEMS in the similar ratio. At this point, the GEMS simulation time is almost the same as that used in the HFSS simulation. The simulation results using GEMS and HFSS are plotted in Figures 5.28 to 5.32.

We can observe from Figures 5.28 to 5.32 that the GEMS simulation result does not change much when the number of cells is reduced from 2.4 to 7.4 Mcells.

However, the significant variation in the HFSS simulation result is observed when the number of unknowns is reduced to one-third and the HFSS result is not acceptable when the simulation time is reduced the level of GEMS.

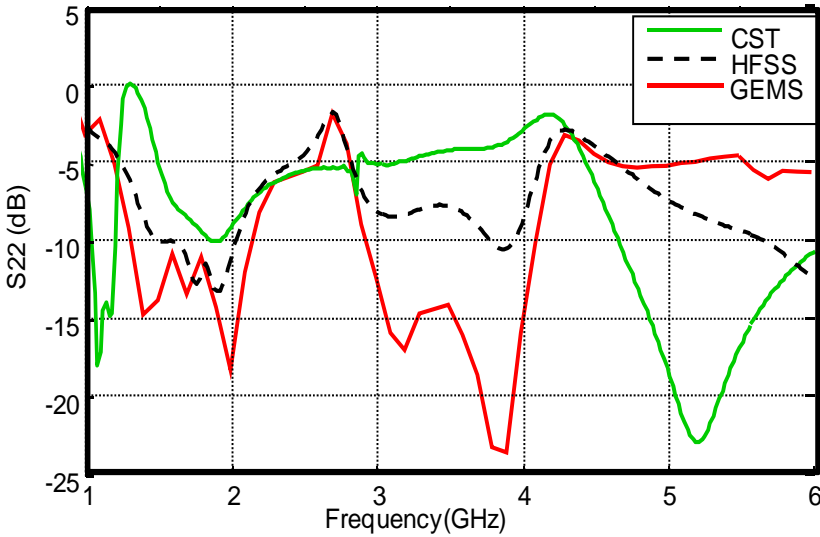


Figure 5.28 S_{22} parameter of the MIMO antenna using different software packages.

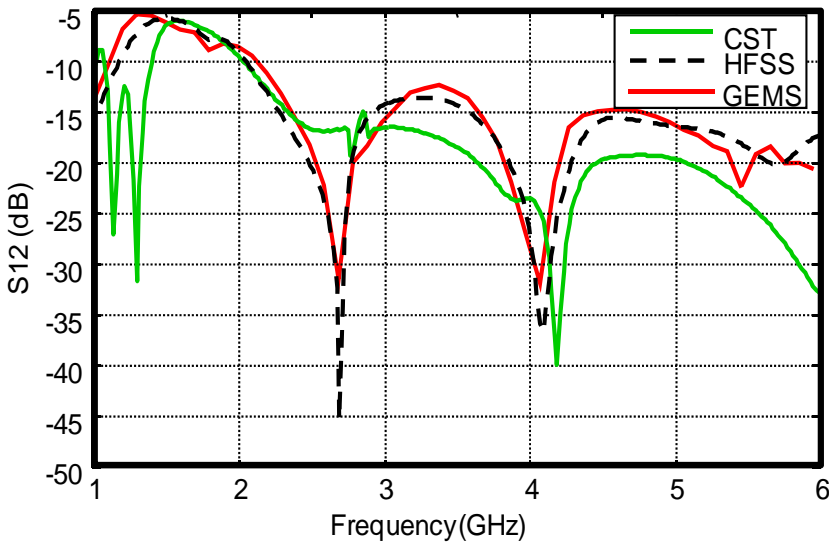


Figure 5.29 S_{21} and S_{12} parameter of the MIMO antenna using the different software packages.

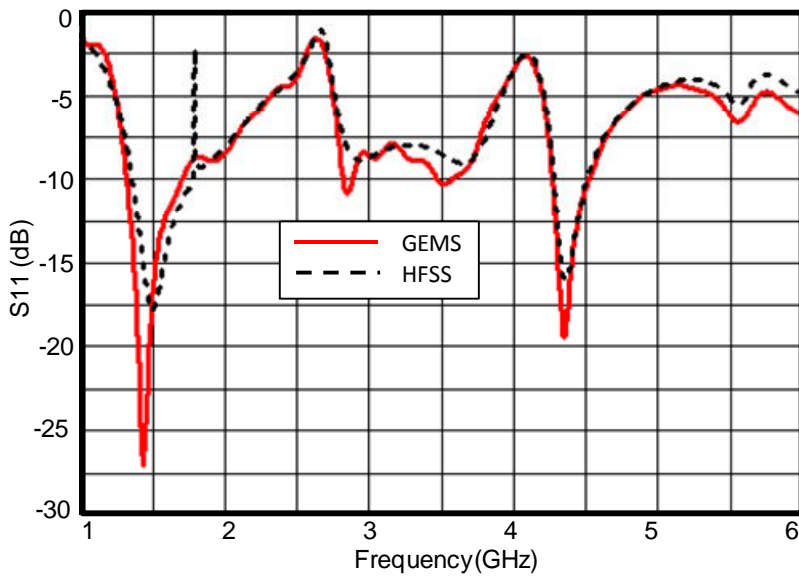


Figure 5.30 S_{11} comparison of the MIMO antenna between HFSS and GEMS.

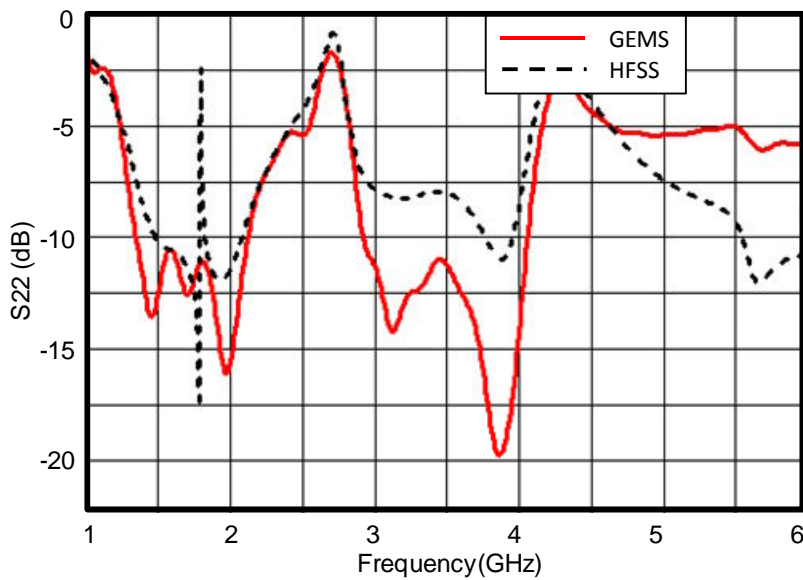


Figure 5.31 S_{22} comparison of the MIMO antenna between HFSS and GEMS.

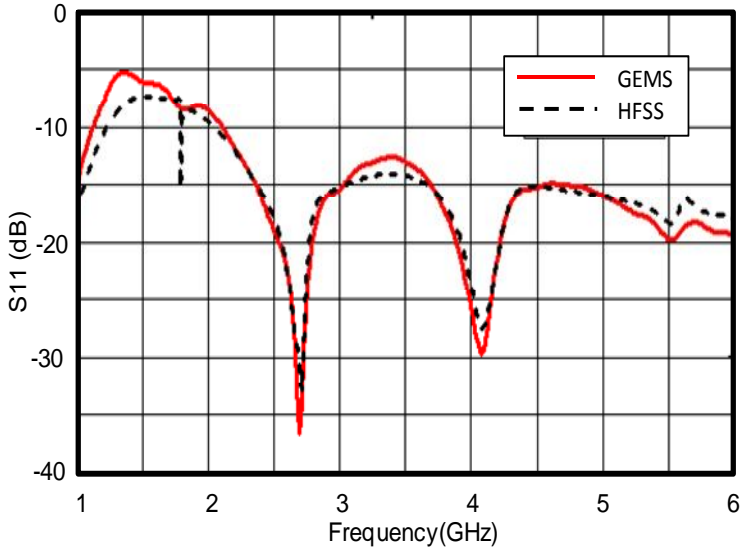


Figure 5.32 S_{21} and S_{12} comparison of the MIMO antenna between HFSS and GEMS.

Summary: The antenna geometry is not large in terms of wavelength; however, it is a little complex due to the fine structure inside the antenna. The software performance strongly depends on the user's experience, and the readers should simulate it using the different software packages and draw their own conclusions.

5.7 ELECTROMAGNETIC BANDGAP STRUCTURE

In this section, we use the periodic boundary condition to simulate an electromagnetic bandgap (EBG) structure [8], as shown in Figure 5.33. The EBG structure is constructed by using the infinitely long dielectric cylinder (radius = 2 mm and the relative dielectric constant = 4.2). The EBG is periodic in the transverse directions and there are six layers in the propagation direction.

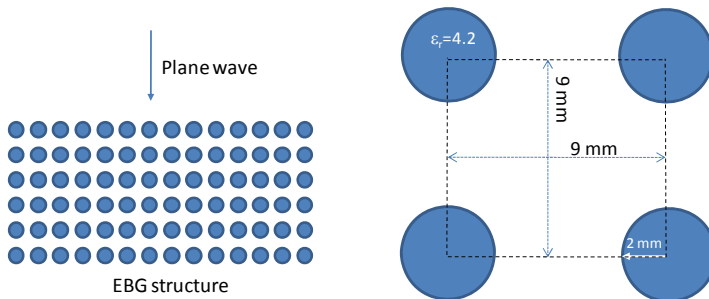


Figure 5.33 EBG structure illuminated by a plane wave and the element dimensions.

Using the periodic boundary condition, we only need to simulate the six dielectric cylinders (one periodic element) along the propagation direction, as shown in Figure 5.34.

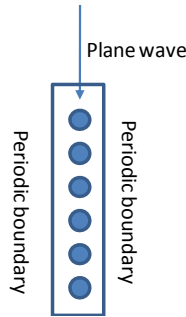


Figure 5.34 Simulation configuration of the EBG structure.

The domain is truncated in the cylinder axis direction by using the periodic boundary condition (PBC). However, the PML should be used to truncate the domain in the propagation direction. Suppose the periodic boundary is used to truncate the domain in the transverse directions; the distance between the periodic boundary to the dielectric cylinder should be equal to one-half distance between the two adjacent cylinders. The periodic boundary should touch the cylinder because the cylinder is infinitely long along the cylinder axial direction. In the propagation direction, there should be 10 cells of white space to leave sufficient room to place the plane wave surface.


Due to the periodic structure of EBG in the transverse directions, the plane wave box will be degenerated to be one surface, which is located at the incident side of EBG structure. In the GEMS simulations, the plane wave surface is always parallel to the X-Y plane regardless of the normal or oblique incidence.

We just need to simulate one element of the EBG structure, and hence, the domain is relatively small. To be safe, we can use the uniform and fine mesh to reach good simulation results. In this example, the frequency band of interest is up to 25 GHz (the wavelength in dielectric at 25 GHz is 6 mm), and we can use the cell size equal to 0.25 mm (about 24 cells per wavelength).

The outputs in the EBG structure simulation are the time-domain reflected and transmitted fields in GEMS. The time-domain reflected field of the higher-order Floquet modes includes the real and imaginary parts; however, the transmitted field of the main mode only has the real part, and the reflected and transmitted fields of main mode have a zero imaginary part.

The simulation information is summarized in Table 5.12. We tried to use the periodic boundary in the CST simulation; however, we cannot find any result in the result tree after the simulation was completed. We use the “Unit Cell” and frequency-domain solver to simulate the EBG structure.

Table 5.12 Simulation Information for the EBG Structure


	HFSS	CST (frequency-domain solver)	GEMS	FEKO
Cell size	N/A	0–30 GHz: steps per wavelength=4, min. number of steps=10	$\Delta x=0.25$, $\Delta y=0.25$, $\Delta z=0.25$, ratio=1	N/A
Number of unknowns	N/A	13,453 tetrahedrons	59,904 cells ($36 \times 8 \times 208$)	N/A
Boundary condition	N/A	N/A	PBC and open boundary	N/A
Simulation time	N/A	10 min., 4 sec (mesh time: 19 sec., solving: 585 sec)	4 min., 38 sec. (mesh generating: 6 sec., preprocessing: 1 sec., solving: 4 min., 31 sec.)	N/A
Simulation memory (MB)	N/A	Solver start: 12.2 Eq. system setup: 74.2 Eq. system solve: 152.9 Mesh refinement: 64.1 Solver run total: 152.9	20 (GEMS solver)	N/A
Environment memory (MB)	N/A	141 (modeler: 77.0, CST design: 64.0)	25.6 (controller: 11.2, solver: 14.4)	N/A
Total memory usage	N/A	>293.9 MB	45.6 MB	N/A
Time steps	N/A	N/A	13,321	N/A
Convergence criterion	N/A	S-parameter error threshold value: 0.01	–30 dB	N/A

In HFSS, we tried to use the different options for the periodic structure simulation, however there is no option to calculate the reflection and transmission coefficients for the periodic structure (to the best of our knowledge).

We tried to use the FEKO to simulate the periodic structure; however, the manual told us that the periodic boundary in FEKO is only good for perfect conductor (to the best of our knowledge).

For the same problem and mesh distribution, we use the GEMS workstation to simulate it and the simulation is summarized in Table 5.13.

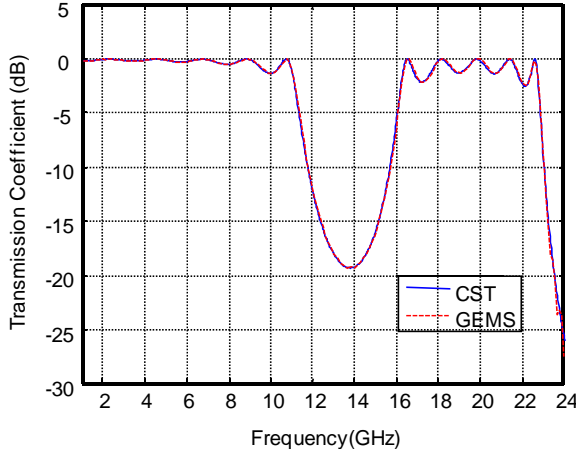
Table 5.13 GEMS Performance for EBG Structure on the Workstation

	Configuration	Operating System	Problem Size	Simulation Time
Workstation	Intel Q6600 quad core 2.4 GHz	Linux	59,904 cells (36×8×208)	51 sec.

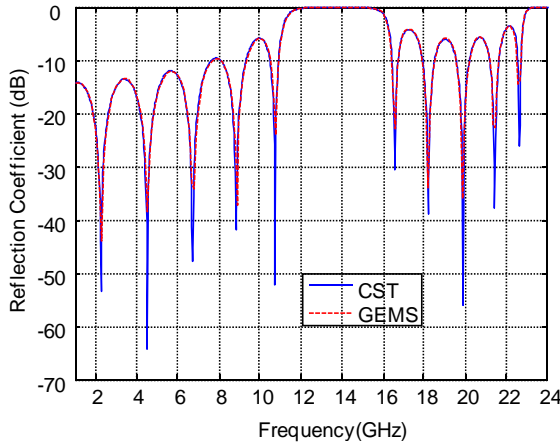
It is a rather straightforward job for us to use the PBC boundary to simulate a periodic structure using GEMS. Its procedure is exactly same as that used in the regular problem. The basic steps in GEMS are described as follows:

1. Specify a new unit and then create a new project.
2. Draw one cylinder and assign the material to it.
3. Use the hardclone or softclone option to generate a cylinder array.
4. Select the no DC differential Gaussian pulse and input the highest frequency in the excitation pulse setting window.
5. Specify the element size in the “Computational Domain” window, and select the PBC boundary in the x - and y -directions. There should be no white space between the PBC boundary and simulation area. Select the PML boundary in the z -direction, and have 10 cells of white space in the z -direction.
6. Design the mesh and the cell size selected to be 0.75 mm.
7. Validate the project and then simulate it on the laptop by clicking on the “Simulate” button.
8. Click on the “Start” button in the GEMS solver window and click on the “DataProcessing” button after the simulation is completed, or we can open a project result file by clicking the “*.GPV” file in the project folder.
9. Open the “Output” folder in the result tree in the GEMS display window. Go to the PBC folder and click on the either “AMP” or “AMP(dB)” in the “FreqDomain” folder to display the reflection or transmission coefficient.
10. In addition to the frequency results, GEMS also generates the time-domain data so we can derive further parameters when it is necessary. For example, we can view and export the time-domain data by clicking the option in the “TimeDomain” folder. We can calculate the reflection and transmission coefficients in any frequency band, which is different from the prespecified one.

The simulation results obtained from GEMS and CST are plotted in Figure 5.35. Although CST uses the unit cell boundary (frequency-domain solver) and GEMS (time-domain solver) uses the periodic boundary condition, the results from CST and GEMS matches very well. If we put the measurement data [21] in Figure 5.35, we will observe that it is in an excellent agreement with the simulation result.



(a) Transmission coefficient of the EBG structure



(b) Reflection coefficient of the EBG structure

Figure 5.35 (a) Reflection and (b) transmission coefficients of the EBG structure using CST and GEMS.

Summary: We only need to simulate one element of the periodic structure taking benefit from the periodic structure. The performance strongly depends on the user's experience, and the readers should simulate it using the different software packages and draw their own conclusions.

5.8 STANDARD SAR TEST

In this part, we use GEMS and CST to simulate the test case used in IEEE Standard 1528 [9]. The phantom configuration is shown in Figure 5.36.

The reference SAR value in the IEEE Standard is also generated by using the FDTD method. We tried to use HFSS and FEKO for the same problem, both of which showed the memory shortage error message. In CST, the SAR calculation is carried out in the data postprocessing. The simulation time in the postprocessing should be taken into account.

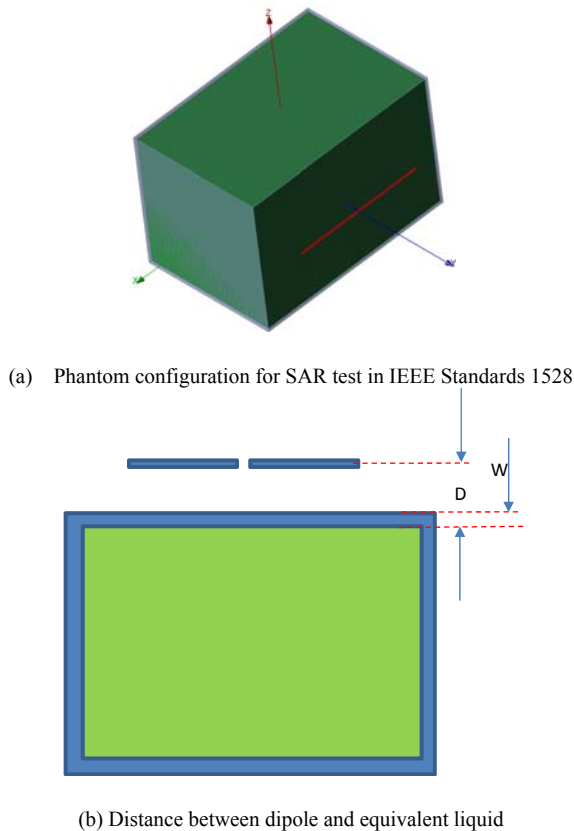


Figure 5.36 (a) Phantom configuration, and (b) distance between the dipole and equivalent liquid box.

In IEEE Standard 1528, the plastic box is filled with the tissue-equivalent liquid, and the dipole is used an excitation outside the plastic shell. The plastic shell has the relative permittivity = 3.7 and the loss tangent = 0.05 for all the test frequencies from 300 to 3,000 MHz. The parameters of tissue-equivalent liquid are listed in Table 5.14.

Table 5.14 Parameters of the Tissue-Equivalent Liquid

Frequency	Relative Permittivity	Electrical (S/m)	Conductivity	Dimensions of Equivalent Liquid
300 MHz	45.3	0.87		1,000×800×170
900 MHz	41.5	0.97		360×300×150
1,800 MHz	40.0	1.40		220×160×150
3,000 MHz	38.5	2.40		220×160×150

The dipole parameters at the different frequencies are listed in Table 5.15. The IEEE Standard 1528 requires that the diameter of dipole be less than 6 mm for frequency 300 MHz and 3.6 mm for frequencies from 900 to 3,000 MHz.

Table 5.15 Dipole Parameters

Frequency	Length of Dipole (mm)	Diameter of Dipole (mm)
300 MHz	396.00	6.35
900 MHz	149.00	3.60
1,800 MHz	72.00	3.60
3,000 MHz	41.50	3.60


The thickness of plastic shell and the distance variation of the distance between the tissue-equivalent liquid and dipole are given in Table 5.16 for the different frequencies.

Table 5.16 Plastic Shell and Distance Between Liquid and Center of Dipole

Frequency	Thickness W (mm)	Distance D (mm)
300 MHz	6.3 mm	15.00
900 MHz	2.0 mm	15.00
1,800 MHz	2.0 mm	10.00
3,000 MHz	2.0 mm	10.00


For the same problem and mesh distribution, we use a workstation to simulate it and the simulation is summarized in Table 5.17.

Table 5.17 GEMS Performance for 300-MHz SAR on the Workstation

	Configuration	Operating System	Problem Size	Simulation Time
Workstation	Intel Q6600 quad core 2.4 GHz	Linux	3,950,496 cells (232×198×86)	10 min., 35 sec.


For frequency 300 MHz, we use the dimensions and materials in Figure 5.36 and Tables 5.14 to 5.16. The simulation information is summarized in Table 5.18.

Table 5.18 Simulation Information for 300 MHz

	CST	GEMS
Cell size	0–800 MHz: $\lambda/15$, mesh line ratio limit: 10	$\Delta x=2$, $\Delta y=1.8$, $\Delta z=1.8$, ratio: 1.06
Number of unknowns	3,764,838 cells (281×231×58)	3,950,496 cells (232×198×86)
Boundary condition	Open (add space)	Open boundary (add space)
Simulation time	33 min., 55 sec. (simulation time: 1,263 sec., 1-g SAR: 282 sec., 10-g SAR: 487 sec., local SAR: 3 sec.)	32 min., 18 sec.
Simulation memory	Matrix calc.: 722.5 MB, Solver: 1,048.4 MB	182 MB (GEMS solver)
Environment Memory	156 MB (modeler: 82.4 MB, CST designer: 74.3 MB)	25.3 MB (controller: 11.1 MB, designer: 14.2 MB)
Total memory usage	>1,104 MB	207 MB
Time steps	1,650	4,221
Convergence criterion	−30 dB	−30 dB


For 900 MHz, we use the dimensions and materials shown in Figure 5.36 and Tables 5.14 to 5.16. The simulation information is summarized in Table 5.19.

Table 5.19 Simulation Information for 900 MHz

	CST	GEMS
Cell size	0–1.5 GHz: $\lambda/15$, Mesh line ratio limit: 10	$\Delta x=1$, $\Delta y=1$, $\Delta z=1$, Ratio: 1.05
Number of unknowns	2,544,696 cells (187×162×84)	2,626,680 cells (210×118×106)
Boundary condition	Open (add space)	Open boundary (add space)
Simulation time	21 min., 41 sec. (simulation time: 704 sec., 1-g SAR: 178 sec., 10-g SAR: 516 sec., local SAR: 3 sec.)	13 min., 46 sec.
Simulation memory	Matrix calc.: 568.8 MB, solver: 668.5 MB	172 MB (GEMS_solver)
Environment memory	146 MB (modeler: 71.4 MB, CST Designer: 74.6 MB)	25.3 MB (controller: 11.1 MB, GEMS designer: 14.2 MB)
Total memory usage	>814 MB	197 MB
Time steps	1,320	4,787
Convergence criterion	–30 dB	–30 dB


For this problem and mesh distribution, we use a workstation to simulate it and the simulation is summarized in Table 5.20.

Table 5.20 GEMS Performance for 900 MHz on the Workstation

	Configuration	Operating System	Problem Size	Simulation Time
Workstation	Intel Q6600 quad core 2.4 GHz	Linux	2,626,680 cells (210×118×106)	4 min., 28 sec.


For frequency 1,800 MHz, we use the dimensions and materials shown in Figure 5.36 and Tables 5.14 to 5.16. The simulation information is summarized in Table 5.21.

Table 5.21 Simulation Information for 1,800 MHz

	CST	GEMS
Cell size	0–2.5 GHz: $\lambda/15$, mesh line ratio limit: 10	$\Delta x=1$, $\Delta y=1$, $\Delta z=1$, x -, z -ratio: 1.02, y -ratio: 1.01
Number of unknowns	3,494,920 cells (188×143×130)	3,581,500 cells (190×145×130)
Boundary condition	Open (add space)	Open boundary (add space)
Simulation time	63 min., 1 sec. (simulation time: 860 sec., 1-g SAR: 598 sec., 10-g SAR: 2,320 sec., local SAR: 3 sec.)	13 min., 43 sec.
Simulation memory	Matrix calc.: 683.8 MB, Solver: 855.5 MB	169 MB (GEMS_solver)
Environment memory	145 MB (modeler: 71.3, CST designer: 74.6)	28.6 MB (controller: 13.6, GEMS designer: 15.0)
Total memory usage	>1,000 MB	197 MB
Time steps	1,100	1,889
Convergence criterion	–30 dB	–30 dB


For 3,000 MHz, we use the dimensions and materials shown in Figure 5.36 and Tables 5.14 to 5.16. The simulation information is summarized in Table 5.22.

Table 5.22 Simulation Information for 3,000 MHz

	CST	GEMS
		
Cell size	0–4 GHz: $\lambda/10$, mesh line ratio limit: 10	$\Delta x=1$, $\Delta y=1$, $\Delta z=1$, x-direction ratio: 1.02, y- and z-direction ratio: 1.01
Number of unknowns	3,931,216 cells (194×149×136)	3,852,940 cells (182×145×146)
Boundary condition	Open (add space)	Open boundary (add space)
Simulation time	75 min., 21 sec. (simulation time: 858 sec., 1-g SAR: 607 sec., 10-g SAR: 30,53 sec., local SAR: 3 sec.)	9 min., 40 sec.
Simulation memory	Matrix calc.: 763.6 MB, Solver: 937.4 MB	176 MB (GEMS_solver)
Environment memory	135 MB (modeler: 71.5 MB, CST designer: 64.3 MB)	25.4 MB (controller: 11.0 MB, GEMS designer: 14.4 MB)
Total memory usage	>1,072 MB	201 MB
Time steps	665	1,259
Convergence criterion	–30 dB	–30 dB

For the same problem and mesh distribution, we use the GEMS workstation to simulate it and the simulation is summarized in Table 5.23.

Table 5.23 GEMS Performance for 3 GHz on the Workstation

	Configuration	Operating System	Problem Size	Simulation Time
Workstation	Intel Q6600 quad core 2.4GHz	Linux	3,852,940 cells (182×145×146)	2 min., 49 sec.

The SAR values simulated by using CST and GEMS are summarized in Table 5.24. Though there are more frequencies given in the IEEE Standard 1528, we only show the SAR values at the four representative frequencies.

Table 5.24 Simulation Result Summary

Frequency	SAR Type	IEEE Standard 1528 (Unit: W/kg)	CST (Unit: W/kg)		GEMS (Unit: W/kg)	
			SAR	Error	SAR	Error
300 MHz	1-g	3.0	2.75	0.25	2.92	0.08
	10-g	2.0	1.62	0.38	1.94	0.06
	Local	4.4	3.77	0.77	4.10	0.31
900 MHz	1-g	10.8	10.65	0.15	10.65	0.15
	10-g	6.9	6.58	0.32	6.81	0.09
	Local	16.4	15.36	1.04	15.86	0.74
1,800 MHz	1-g	38.1	36.19	1.81	37.56	0.60
	10-g	19.8	20.43	0.63	19.65	0.15
	Local	69.5	68.46	1.05	66.75	2.85
3,000 MHz	1-g	63.8	63.38	0.42	62.52	1.30
	10-g	25.7	24.67	1.03	25.38	0.43
	Local	140.2	146.99	6.79	138.36	1.64
Average error			1.22		0.70	

We plot the SAR difference between CST and GEMS for 1-g SAR, 10-g SAR, and local peak SAR in Figures 5.37 to 5.39. It is evident from Figures 5.37 to 5.39 that GEMS is much more accurate than CST. The x- and y-axes in Figures 5.37 to 5.39 stand for the frequencies and the relative SAR error to the reference given in the IEEE Standard 1528, respectively. The relative error in Figures 5.37 to 5.39 is calculated using the formula:

$$SAR_Relative_Error = \frac{SAR_{Simulation} - SAR_{Reference}}{SAR_{Reference}}$$

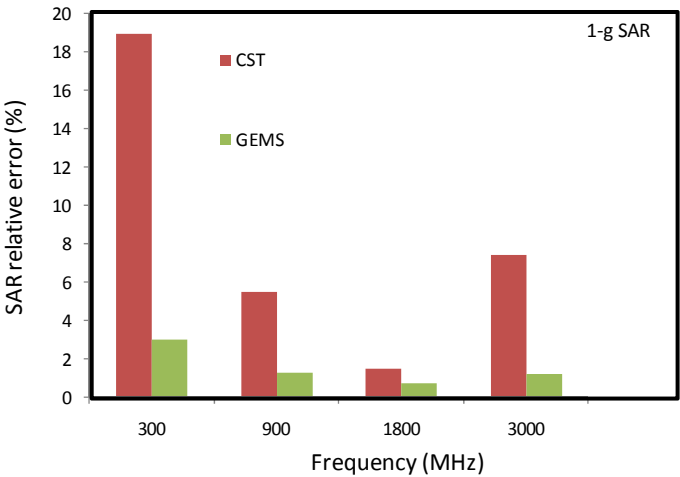


Figure 5.37 Relative 1-g SAR error of GEMS and CST compared to the reference in IEEE Standard 1528.

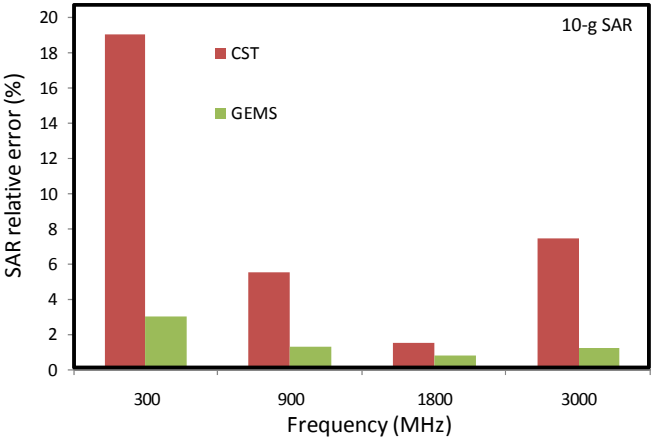


Figure 5.38 Relative 10-g SAR error of GEMS and CST compared to the reference in IEEE Standard 1528.

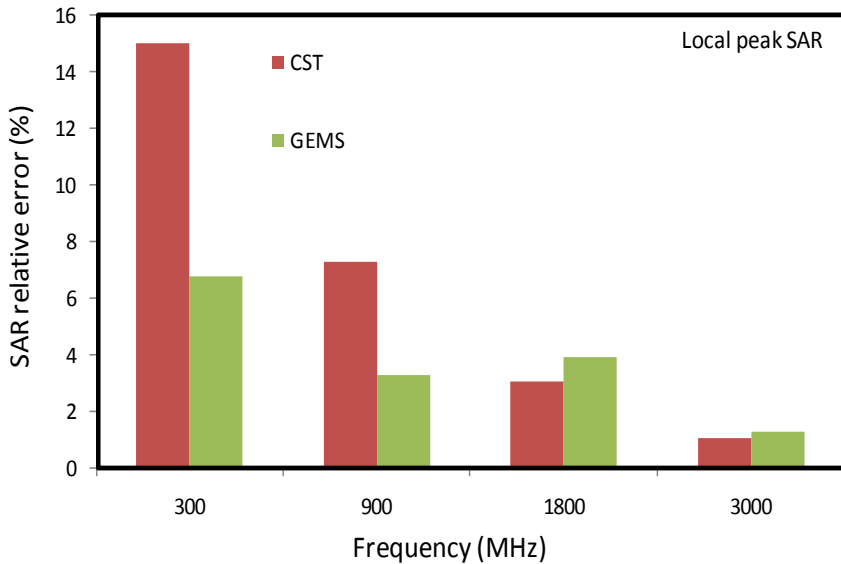


Figure 5.39 Relative peak SAR error of GEMS and CST compared to the reference in IEEE Standard 1528.

For the SAR calculation, GEMS can output the 1-g SAR, 10-g SAR, local peak SAR, average SAR in the specified region, and 3-D SAR distribution. For example, the 3-D SAR distribution at 900 MHz is shown in Figure 5.40. The 3-D SAR distribution allows us to view the SAR distribution inside an object, as shown in Figure 5.41.

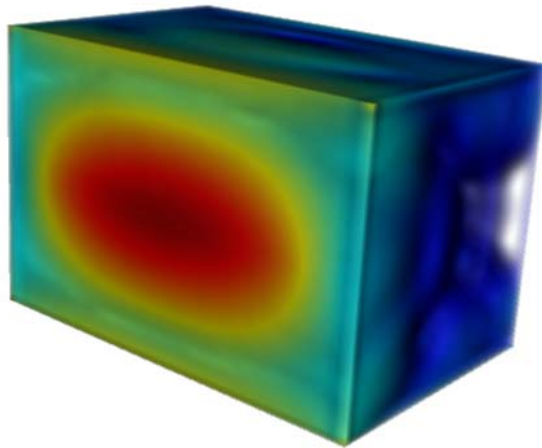


Figure 5.40 Normalized 3-D SAR distribution inside the Phantom structure at 900 MHz.

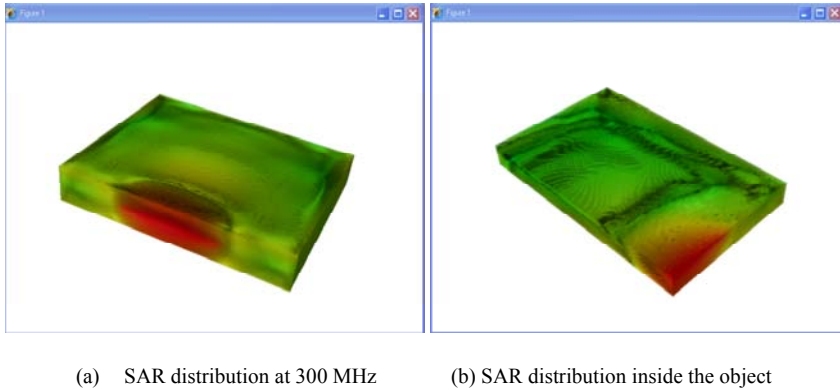


Figure 5.41 Normalized 3-D SAR distribution (a) whole structure and (b) partial structure at 300 MHz obtained by using the GEMS software package.

Summary: The software performance strongly depends on the user's experience, and readers should simulate it using the different software packages and draw their own conclusions.

5.9 WAVEGUIDE FILTER

In this section, we introduce the simulation method for the high Q systems using the time-domain solver. One waveguide filter, as shown in Figure 5.42, whose dimensions were taken from the literature [10], includes five open cavities and is a very good passband filter in the frequency range from 11.8 to 12.3 GHz. The original paper shows five poles; however, following the dimensions given in the paper, we can only see four poles.

It is a challenging problem for the time-domain solver since it may require running the time-domain solver very long to get the convergence solution. There exist two techniques that can be used to efficiently achieve the convergence solution. In the first technique, a window function such as the Hamming window function or Kaiser function is used to truncate the time-domain signatures. We can get very good results for the S-parameters and impedance. The main problem caused by the window functions is that the power conservation will not be satisfied in the passband, for example, the summation of $|S_{11}|^2$ and $|S_{21}|^2$ for a two-port filter is smaller than 1 in the passband. In the second technique, we only need to run the simulation in a short time period to get the major information of the resonant structure, and then the extrapolation technique is applied to get the complete system time-domain response. However, this approach strongly depends on the system property and the user's experience. GEMS provides both methods, but their performance may vary from case to case.

In this part, we introduce the detailed simulation procedure for the waveguide filter, and present the simulation results from HFSS and CST as the reference. As the statements in the previous examples shown, the results from other software

packages were carried by the independent third party. The results may vary from user to user and from version to version.

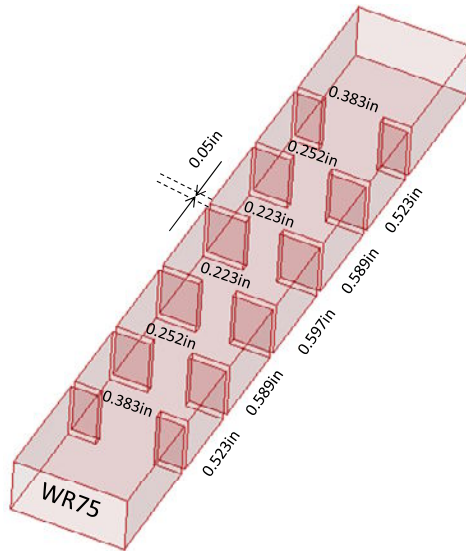
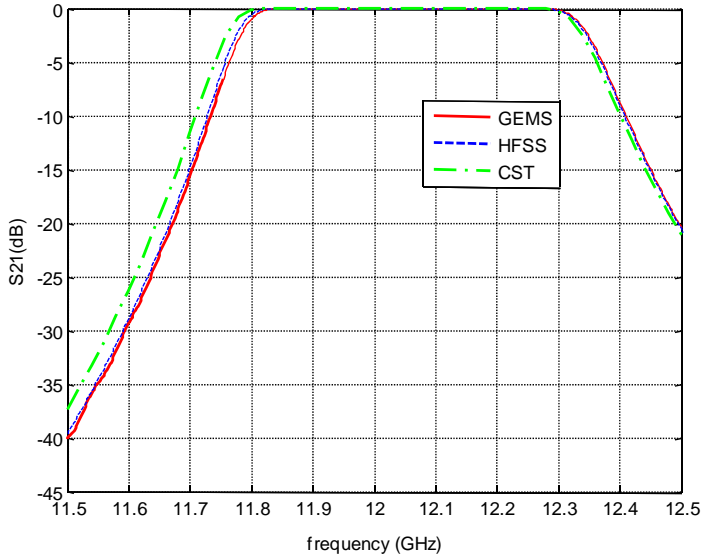


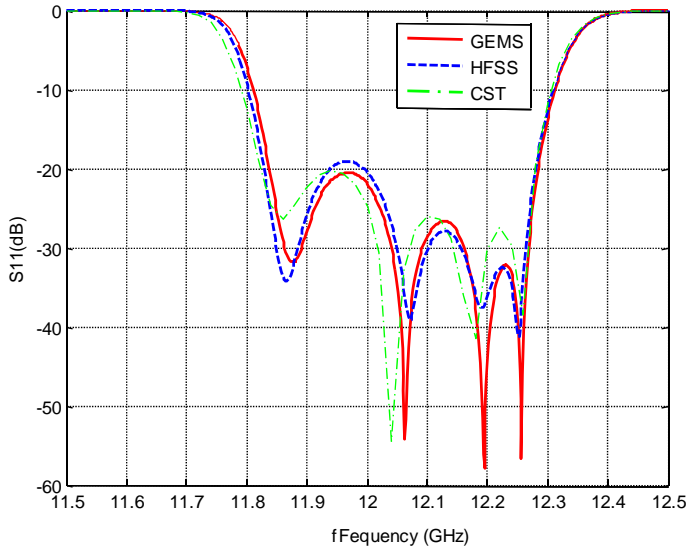
Figure 5.42 Configuration and dimensions of the waveguide filter and the dimensions of each cavity.

The design of the waveguide filter is in the SAT format rather than in a solid structure. To generate a hollow waveguide filter model, we can select one surface of the solid model and copy and paste the selected surfaces. Since the two ends of filter are opened, we deselect the two end surfaces to get a group of surfaces. We then delete the original model, and unite the surfaces as a single object and then assign it to be a perfect electric conductor (PEC). We can use GEMS to simulate it directly since GEMS allows us to simulate the infinitely thin shell. Use the PML boundary to truncate the two opened ports and use the PEC boundary condition to truncate the four sides of waveguide. We do not need any white space between the filter to the domain boundary since the boundary should touch the filter in the six directions.

We use TE_{10} mode to excite one port of filter and measure the transmission on TE_{10} mode at the output port. The cutoff frequency of WR75 (width=19.5 mm and height=9.525 mm) extracted in GEMS is 7.868 GHz. The domain is discretized into $57 \times 26 \times 288$ nonuniform cells with the minimum cell size 0.3 mm and ratio 1.03. The convergence criterion is chosen to be -50 dB. The S-parameters are plotted in Figure 5.43 with HFSS and CST results together for the sake of comparison.



(a) S_{11} of the waveguide filter from the different packages



(b) S_{21} of the waveguide filter from the different packages

Figure 5.43 S-parameters of the waveguide filter using the different software packages (a) S_{11} and (b) S_{21} .

The power conservation is plotted in Figure 5.44. We can observe from Figure 5.44 that both CST and HFSS show better power conservation than GEMS. That is a greater number of cells to describe the mode pattern accurately. In addition, we plot the field variation along the central line of the filter at frequency 12 GHz. The field distributions at the different frequencies are shown in Figure 5.45. The field distribution inside the filter at the different frequencies is shown in Figure 5.46.

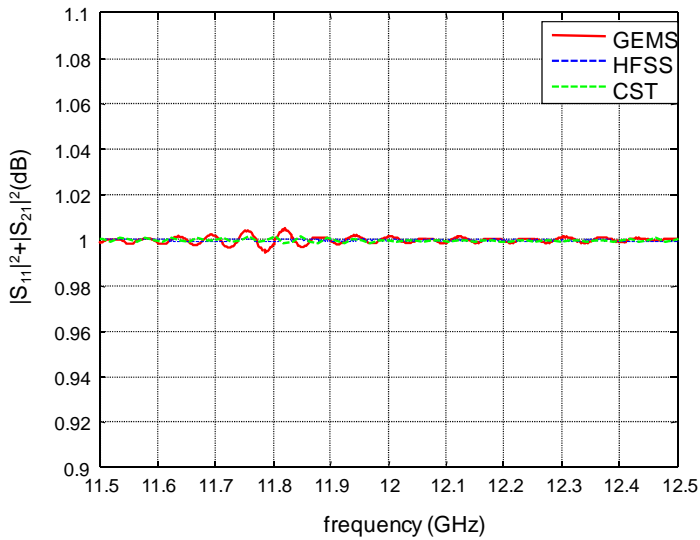


Figure 5.44 Power conservation versus frequencies.

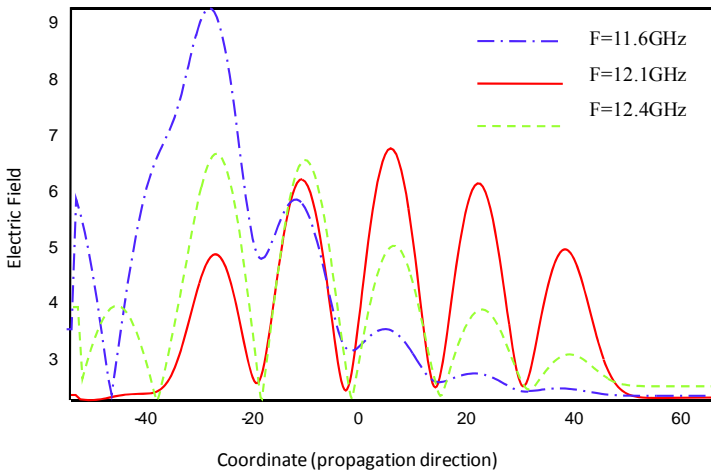


Figure 5.45 Field E_y variation along the filter with frequencies.

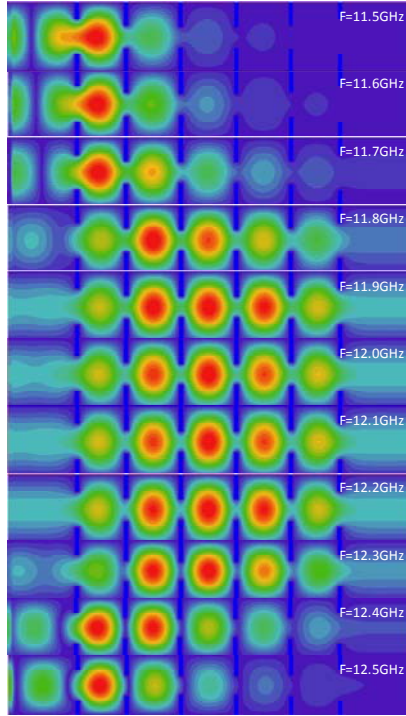


Figure 5.46 Normalized field E_y distribution inside the waveguide filter at $y=0.9525$ mm.

In GEMS, we use three ways to terminate the simulation: (1) check the time-domain signature, (2) check the S-parameters in frequency domain, and (3) check the power level (advanced option). For example, if you use -50 dB as a convergence criterion and the number of time steps for one domain period along the longest direction is N , we will pick the maximum absolute value in the last N steps and compare to the global peak value, the code will stop if the ratio is less than -50 dB. We also compare the frequency-domain result at each frequency and the simulation will stop when the maximum difference is less than -50 dB. The power conservation checking option is designed for the lossless system, in which we use the formula $1 - |S_{11}|^2 - |S_{21}|^2$ to check the power convergence for a two-port network system. Without a window function, we directly run it until reaching the convergence criterion -50 dB. GEMS achieved the convergence solution at about 35,000 time steps.

Unlike the discussion in the previous examples, we will not show the detailed comparison here since the simulation time may be hugely different using a different mesh size for similar results. In CST, when using the original CAD file to generate a waveguide that has an infinitely thin waveguide wall, the CST results

have a significant shift from both GEMS and HFSS. When the waveguide is changed to be a finite thickness or the background is filled with PEC, the CST results are close to the HFSS and GEMS.

Once again, the examples presented in this chapter are used to help the readers get a basic idea of how to apply the different computational approaches to simulate some simple examples. The performance, ability, accuracy, and memory usage of each EM software package on different practical applications may vary significantly. For example, HFSS can generate the adaptive mesh without requiring any experience in these simple examples, and HFSS can generate reasonable results except the dielectric sphere and Vivaldi antenna. However, for most practical applications such as the cell phone antenna in Section 5.6, using the default mesh in HFSS, you can either simulate them limited to the available memory or take a very long time to get the results, or you are required to work on the model modifications for days or weeks before the simulation. Therefore, the best way to generate an adaptive mesh in GEMS is that based on the problem you are solving, you can directly provide the EM software the minimum cell size that is equal to the dimension of interest or half, and then adjust the mesh ratio according the available memory and expected simulation time.

Before closing this chapter, we do a standard test for the time-domain solvers, in which a hollow box is truncated by using the PEC boundary and discretized into the uniform mesh. Using a computer installed with 2-GB memory (Windows XP OS), GEMS can solve a domain size with 54 million unknowns ($300 \times 300 \times 600$), but CST on the same computer can only solve a domain size with 8 million unknowns ($200 \times 200 \times 200$). Now let us test the performance of GEMS and CST for a hollow domain with 1 million unknowns ($100 \times 100 \times 100$). The simulation time and memory requirement are summarized in Table 5.25.

Table 5.25 Simulation Summary

Option	CST	GEMS
Problem size (unknowns)	$100 \times 100 \times 100$	$100 \times 100 \times 100$
Number of time steps	1,000	1,000
Simulation time	112 sec.	46 sec.
Memory usage	301.5 MB	50.6 MB

References

- [1] HFSS, <http://www.ansoft.com>.
- [2] CST, <http://www.cst.com>.
- [3] FEKO, <http://www.feko.info>.
- [4] GEMS, <http://www.2comu.com>.
- [5] http://www.mwee.com/magazine/2000/cad_benchmark.html.
- [6] Q. Rao and D. Wang, "A Compact Dual-Port Diversity Antenna for Long-Term Evolution Handheld Devices," *IEEE Transactions on Vehicular Technology*, March 2010, pp. 1319–1329.
- [7] R. F. Harrington, *Time-Harmonic Electromagnetic Fields*, New York: Wiley-IEEE Press, 2001.
- [8] A. Roden and S. Gedney, "Time-Domain Analysis of Periodic Structures at Oblique Incidence: Orthogonal and Nonorthogonal FDTD Implementations," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 46, No. 4, April 1998, pp. 420–427.
- [9] "Recommended Practice for Determining the Peak Spatial-Average Specific Absorption Rate (SAR) in the Human Head from Wireless Communications Devices: Measurement Techniques," IEEE Standard 1528, 2003.
- [10] M. Yu, "Power-Handling Capability for RF Filters," *IEEE Microwave Magazine*, Vol. 8, No. 5, October 2007, pp. 89–97.

Chapter 6

Large Multiscale Problem Solving

In this chapter, we present several practical applications that are simulated by using the FDTD method on either a powerful workstation or a cluster. We use these examples to introduce the key simulation techniques that will significantly affect the simulation performance and the computer resource requirements [1–4].

6.1 RADIO FREQUENCY PROTECTION

A SAM head model [5] with a smart cell phone aside is shown in Figure 6.1. The human model includes a plastic shell (relative permittivity = 5 and the conductivity = 0.0125 S/m) and is filled with a liquid whose properties are similar to that of the human tissue. It has a relative permittivity of 39 and a conductivity of 1.89 S/m.

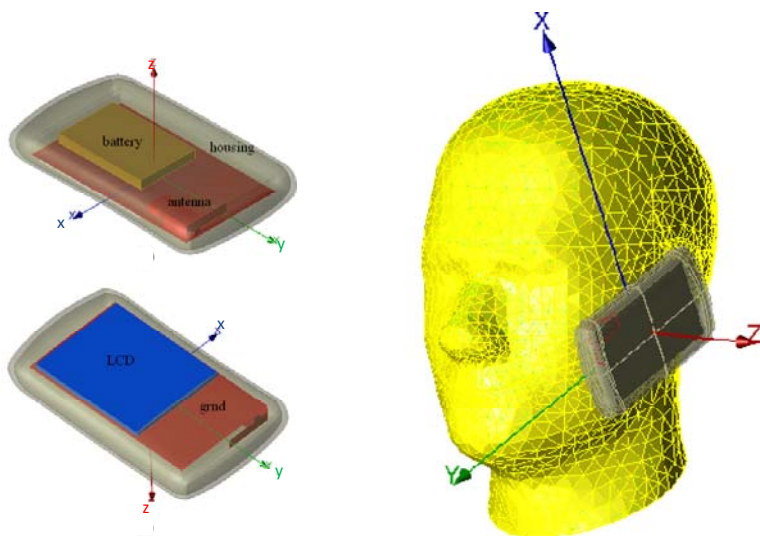
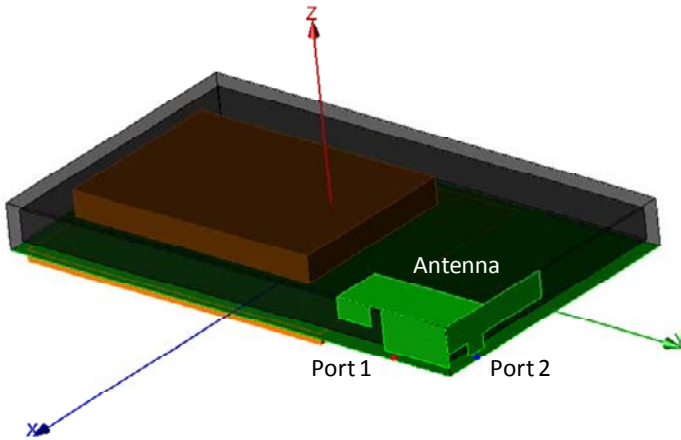
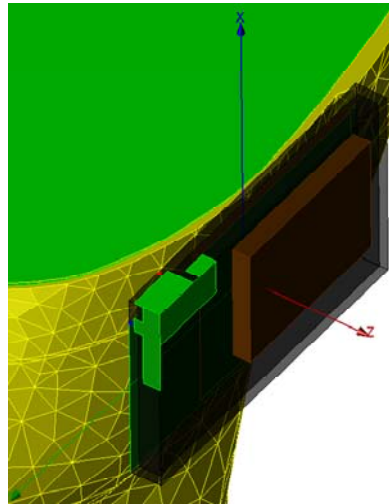


Figure 6.1 A practical cell phone with the SAM head model for the SAR and S-parameter calculation.

There are several important factors we need to consider when we design the simulation: (1) two fine structures, the antenna and shell of the human model, as shown in Figures 6.2 and 6.3, (2) antenna orientation, and (3) the distance between the cell phone and human model. The fine features in the cell phone will determine the minimum cell size, which will decide the simulation performance and accuracy. The antenna ports must be oriented along with the system axis to generate the correct simulation results. Even if we cannot do it in some special case, we have to add an extension part to meet the requirement. The distance between the cell phone and human model will demonstrate the correct SAR level.



(a) Antenna structure inside the cell phone



(b) Cross section of the human head model

Figure 6.2 (a) Cell phone configuration and (b) cross section of the SAM human head model.

Now we explain how to set the correct simulation options in the FDTD simulation. We first set the cell phone in the proper orientation so that the antenna ports are parallel to the system axis. If possible, the major antenna pieces are also set to be parallel to the system axis. Since there is no fine structure inside the human head model, the orientation of the head model is not so important. However, if we like to plot the 3-D far-field pattern referring to the human head model, we may need to rotate the 3-D far field according to the requirements. The shell of the human head model is in a complex shape, but the conformal technique can handle it regardless of the orientation. Therefore, we set the position and orientation of the cell phone in the desired way, and then change the position and orientation of the human head model accordingly.

In order to set the distance between the cell phone and the head model, we can define two points, one on the cell phone and the other on the human model, as shown in Figure 6.3. The distance between the cell phone and the human head model can be adjusted by controlling the distance of the two points.

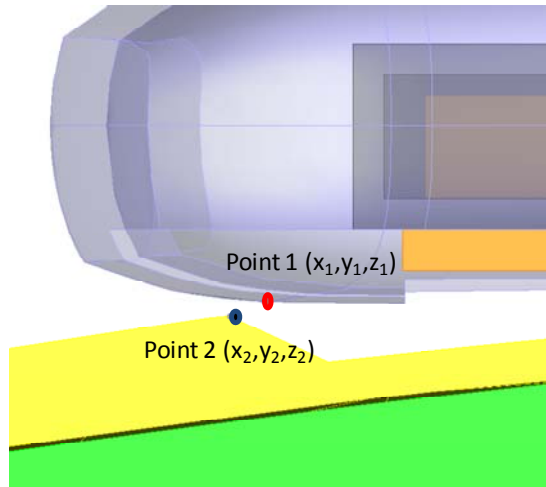


Figure 6.3 Reference points defined in the cell phone and the head model to set up the phone relative position.

This is an open space problem, and the absorbing boundary is applied to truncate the six walls, and there are six cells of white space between the objects and absorbing boundary. For this SAM head model, both the plastic shell and inner equivalent material are not dispersive, and hence the simulation performance is relatively good compared to the dispersive case.

We select the cell size based on the fine structure in the phone model. In this example, we need to use two to three cells inside the excitation gap to get good results. In some cases, there are many fine features around the feed port, and we may not capture the key points in the feed structure. To let the grids capture the key points in the feed structure, we can define some key points on the port

structure, and force the grids to pass these key points when we use the automesh generation procedure, as shown in Figure 6.4.

For the antenna problems, the antenna material can be copper (good conductor) or PEC that does not affect much on the return and far-field pattern. We smartly use the copper or PEC material only when we would like to reduce the simulation time or memory requirement. In general, the PEC is good for the most cases.

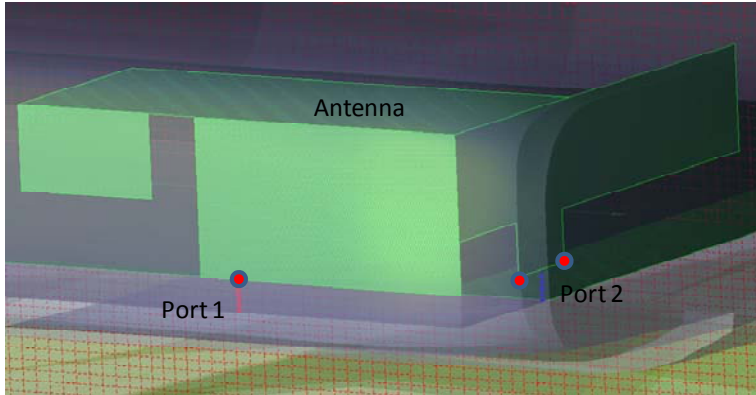
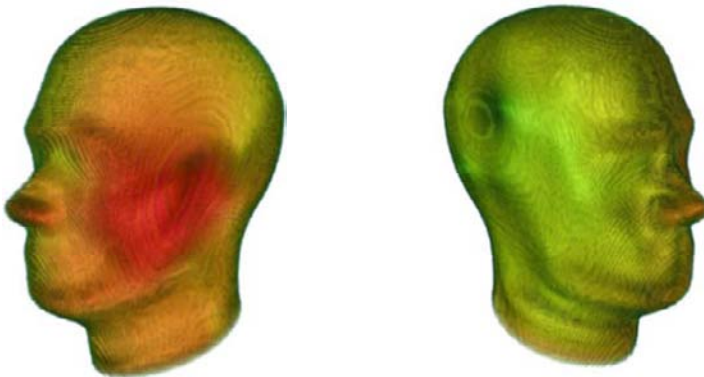


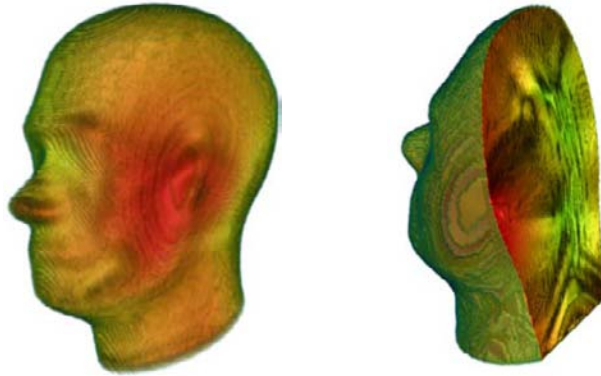
Figure 6.4 Antenna ports with the key points and the local mesh distribution.

We can use the pure Gaussian pulse or differential pulse as the excitation pulse for this problem. The 3-dB beamwidth frequency is selected to be 2.7 GHz, which is the highest frequency of interest. The 3-D SAR distributions at 2.4 GHz, 2.6 GHz, and 3.0 GHz inside the head model are shown in Figures 6.5 to 6.7, which enable us to check to see if the phone is compliant.



(a) SAR distribution on the phone side (b) SAR distribution in the shade region

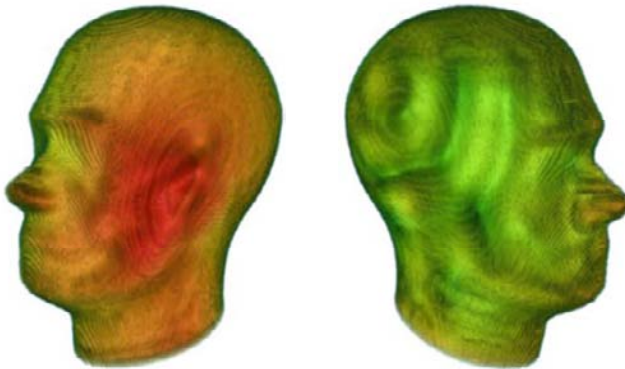
Figure 6.5 The 3-D SAR distribution (a) on the phone side and (b) in the shade region on the decibel scale inside the human head at 2.4 GHz.



(a) SAR distribution on the phone side

(b) SAR distribution inside the model

Figure 6.6 The 3-D SAR distribution (a) on the phone side and (b) inside the model on the decibel scale inside the human head at 2.8 GHz.



(a) SAR distribution on the phone side

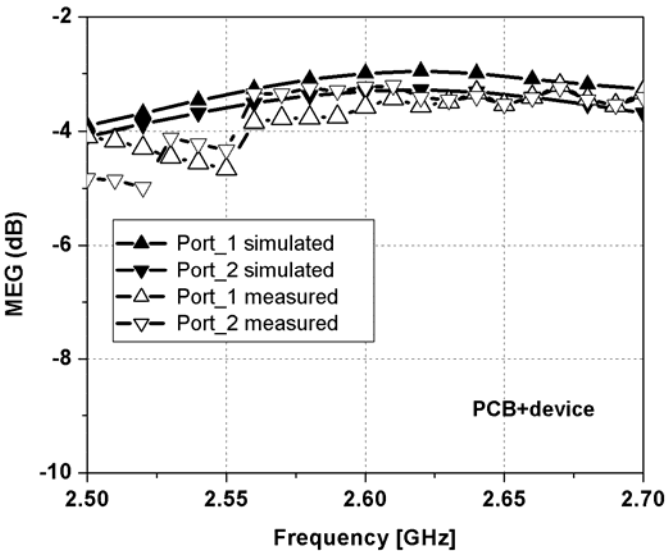
(b) SAR distribution in the shade region

Figure 6.7 The 3-D SAR distribution (a) on the phone side and (b) in the shade region on the decibel scale inside the human head at 3.0 GHz.

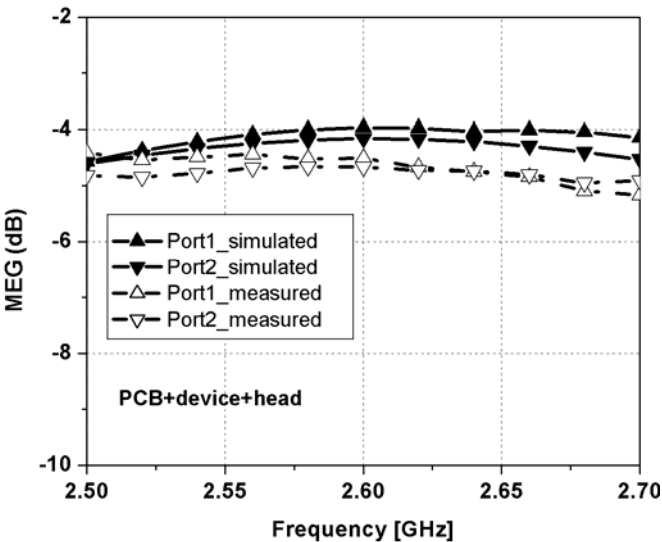
To calculate the SAR, we need to assign the material density to the structure; otherwise, the FDTD code cannot generate the SAR output directly. The SAR is 3-D data that is usually big and its output is time-consuming. To improve the simulation performance, we can select a small region where the SAR value is interesting. The FDTD code only calculates the SAR distribution, 1-g SAR, 10-g SAR, peak SAR, and average SAR in the specified region.

We compare the simulated results in Figure 6.8 for mean effective gain (MEG) with the measured data to show that the comparison between the two is good. The S-parameters of the cell phone structure are plotted in Figure 6.9,

together with the measurement data, and we observe good agreement between the two.



(a) MEG of cell phone without the SAM head model



(b) MEG of cell phone with the SAM head model

Figure 6.8 Mean effective gain (MEG) of cell phone (a) without the human head and (b) with the human head.

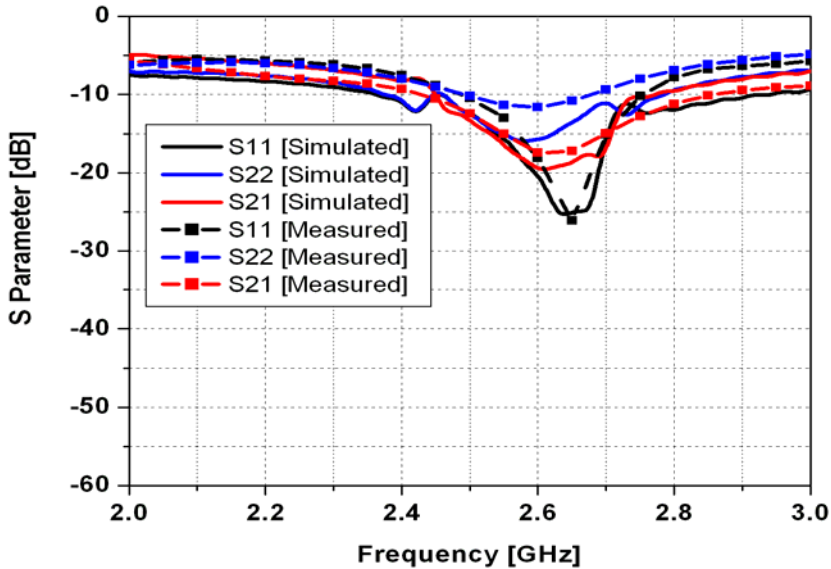


Figure 6.9 S-parameter of cell phone model using the FDTD method.

6.2 COMB-SHAPED ANTENNA ARRAY

The comb-shaped element of antenna array [7–12] is shown in Figure 6.10. This is an open space problem and the absorbing boundary is applied to truncate all the six walls without any white space between the antenna and absorbing boundary. The frequency band of interest is from 2 to 11 GHz. The differential Gaussian pulse with the 3-dB beamwidth of 11 GHz is used as the excitation pulse. The project model is shown in Figure 6.11.



Figure 6.10 Configuration of the comb-shaped antenna element used in the experiment.

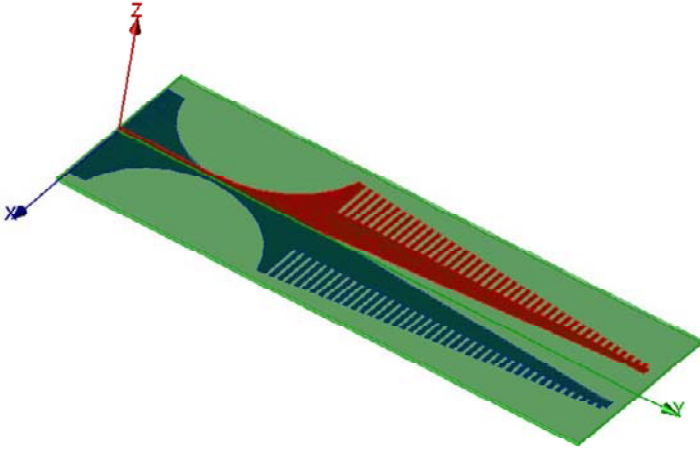


Figure 6.11 Element configuration of comb-shaped antenna array built for simulation.

We use three cells inside the gap between two arms, as shown in Figure 6.12. The return loss and peak gain of a pair of comb elements obtained from both the measurement and simulation are plotted in Figure 6.13. It is observed from Figure 6.13 that the deviation between the measurement and the simulation at the higher frequencies occurs because the measurement is not accurate at the higher frequencies. The good agreement between the simulation and measurement is evident from Figure 6.14. It is worthwhile to mention that the peak gain plotted in Figure 6.14 is in the E-plane and H-plane since the measurement was carried out in the two major cuts.

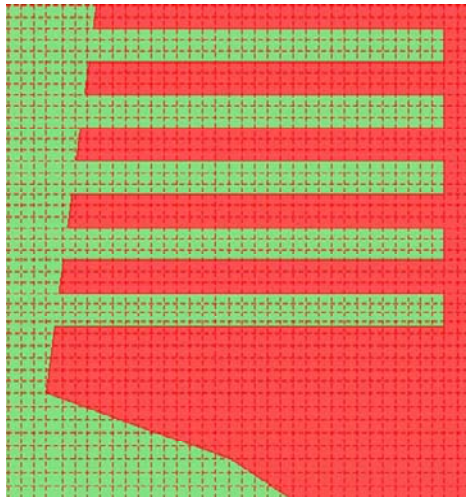


Figure 6.12 Localized mesh distribution for the description of the fine structure in the antenna.

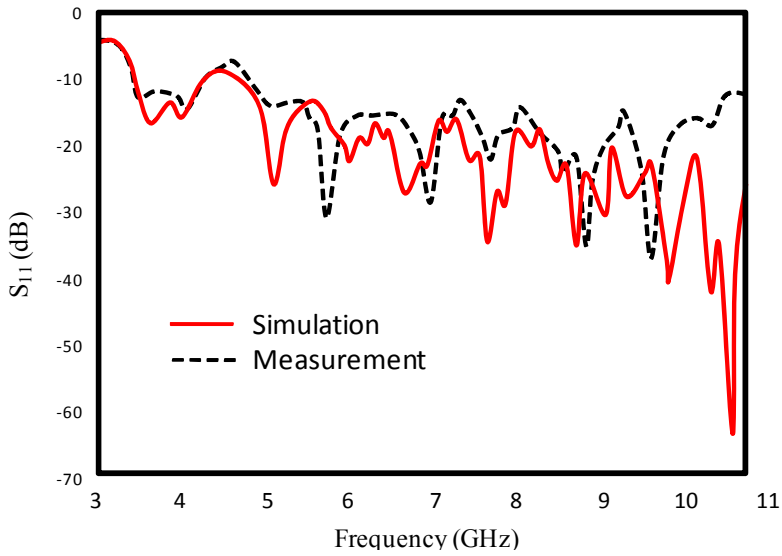


Figure 6.13 Comparison of the return loss of the comb-shaped antenna element between the simulation and measurement.

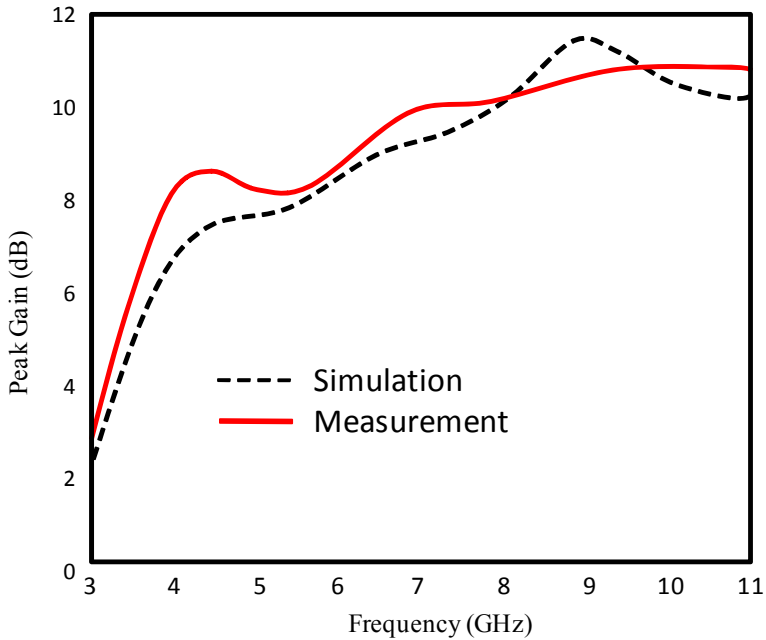


Figure 6.14 Comparison of the peak gain of the comb-shaped element between the simulation and measurement.

The far-field pattern of the comb-shaped element obtained from both the measurement and the FDTD simulation are plotted in Figure 6.15. Once again, the good agreement between the FDTD simulation and measurement is evident.

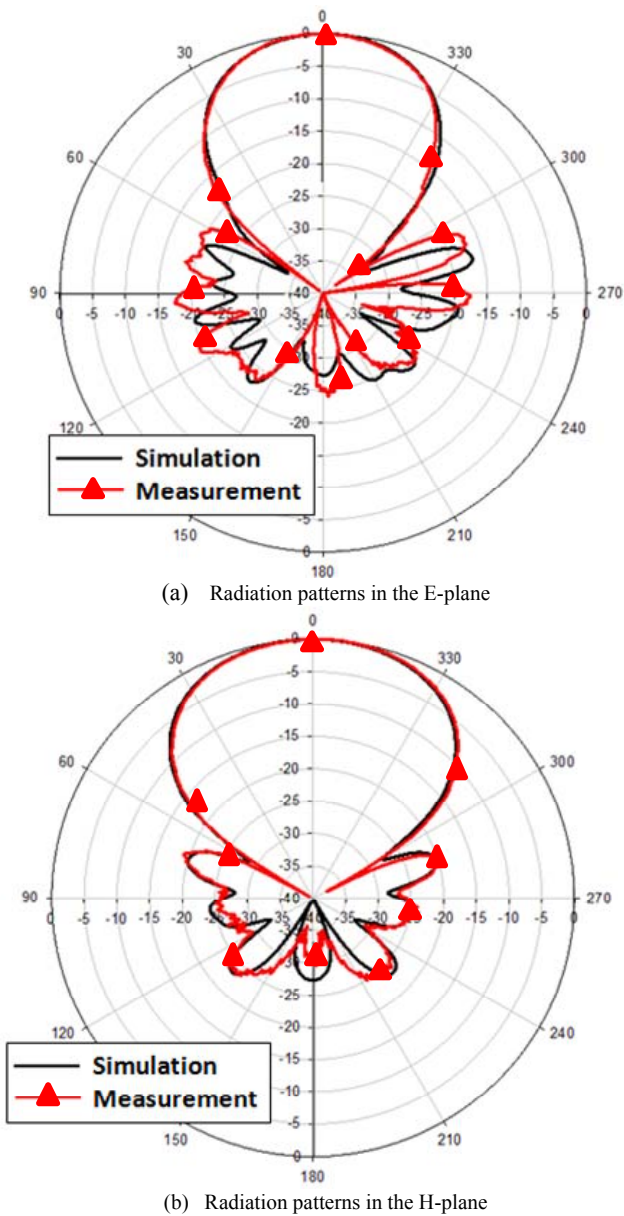


Figure 6.15 Radiation patterns (a) in the E-plane and (b) in the H-plane of the comb-shaped antenna element at 7 GHz.

The configuration and dimensions of a four-way UWB power divider are shown in Figure 6.16, which is used as the feed network of the antenna array. The permittivity and loss tangent of substrate (FR4) are 4.4 and 0.0254, respectively. The prototype of the antenna array is shown in Figure 6.17.

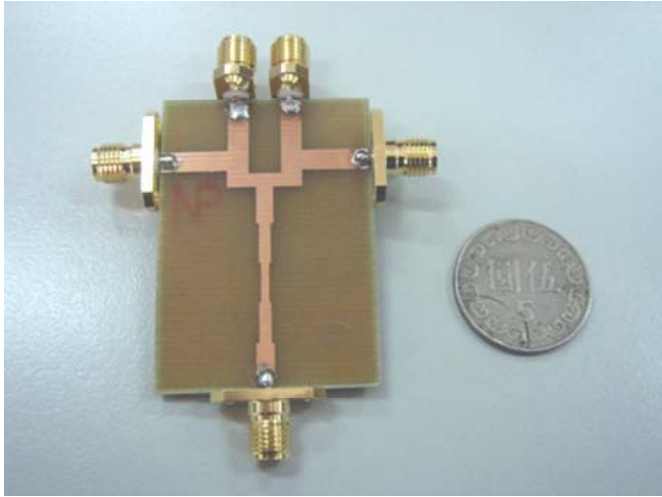
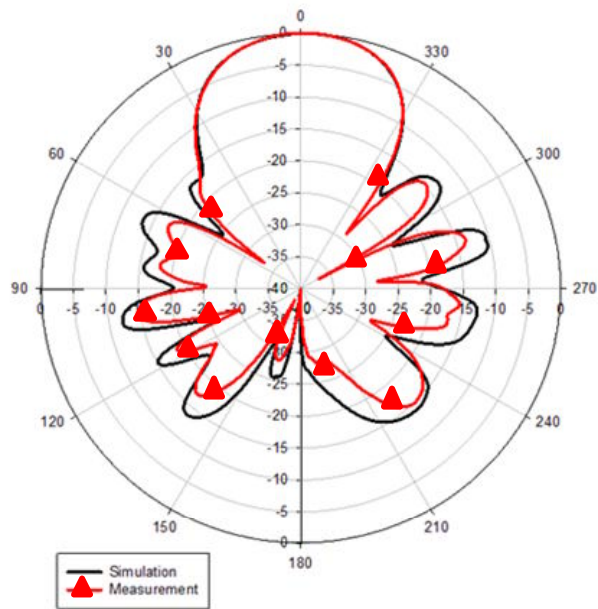


Figure 6.16 Configuration of the four-way UWB power divider.

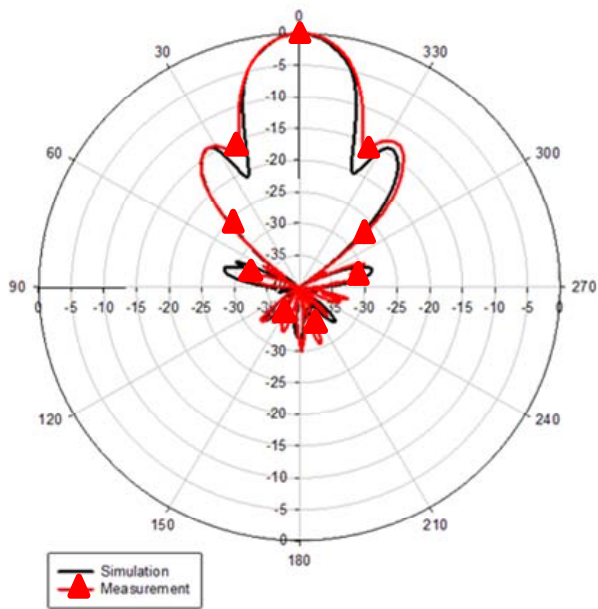


Figure 6.17 Prototype of the UWB antenna array that is used in the experiment.

The far-field patterns of the UWB antenna array in the E-plane and H-plane at 7 GHz are plotted in Figure 6.18, and the good agreement between the measurement and the FDTD simulation can be observed from Figure 6.18.



(a) Radiation pattern in the E-plane.



(b) Radiation pattern in the H-plane.

Figure 6.18 Radiation patterns of the UWB antenna array at 7 GHz.

6.3 TACKING REFLECTOR ANTENNA

A high gain antenna [13] includes a parabolic reflector antenna, as shown in Figure 6.19 and a feed system, as shown in Figure 6.20.

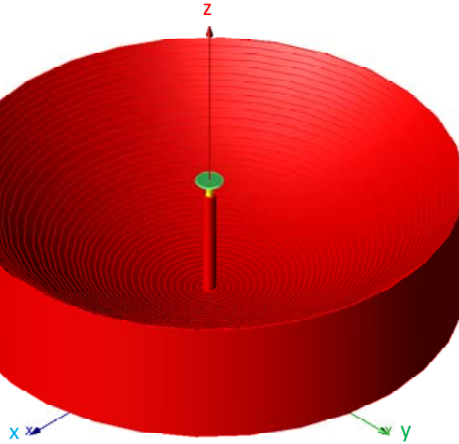


Figure 6.19 Configuration of the high gain reflector antenna.

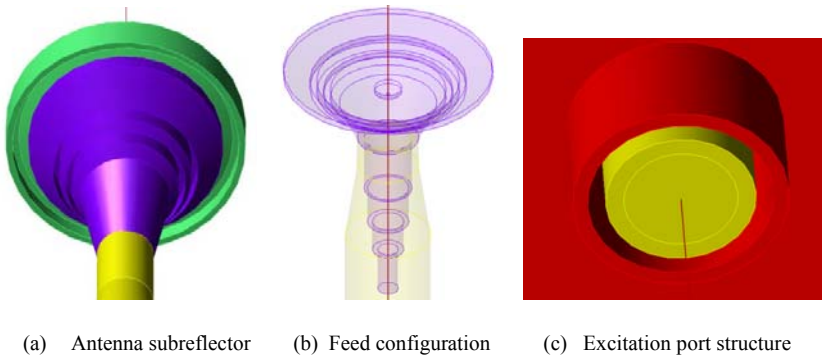


Figure 6.20 Feed system configuration and the detailed structure (a) antenna subreflector, (b) feed configuration, and (c) excitation port structure.

The diameter of the main reflector is 700 mm, which is about 49 wavelengths at 21 GHz. The fill-in material inside the feed coaxial cable is Teflon (dielectric constant is 2.08). Hence, the wavelength inside the fill-in dielectric is shorter than that in free space.

The antenna dimension in the vertical direction is 316 mm, that is, about 22 wavelengths. The fine structure dimension in both the horizontal and vertical directions is 0.4 mm, as shown in Figure 6.21.

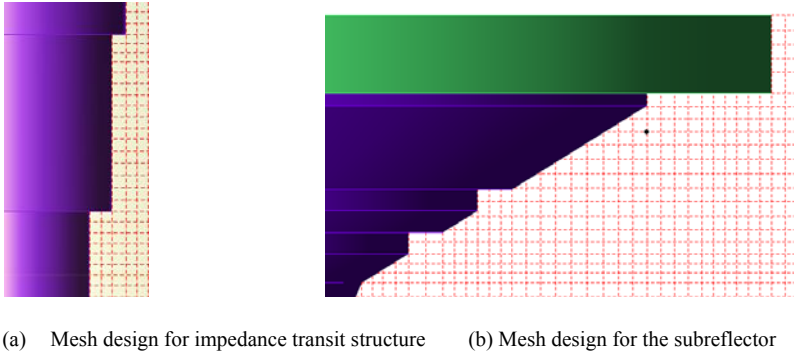


Figure 6.21 Mesh distribution in the fine structure areas. (a) Mesh captures the fine structure in the horizontal direction, and (b) mesh captures the fine structure in the vertical direction.

The basic mode supported by the coaxial cable is the transverse electromagnetic (TEM), therefore, we first use the TEM mode to excite the antenna and calculate its return loss and far-field pattern. If we use the minimum cell size to be 0.4 mm in the three directions and the ratio between adjacent cells to be 1.05, the total number of cells will be $1,450 \times 1,450 \times 930$ (1,955 Mcells) that requires at least 40 GB of memory. Using its symmetric property, we can only simulate the quarter domain and then reduce the problem size to the one quarter of the original problem. The return loss and 2-D and 3-D directivity patterns are shown in the Figures 6.22 to 6.24.

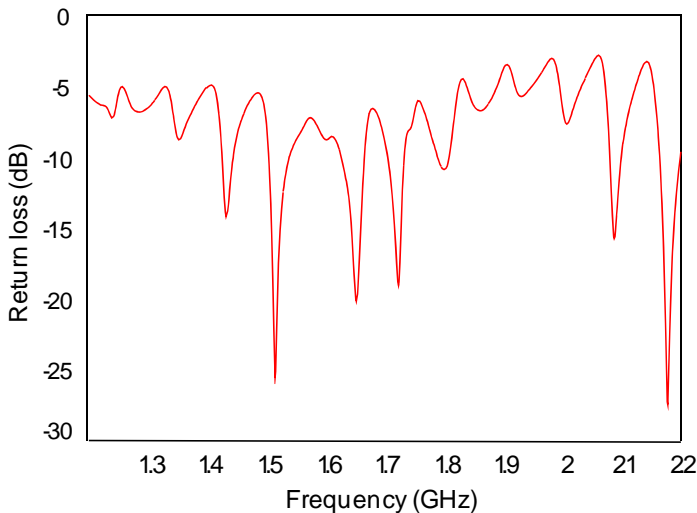


Figure 6.22 Return loss of the high gain antenna when the TEM mode excitation is applied.

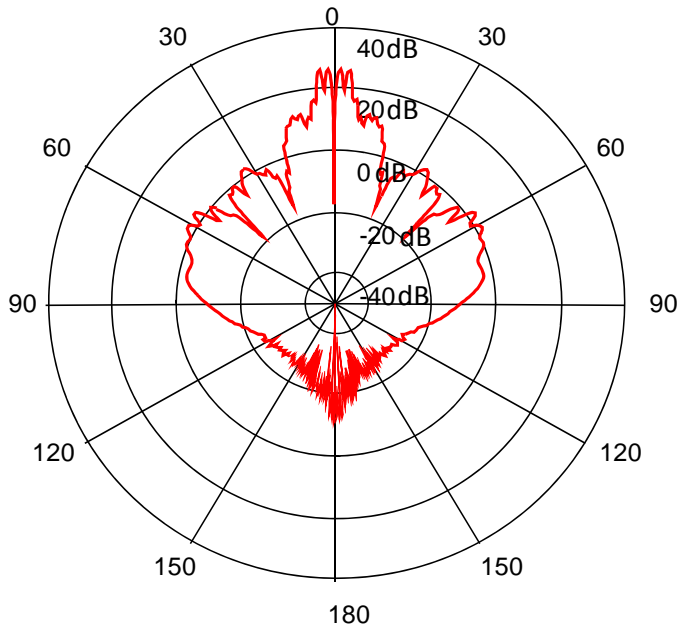


Figure 6.23 Directivity pattern of the high gain antenna at 20.74 GHz for the TEM mode excitation.

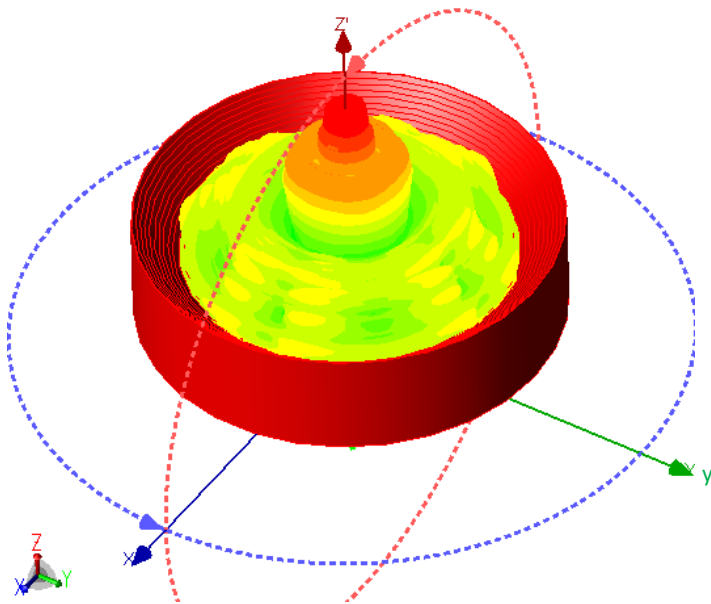


Figure 6.24 The 3-D directivity pattern of the high gain antenna at 20.74 GHz for the TEM mode excitation.

Although we do not show the experimental results here, the return loss and far-field pattern match well with what the designer expects. Next, we use TE_{21} mode [3, 12] to excite the reflector antenna, as shown in Figure 6.25(a). The TE_{21} mode is also symmetric about the x - and y -axes, and the one-quarter mode pattern is shown in Figure 6.25(b).

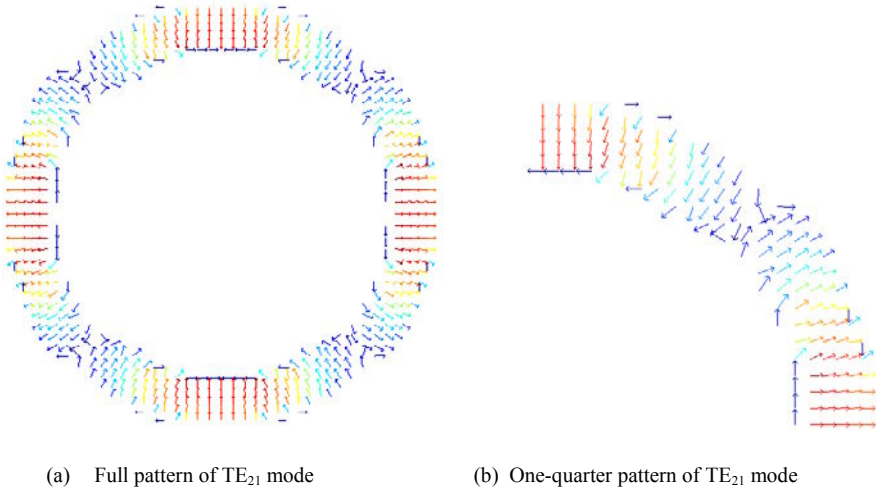


Figure 6.25 (a) Full and (b) one-quarter patterns for the TE_{21} mode excitation extracted by the numerical method FDFD.

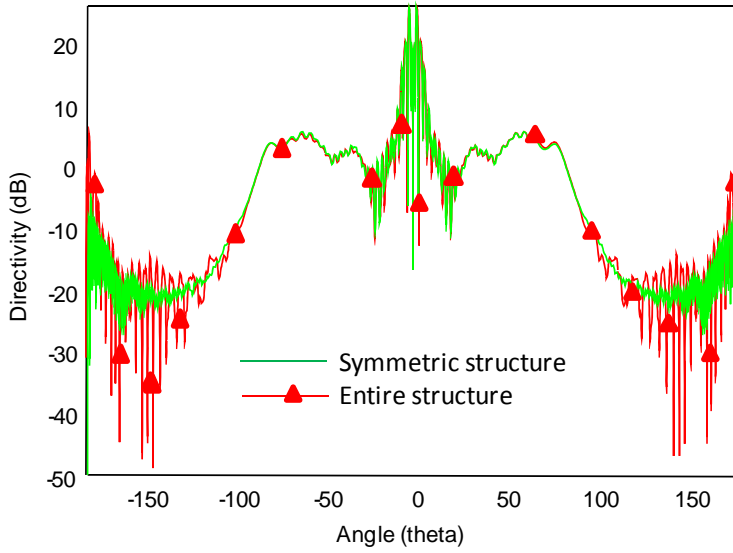
TE_{21} mode has two degenerated patterns and the cutoff frequencies of the two degenerated modes are 11.93 and 12.22 GHz, respectively. The two modes will generate similar far-field patterns and return loss. However, the far-field patterns will rotate 45 degrees for two degenerated excitation modes.

The TEM mode pattern in the coaxial cable can be expressed in the analytic formula. However, the electric field pattern of TE_{21} mode cannot be obtained by the analytic method. The finite-difference frequency-domain (FDFD) method can be used to extract the TE_{21} mode patterns numerically [3].

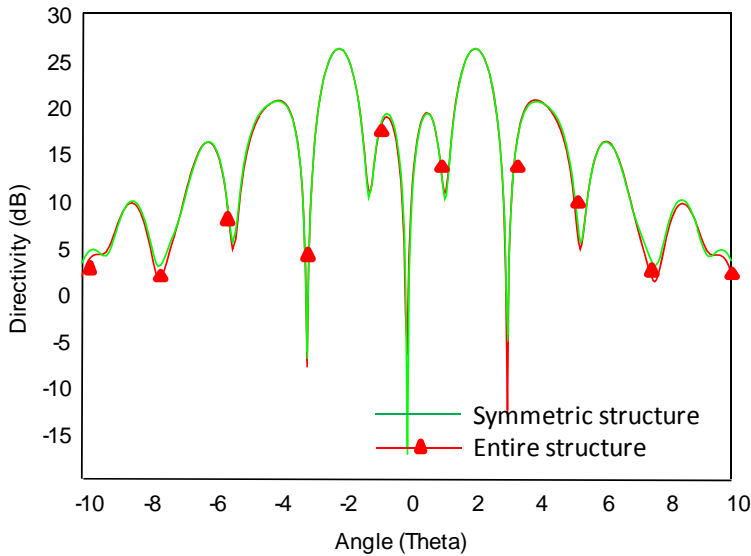
In order to avoid the interpolation operations, we use the same mesh in both the FDTD and FDFD methods. Unlike the FDTD method, the FDFD method requires solving the matrix equations that have lower efficiency than the FDTD method. If the cross section is larger, the mode extraction procedure will take a while.

In order to show the accuracy of the symmetric simulation method, we simulate both the full structure and one-quarter geometry. The directivity patterns at $f=20.74$ GHz and in $\phi=0^\circ$ plane for the symmetric and full structures are shown in Figure 6.26. The difference only happens in the very low level due to the different mesh distributions, namely, excellent agreement is between the

symmetric and full structures. The return loss for the symmetric and full structures is shown in Figure 6.27.



(a) Directivity patterns in the range $\theta = -180^\circ$ to 180°



(b) Directivity patterns in the range $\theta = -10^\circ$ to 10°

Figure 6.26 Directivity patterns in $\phi = 0^\circ$ plane at $f = 20.74$ GHz for the full and one-quarter structures (a) in the range $\theta = -180^\circ$ to 180° and (b) in the range $\theta = -10^\circ$ to 10° .

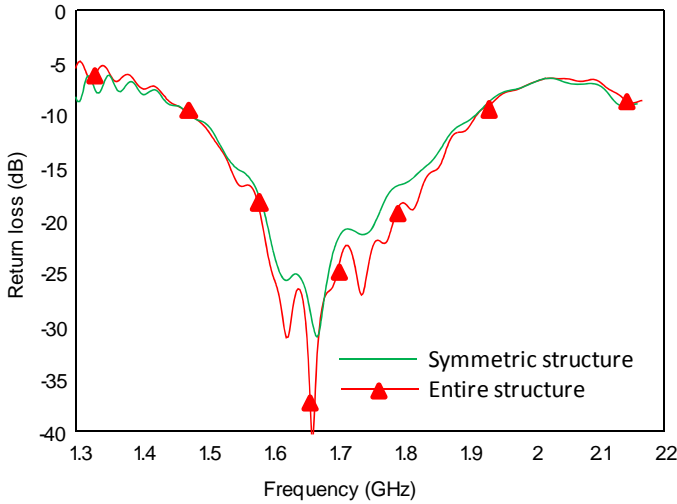


Figure 6.27 Return loss of high gain antenna for the full and quarter structures.

The directivity patterns at $f=20.74$ GHz and $\phi=90^\circ$ for the one-quarter and full structures are shown in the Figure 6.28. It is evident from Figure 6.28 that there is excellent agreement between the one-quarter and full structures. Since the excitation mode is not symmetric, it is necessary for us to check the far-field pattern at the different ϕ . The directivity patterns in $\phi=0^\circ$, 45° , 90° , and 135° planes at $f=20.74$ GHz are plotted in Figure 6.29. It is obvious from Figure 6.29 that the different patterns in the $\phi=0^\circ$ and 90° planes from those in the $\phi=45^\circ$ and 135° planes.

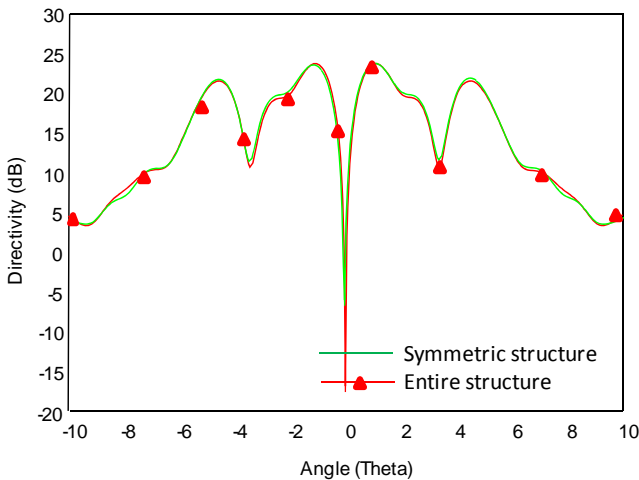


Figure 6.28 Directivity patterns in $\phi=90^\circ$ plane at 20.74 GHz for the full and one-quarter structures.

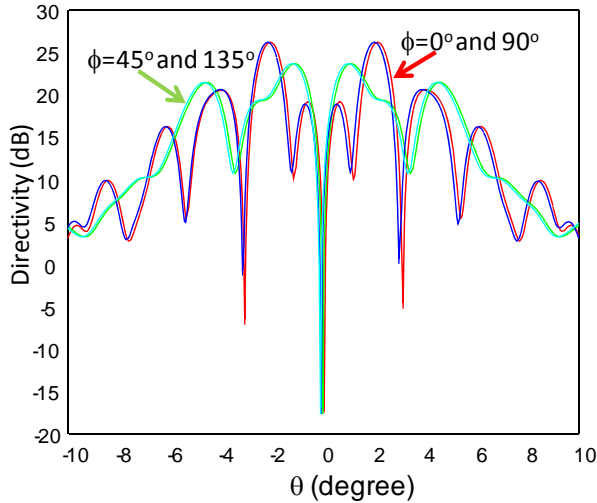


Figure 6.29 Directivity patterns in $\phi=0^\circ$, 45° , 90° , and 135° planes at 20.74 GHz.

6.4 PARTIAL SYMMETRIC PROBLEM SIMULATION

A reflector antenna fed by a patch array that includes 32 patch elements, as shown in Figure 6.30. The diameter of the reflector is 12,000 mm and the distance from the feed array to the reflector is 6,600 mm. The frequency band of interest is from 0.5 to 2 GHz. Because the fine structures are involved in the feed support and patch array, it is very hard or impossible to simulate it using a single computer. The feed structure is a patch array, and hence, it cannot be directly simulated by using the symmetric method. In the regular symmetric simulation method, the excitation ports should be at least four (if it is not on the axis) if we use the PEC or perfect magnetic conductor (PMC) boundary to truncate the structure. In order to calculate the S-parameters, we need to excite one port at one time and terminate all other ports using the match loads. To this end, we need to simulate the symmetric structure four times to remove the contribution from the three images in the near field and far fields. We can combine the results from the four simulations to get the final result that is generated by the one-port excitation. To validate this method, first we simulate the entire patch array (without the reflector) that is excited by using a single port and the rest of the ports are terminated by using the match loads, as shown in Figure 6.31. Then we simulate the one quarter of the structure four times, in which the boundary condition is different in each simulation, for example, PEC at the X_{min} and Y_{min} walls in the first simulation, PMC at the X_{min} and Y_{min} walls in the second simulation, PEC at the X_{min} wall and PMC at the Y_{min} wall in the third simulation, and PMC at the X_{min} wall and PEC at the Y_{min} wall in the fourth simulation, as shown in Figure 6.32.

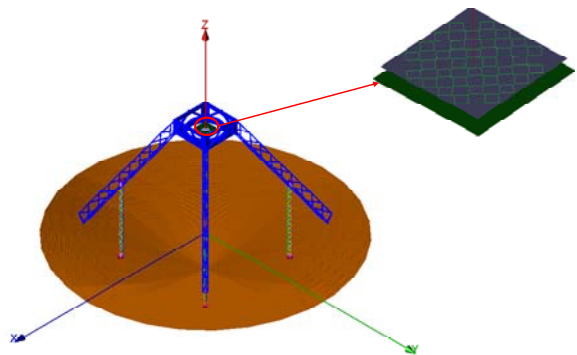


Figure 6.30 Reflector antenna with feed support structure fed by the 32-element patch array.

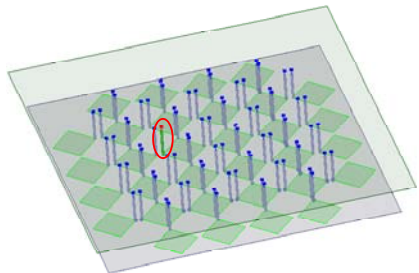


Figure 6.31 Patch array with one-port excitation and the rest of the ports terminated by using the match loads.

We can get the port voltage and current directly by calculating the vector summation of the four results obtained in the four simulations and divided by a factor 4, as shown in Figure 6.32. However, for the S-parameters and far-field patterns, we cannot simply calculate them by adding the four results together. For the S-parameters, we need to calculate the total port voltages and currents, and then calculate the S-parameters from the total voltages and currents. For the far-field patterns, we need to calculate one far-field component at each time.

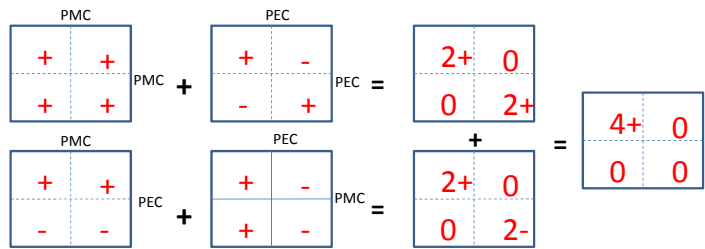


Figure 6.32 Excitation and its images using the different boundary conditions.

In the patch array, each patch is connected to two probes, and the excitation source or match loads are located in the small gap between the probe and ground plane. The measured port voltages and currents at each port are used to calculate the S_{11} using the following formula [12]:

$$S_{11} = \frac{V/\sqrt{Z_0} - I\sqrt{Z_0}}{V/\sqrt{Z_0} + I\sqrt{Z_0}} \quad (6.1)$$

where Z_0 is the normalized port impedance. Suppose that the port voltages and currents in the four simulations are $V_1, V_2, V_3, V_4, I_1, I_2, I_3,$ and I_4 , the total voltage and current can be expressed as follows:

$$\begin{aligned} V &= (V_1 + V_2 + V_3 + V_4)/4 \\ I &= (I_1 + I_2 + I_3 + I_4)/4 \end{aligned} \quad (6.2)$$

S_{11} obtained from the entire structure simulation and the combination of the four separate simulations are plotted in Figure 6.33.

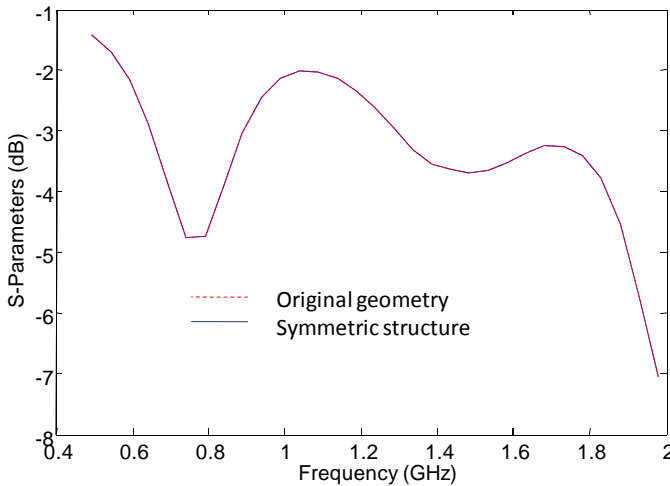


Figure 6.33 S_{11} obtained from the entire structure simulation and four symmetric simulations.

Follow a similar procedure to calculate the far-field patterns. As mentioned above, we need to calculate the summation of E_θ and E_ϕ , respectively. The slight difference between the entire and symmetric structure simulations is caused by the spatial downsampling used in the far-field pattern calculation, as shown in Figure 6.34. It is evident from Figure 6.34 that we can get the solution of the large problems without simulating the entire structure if the geometry is symmetric.

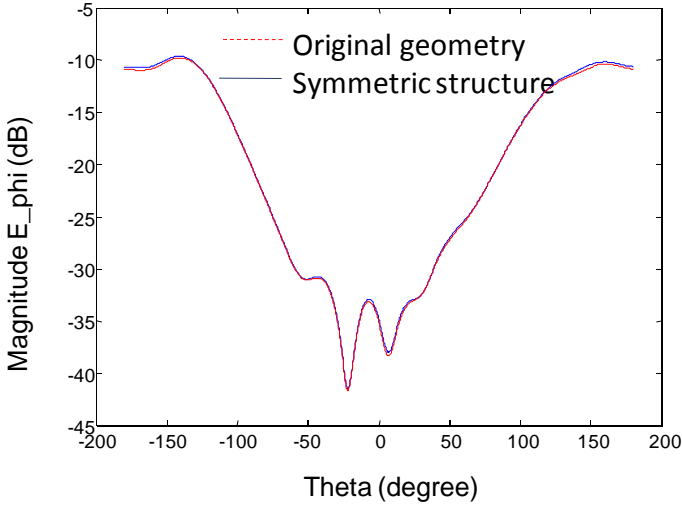
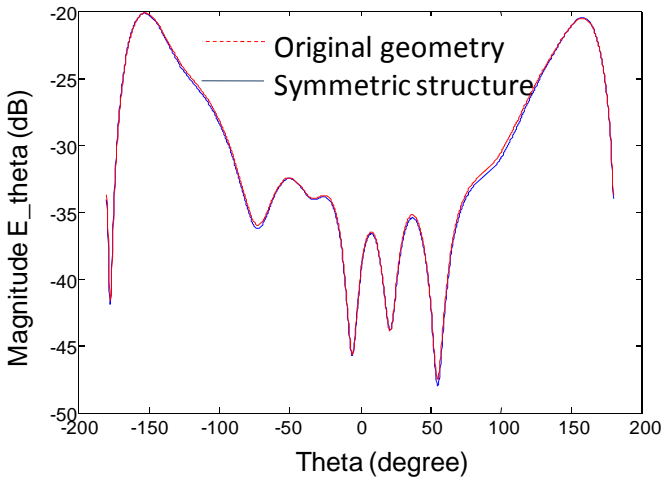
(a) E_ϕ pattern in the $\phi=0^\circ$ plane(b) E_θ pattern in the $\phi=0^\circ$ plane

Figure 6.34 Far-field patterns: (a) E_ϕ pattern in the $\phi=0^\circ$ plane and (b) E_θ pattern in the $\phi=0^\circ$ plane from the entire structure simulation and four symmetric simulations.

Next, we place the patch array at the focal point of the reflector antenna, as shown in Figure 6.35. Only one quarter of the antenna structure is included in the computational domain. The excitation and output ports are marked in Figure 6.36. We simulate one-quarter of antenna structure four times using the different boundary conditions and get the S-parameters and far-field patterns of the original

antenna problem, as shown in Figure 6.37. As expected, the far-field patterns are not symmetric because only one port is excited that is not symmetric.

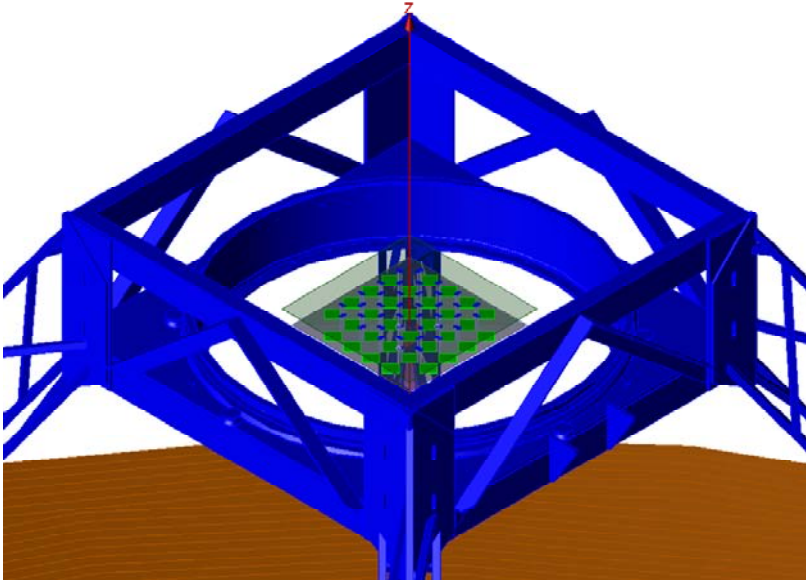


Figure 6.35 Relative location of patch array in the reflection antenna system, and only one-quarter of the antenna structure is included inside the computational domain.

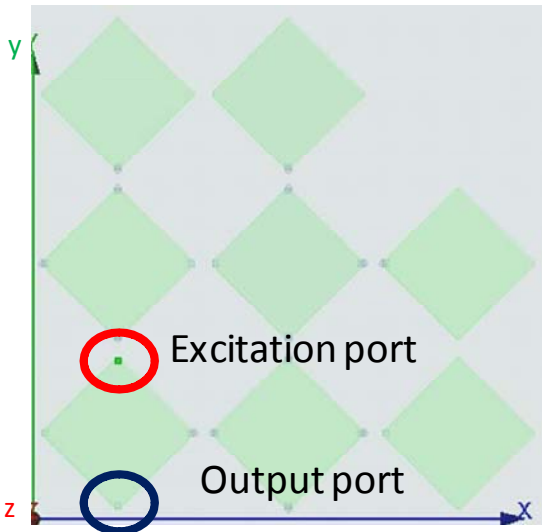
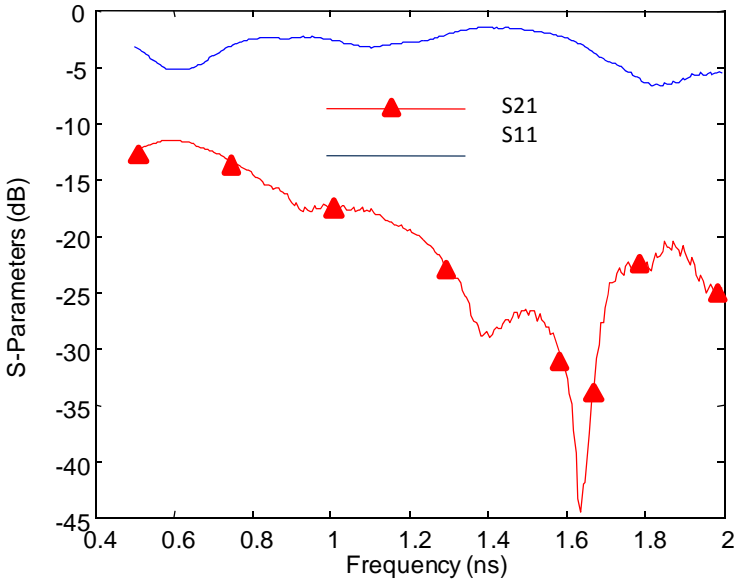
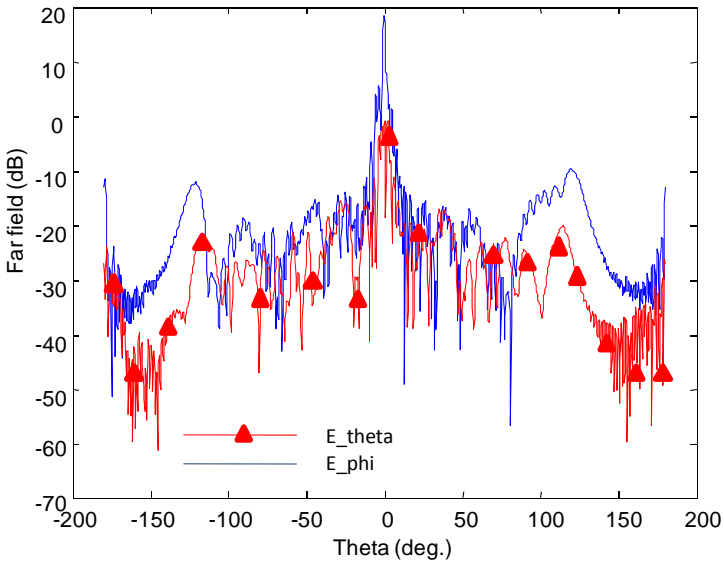


Figure 6.36 Excitation and output ports in the symmetric simulation.



(a) S-parameters when the patch array is used as an antenna feed



(b) Far-field patterns of E_θ and E_ϕ in $\phi=0^\circ$ plane

Figure 6.37 (a) S-parameters and (b) far-field pattern obtained from the symmetric structure.

6.5 FLAT REFLECTOR ANTENNA

A flat reflector antenna is fed by a rectangular horn that is excited by using the TE_{10} mode, as shown in Figure 6.38.

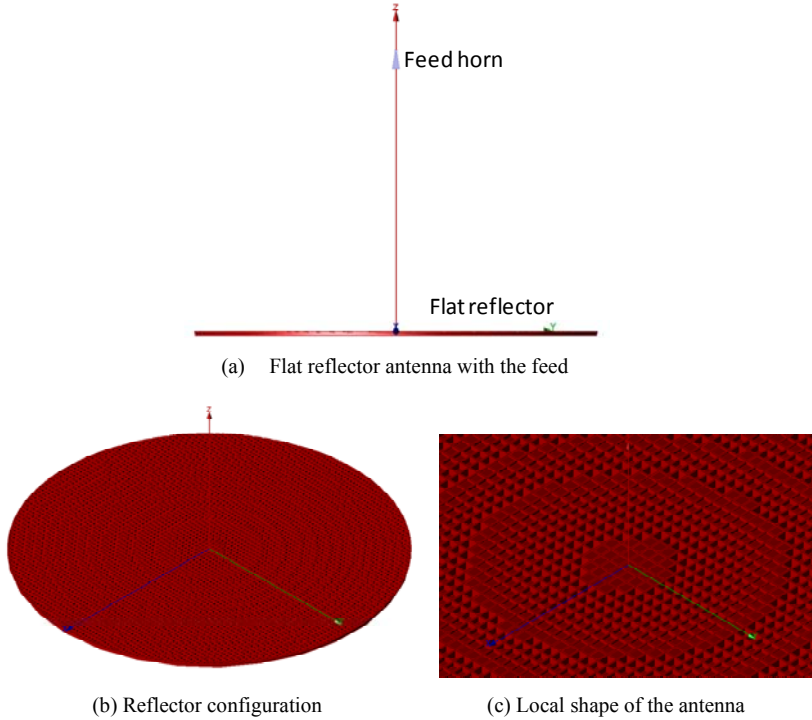


Figure 6.38 (a) Flat reflector antenna configuration, (b) the pattern of the holes, and (c) the zoomed hole pattern. The holes have different depth to force the wave to focus on the focal point.

The diameter of the reflector is 300 mm and the distance from the feed horn to the reflector is 210 mm. The frequency band of interest is 71 to 86 GHz. It is very hard to simulate it using a single computer since it is a relatively electrically large problem ($86 \times 86 \times 60$ wavelengths). Since the structure is symmetric about the x - and y -axes, we only need to simulate one-quarter of the structure and can get the solution of the original problem. The PEC and PMC boundary at the X_{min} and Y_{min} walls are determined by the polarization of the excitation. For example, if the excitation is E_y inside the horn, the PMC and PEC boundary should be applied to the X_{min} and Y_{min} walls, respectively.

In the original design, the feed horn has a finite thickness that is 0.01 mm. Since the thickness of the horn does not affect much on both the return loss and far-field pattern, we can ignore it to reduce the memory requirement and

simulation time. Though the feed port is open in the model, it is terminated by a match load in the practical applications. Here we use a match load to terminate the feed port and then add the TE_{10} mode excitation. The feed horn configuration and 3-D far-field pattern generated by the feed horn only are shown in Figure 6.39.

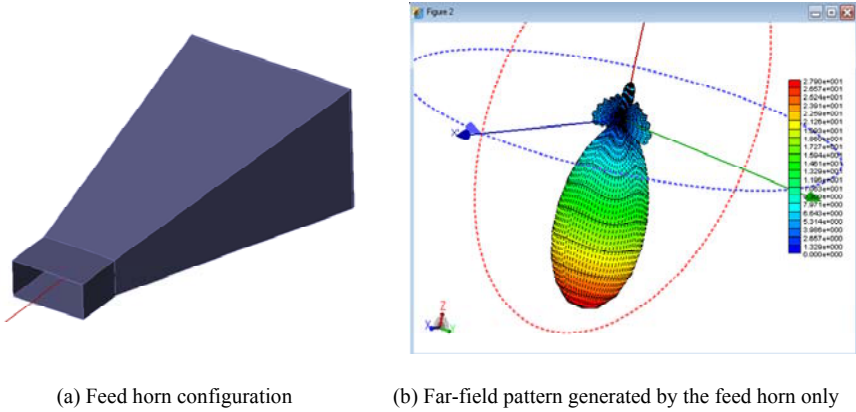


Figure 6.39 (a) Feed horn configuration used for the excitation feed and (b) 3-D far-field pattern of the horn antenna.

The output parameters include the aperture field distributions above the feed horn and underneath reflector antenna, as shown in Figure 6.40, return loss, and gain pattern.

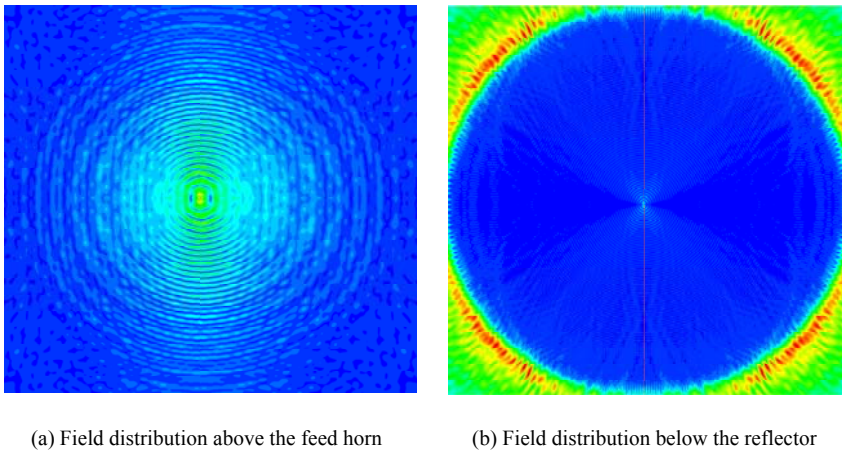


Figure 6.40 Aperture field distributions measured (a) above the feed horn and (b) below the reflector.

For the sake of comparison, we plot the return loss for the feed horn only and feed horn with the reflector in Figure 6.41. The gain patterns at 73.5 and 75 GHz are plotted in Figures 6.42 and 6.43.

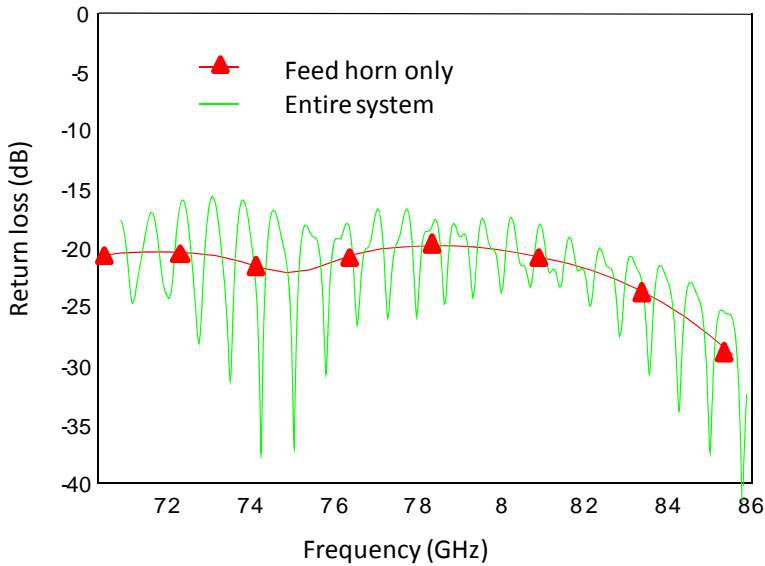


Figure 6.41 Comparison of the return loss with and without the reflector.

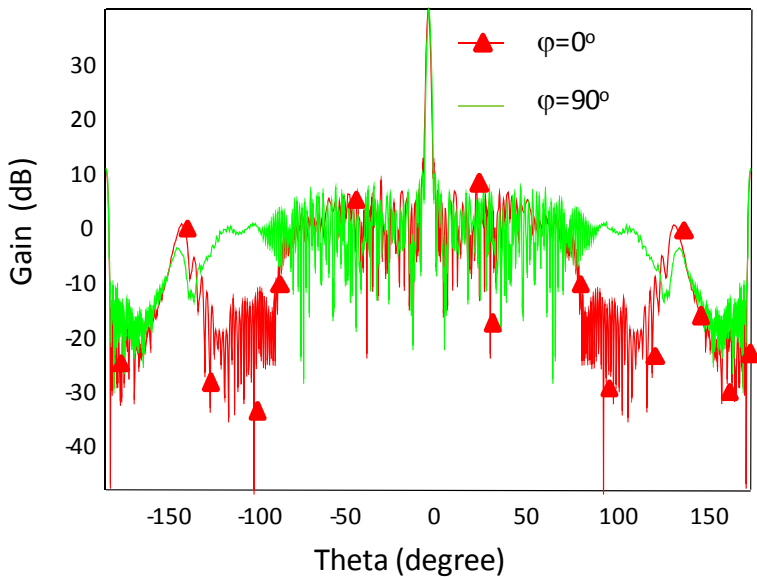


Figure 6.42 Gain patterns of the reflector antenna at $\phi=0^\circ$ and 90° at 73.5 GHz.

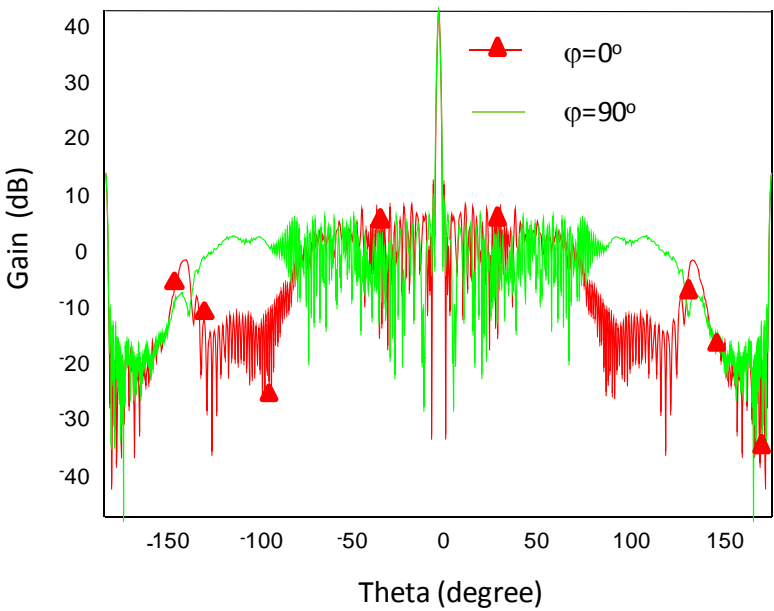


Figure 6.43 Gain patterns of the reflector antenna at $\phi=0^\circ$ and 90° at 75 GHz.

The gain values and half-power beamwidth (HPBW) at 73.5 and 75 GHz obtained by the FDTD simulation and measurement are shown in Table 6.1. It is evident from Table 6.1 that the FDTD simulation results are in excellent agreement with the measurement data.

Table 6.1 Simulation and Measurement Gain and HPBW

	HPBW (75 GHz)	HPBW (73.5 GHz)	Gain (75 GHz)	Gain (73.5 GHz)
Simulation	1.08°	1.4°	41.988 dB	39.588 dB
Measurement	1°	N/A	42 dB	39.5 dB

6.6 SIGNAL INTEGRITY

In this example, we will use the parallel conformal FDTD method to simulate six parallel strip lines having a trapezoidal cross section, which are embedded in an anisotropic material. The thicknesses and dielectric constants of the four layers of dielectrics and the dimensions of the strip lines are marked in Figure 6.44. The strip lines are copper and the structure is backed by a PEC ground plane. The

dimensions of the dielectric layers and the ground plane in the horizontal directions are $16 \text{ mm} \times 16 \text{ mm}$.

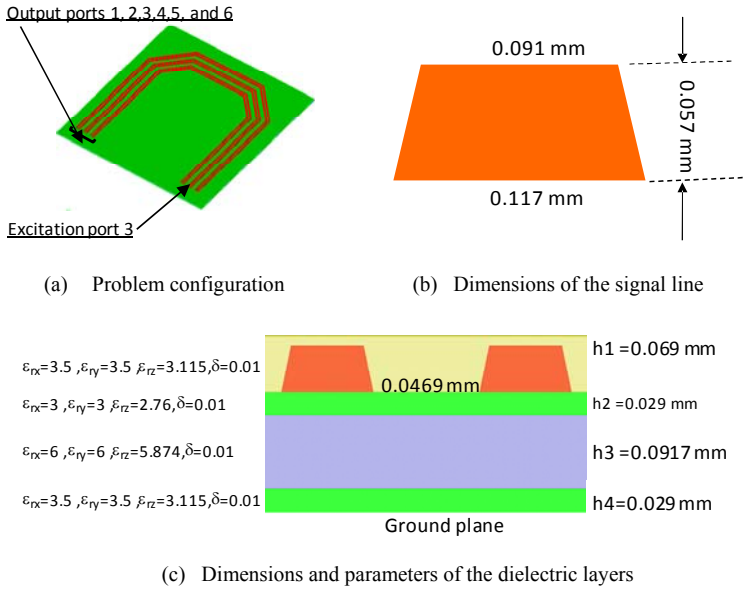


Figure 6.44 (a) Problem configuration, (b) material parameters, and (c) dimensions for a circuit consisting, in part, of six parallel strip lines.

The fine structure in the problem geometry is the thickness of the adhesive material (0.012 mm) and the gap (0.029 mm) between the strip lines and the top surface. Hence, the minimum cell size in the vertical direction is taken to be 0.012 mm. The minimum dimension in the horizontal direction is the width of the strip lines. Since part of the strip line is slanted, we need to use at least two cells inside the strip line to ensure that the strip line is not broken in the slanted part. The domain is discretized into 794,000 ($254 \times 194 \times 15$) cells. The mesh distribution across the strip lines is shown in Figure 6.45. The slanted strip lines and the trapezoidal cross section are modeled using the conformal technique.

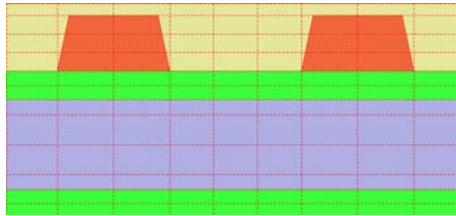
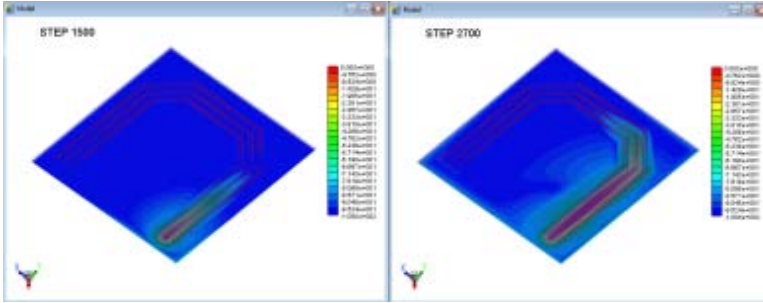


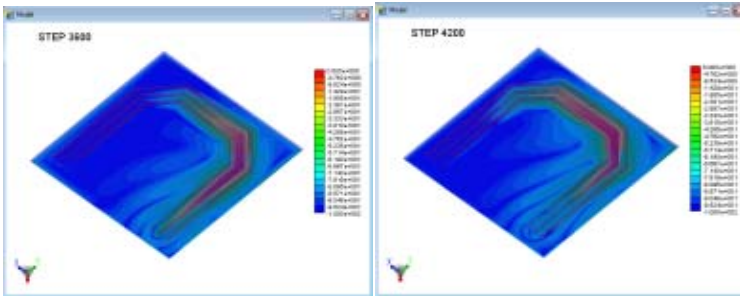
Figure 6.45 Local mesh distribution across the strip lines and the grids to capture the key points on the strip line and the interfaces of the dielectric layers.

The complex frequency shifted–perfectly matched layer (CFL-PML) [5] only requires two cells of white space to achieve an accurate result. The time-domain field distributions inside the top dielectric layer are shown in Figure 6.46 when the third strip line is excited and others are terminated using 50 Ω matched loads. It is worth mentioning that the excitation pulse is a pure Gaussian pulse whose 3-dB beamwidth frequency is 20 GHz.



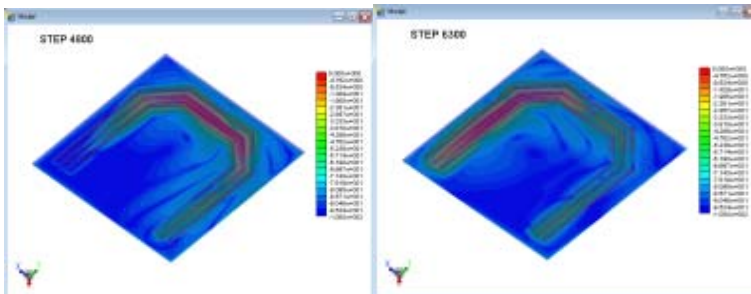
(a) Field distribution at 0.054 ns

(b) Field distribution at 0.097 ns



(c) Field distribution at 0.122 ns

(d) Field distribution at 0.151 ns

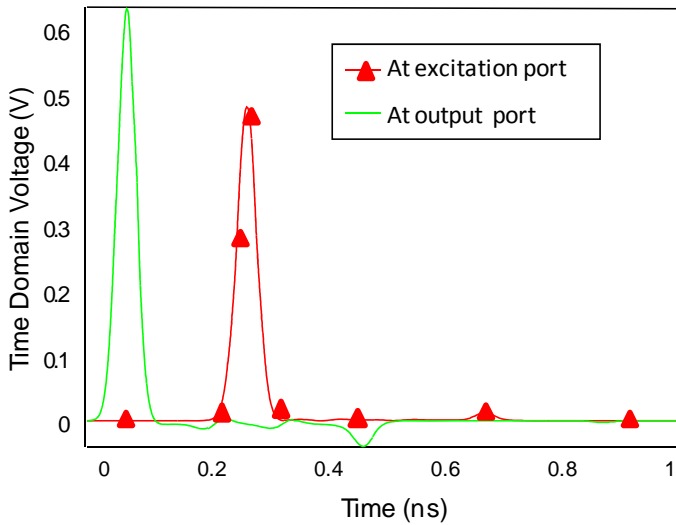


(e) Field distribution at 0.173 ns

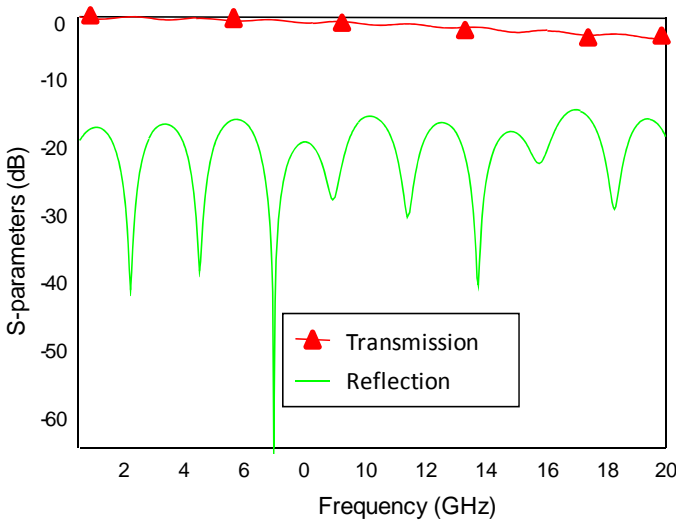
(f) Field distribution at 0.227 ns

Figure 6.46 Time variation of the electric field distribution inside the dielectric layer (a) $t=0.054$ ns, (b) $t=0.097$ ns, (c) $t=0.122$, (d) $t=0.151$ ns, (e) $t=0.173$ ns, and (f) $t=0.227$ ns.

The time-domain voltages measured at the excitation and output ports are plotted in Figure 6.47(a). The reflection and transmission coefficients at the two ports of the excited line are plotted in Figure 6.47(b). The voltages measured at the six output ports are plotted in Figure 6.48(a). The reflection and transmission coefficients measured at the six output ports are plotted in Figure 6.48(b).

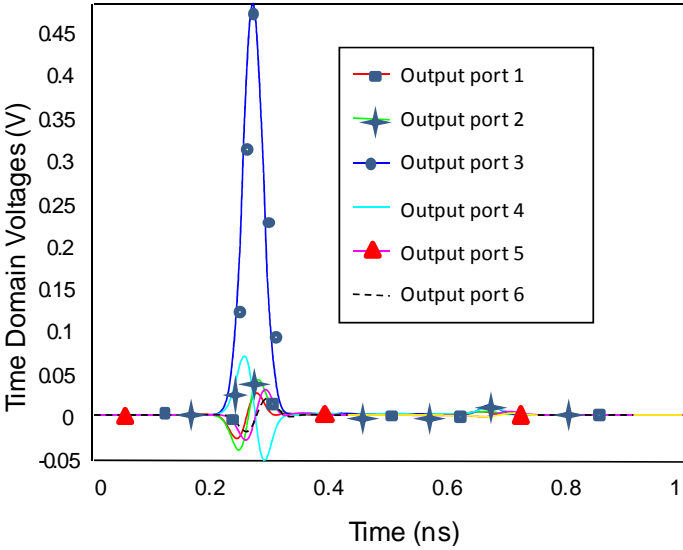


(a) Time-domain voltages at the two ports of the third strip line

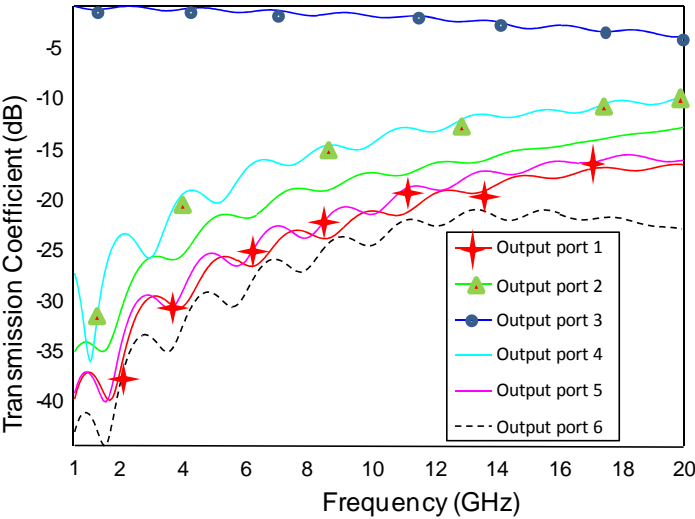


(b) Reflection and transmission coefficients

Figure 6.47 (a) Time-domain signals and (b) reflection and transmission coefficients at the two ports of the excited strip line.



(a) Time-domain voltages measured at the six output ports



(b) Transmission coefficients at the six output ports

Figure 6.48 (a) Time-domain signals and (b) transmission coefficients measured at the six output ports when the third strip line is excited.

6.7 CARD READER EMI ANALYSIS

Configuration of a card reader is shown in Figure 6.49. In this example, we need to calculate the field strength behind the card reader coupled from the outside when a plane wave is incident from the z -direction. Since the card reader is installed in the electronic device (say, a computer), the card reader is surrounded by the PEC walls, and hence, the power can only come to the card reader through the card inset slot in the front face, as shown in Figure 6.49. The coupled power needs to pass the front slot and signal connecting pieces to reach the surface behind the card, as shown in Figure 6.50.

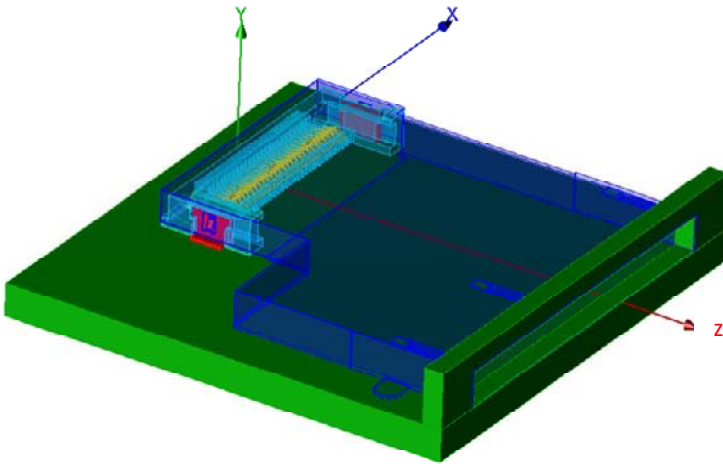


Figure 6.49 Configuration of the card reader and the power penetrating through the front slot of the card.

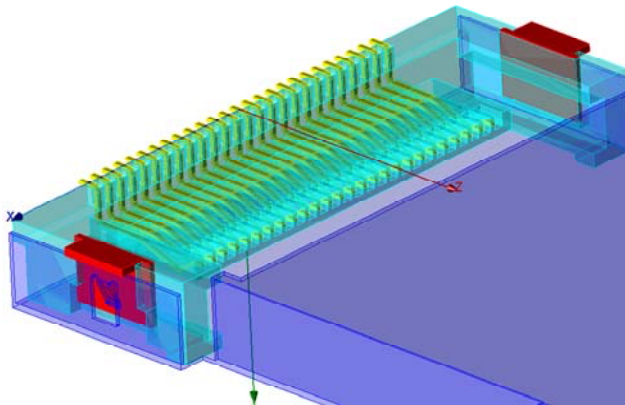


Figure 6.50 Configuration of the connecting pieces between the card reader and board on which the card will set.

The absorbing boundary is applied to truncate four walls without the white space in the $-x$ -, $-y$ -, x -, and y -directions, and with the white space in the $-z$ - and z -directions. Since there is no white space in the $-x$ -, $-y$ -, x - and y -directions, there is no direct power going through the card reader except through the slot, as shown in Figure 6.51.

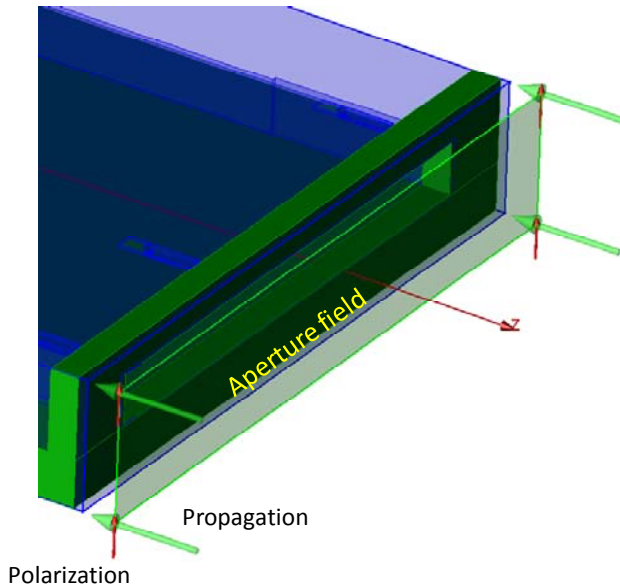


Figure 6.51 Incident plane wave illuminating on the card reader with the vertical polarization.

In this example, the frequency band of interest is from 10 MHz to 1 GHz. The differential Gaussian pulse is used as the excitation pulse and the 3-dB beamwidth is 1 GHz. The fine structure inside the card reader is the connecting piece, as shown in Figure 6.52, and the minimum dimension is 0.4 mm. The minimum cell size is taken to be 0.2 mm, namely, there are two cells across the connecting piece. The mesh size ratio of the adjacent cells is not important; however, the interesting region or structures should be meshed properly. During the mesh generation, if we need to describe the field exactly, we can define some key points on the important position and ensure that the mesh can capture the field variation at the important points.

The output parameter in this example is the coupling level of the power through the card reader. We define an observation surface behind the card, as shown in Figure 6.53. In the FDTD simulation, we calculate the field distribution on the observation surface at the specified frequencies. The field distribution at 1 GHz is shown in Figure 6.54. We search and store the maximum value on the surface. When the simulation is convergent, we output the maximum values on the observation surface into a data file and plot it in Figure 6.55.

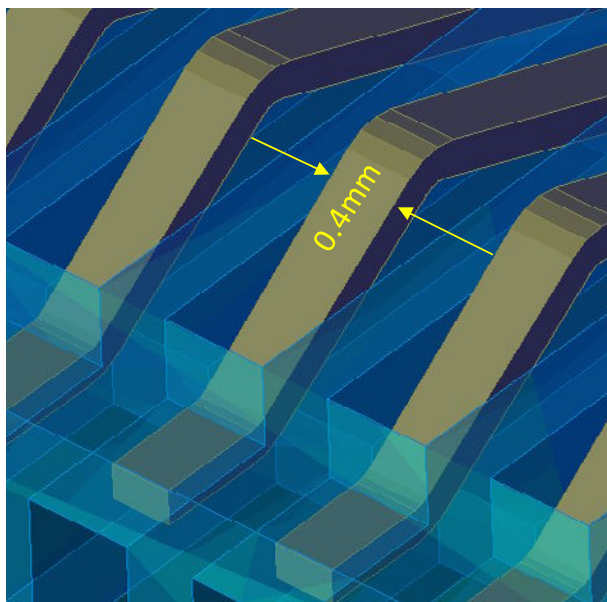


Figure 6.52 Fine features (connecting pieces) in the card reader; the minimum mesh selection is based on these small pieces.

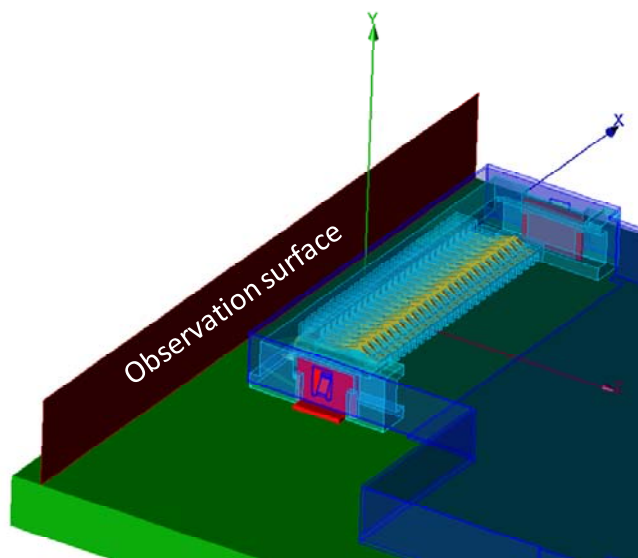


Figure 6.53 Observation surface behind the card reader on which the electric field distribution will be calculated.

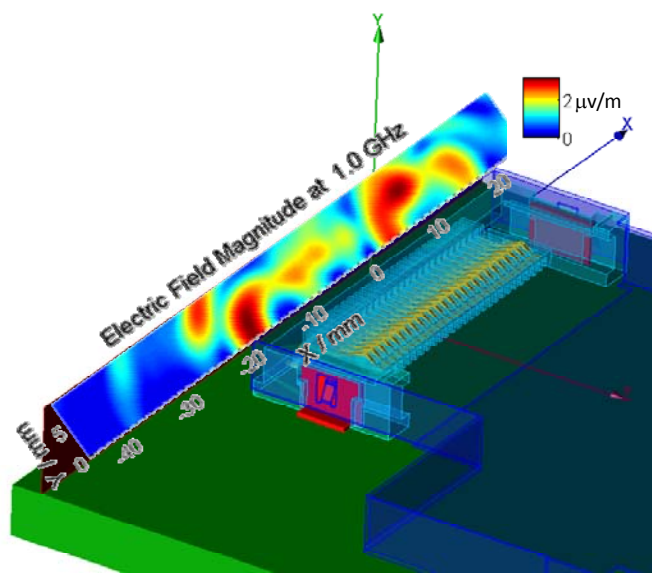


Figure 6.54 Electric field distribution on the observation surface at 1 GHz.

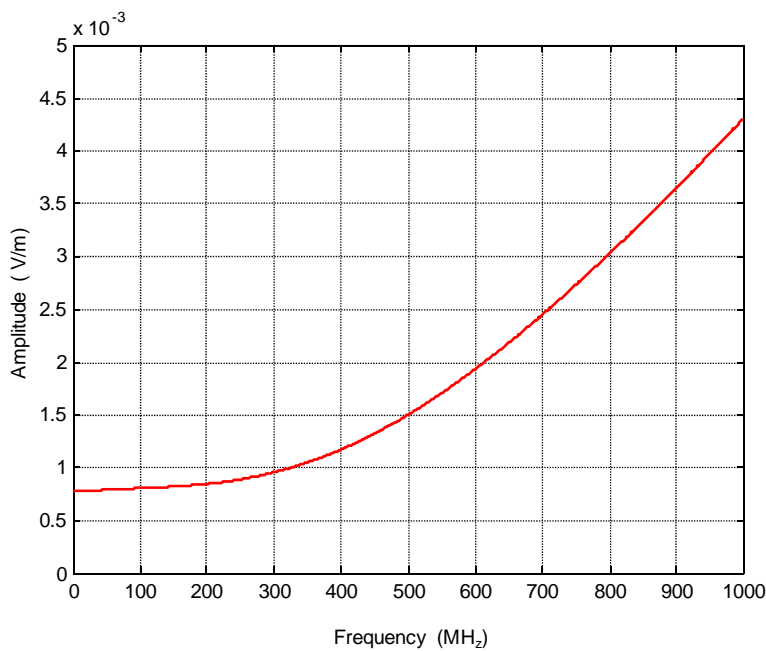


Figure 6.55 Electric field strength varies with the frequencies on the observation surface.

6.8 CLOAKING STRUCTURE

The cloaking structure [14–16] consists of a copper kernel and 10 layers of open split rings, as shown in Figure 6.56. The relative permittivity, loss tangent, and thickness of the dielectric layers on which the rings are mounted are 2.33, 0.001555, and 0.38 mm, respectively. The width and length of the open split rings are $3,985\text{ }\mu\text{m}$ and $3,000\text{ }\mu\text{m}$, respectively. The width of the ring strip is $200\text{ }\mu\text{m}$. The radius of the cylindrical kernel is 25 mm. We can take advantage of its symmetry and only simulate half structure to reduce the computational burden, although the problem still remains large, as is evident from Table 6.2.

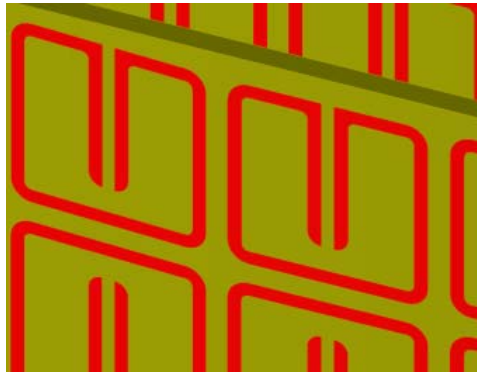
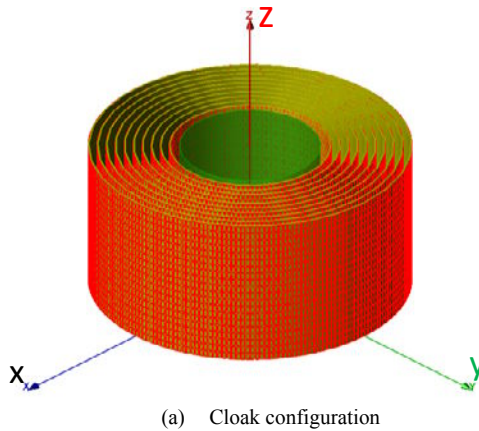


Figure 6.56 (a) Cloaking model with a copper kernel and 10 layers of open split rings. (b) The 10-layer open rings are mounted on 10 thin dielectric layers, respectively.

Table 6.2 Simulation Details for the Cloaking Structure

Options	Description
Hardware platform	IBM BlueGene/P
Polarization	E_θ
Number of unknowns	1.166 billion cells (792×1,766×834)
Excitation	Plane wave source
Incident direction ($\theta^{inc}, \varphi^{inc}$)	(135°, 270°)
Boundary	Six-layer PML
Symmetrical boundary	Applied
Memory usage	40.3 GB
Number of CPUs	510
Number of time steps	12,000 (convergent to −30 dB)

This problem has not been simulated in the past, except by using a simplified equivalent model for the structure. However, such a model does not take into account the anisotropic nature of the split-ring elements that form the cloak, and what is needed to accurately predict its performance for the case of oblique incidence. Note that the actual physical structure is much too large to be handled by a single processor, for both the normal as well as oblique incidence cases.

For our simulation, we assume an illuminating plane wave that is incident on the cloaking structure at an angle of $\theta=135^\circ$. We observe from Figure 6.57 that there is low reflection from the cloaking structure as the plane wave hits the cloaked surface. We observe, therefore, the performance of the cloaking structure, which has been designed for normal incidence, is less than ideal when the angle of incidence is oblique. Incidentally, this is also true when the frequency of the incident wave differs from the one for which the cloak was originally designed. We point out, once again, that it is not possible to predict the performance of the cloaking structure when operating under general conditions by using the effective medium approach.

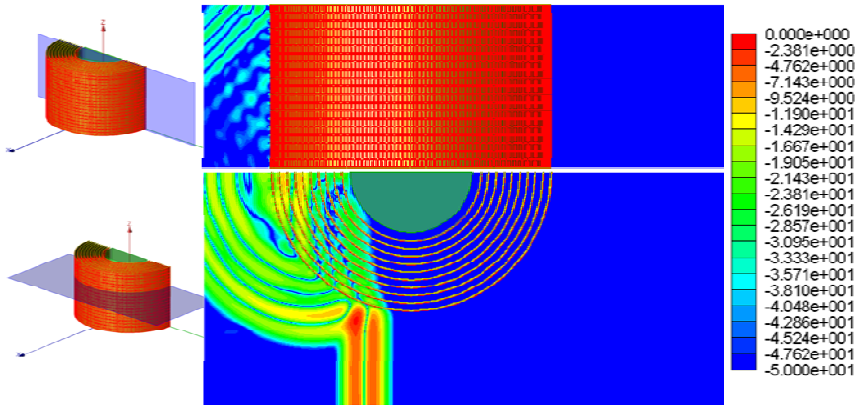


Figure 6.57 Scattering field pattern in the cloaking structure in the decibel scale when illuminated by a plane wave at oblique incidence.

References

- [1] A. Taflove and S. Hagness, (eds.), *Computational Electromagnetics: The Finite-Difference Time-Domain Method*, 3rd ed., Norwood, MA: Artech House, 2005.
- [2] W. Yu and R. Mittra, *Conformal Finite-Difference Time-Domain Maxwell's Equations Solver: Software and User's Guide*, Norwood, MA: Artech House, 2004.
- [3] W. Yu, et al., *Parallel Finite-Difference Time-Domain Method*, Norwood, MA: Artech House, 2006.
- [4] W. Yu, et al., *Electromagnetic Simulation Techniques Based on the FDTD Method*, New York: John Wiley and Sons, 2009.
- [5] IEEE Standard 1528 Tutorials, www.ices-emsafety.org/documents/200906_Attachment_3.pdf.
- [6] P. Hill and Y. Hao, *Antenna Propagation Body-Centric Wireless Network*, Norwood, MA: Artech House, 2006.
- [7] P. Li, et al., "A 4-Element Ultra-Wideband Taper-Slot-Fed Antenna Array," *Proc. IEEE Int. AP-S Symp. Dig.*, 2006, pp. 4475–4478.
- [8] S. Sugawara, et al., "A mm-Wave Taper Slot Antenna with Improved Radiation Pattern," *IEEE MTT-S Int. Microw. Symp. Dig.*, 1997, pp. 959–962.
- [9] H. Sato, et al., "Broadband FDTD Analysis of Fermi Antenna with Narrow Width Substrate," *IEEE AP-S Int. Symp., Dig.*, Vol. 1, 2003, pp. 261–264.
- [10] H. Sato, K. Sawaya, Y. Wagatsuma, and K. Mizuno, "Design of Narrow-Width Fermi Antenna with Circular Radiation Pattern," *IEEE AP-S Int. Symp. Dig.*, Vol. 4, June 2004, pp. 312–315.
- [11] M. Kishihara, K. Yamane, I. Ohta, and T. Kawai, "A Design of Multi-Stage, Multi-Way Microstrip Power Dividers with Broadband Properties," *Proc. IEEE MTT-S, Int. Microw. Symp. Dig.*, June 2004, pp. 69–72.
- [12] D. M. Pozar, *Microwave Engineering*, 2nd, ed., New York: John Wiley & Sons, 1998.
- [13] C. Balanis, *Antenna Theory: Analysis and Design*, 3rd ed., New York: Wiley-Interscience, 2005.

- [14] C. Argyropoulos, Y. Zhao, and Y. Hao, "A Radially-Dependent Dispersive Finite-Difference Time-Domain Method for the Evaluation of Electromagnetic Cloaks," *IEEE Transactions on Antennas and Propagation*, Vol. 57, No. 5, May 2009, pp. 1432–1441.
- [15] D. Xiao and H. T. Johnson, "Approximate Optical Cloaking in an Axisymmetric Silicon Photonic Crystal Structure," *Optical Letters*, Vol. 33, No. 8, 2008, pp. 860–862.
- [16] J. B. Pendry, D. Schurig, and D. R. Smith, "Controlling Electromagnetic Fields," *Science*, Vol. 312, No. 1780, 2006.

Chapter 7

Summary

In this chapter, we summarize the important factors in the FDTD method, FDTD software development, and simulation techniques. This summary is based on the authors' experiences during the past 20 years of FDTD method research. Due to different backgrounds, readers may have different opinions on some factors. However, no matter which way we choose, the final goal is to use the FDTD method efficiently to solve practical problems.

7.1 FDTD METHOD

The FDTD method is one kind of numerical technique that is used to solve Maxwell's equations in the time domain [1]. Due to its explicit recursive format, it is not necessary to solve Green's function and matrix equations in the FDTD method. Using the central difference formulation with the uniform mesh, the first-order difference formulation has the second-order accuracy [1]. Higher-order difference formulation [2] can reduce the numerical dispersion; however, it will dramatically increase the simulation time and memory requirement. The FDTD update equations do not contain the boundary information; therefore, the boundary to truncate the computational domain is required. For the open space problems, the desired absorbing boundary condition is PML [3, 4] today. Among the different versions of PML, the CPML [4] is the best choice today because its formulation is not related to the type of materials inside the computational domain. For the closed space problems, the selection of boundary is based on the property of problems.

To solve the practical problems, it is not sufficient for us to have only the FDTD update equations and boundary condition. We need to consider the following factors:

1. The size of the time step is chosen to meet the Courant condition [5] to achieve a stable solution.
2. Cell size must be small enough to satisfy the requirement of numerical dispersion level [1].

3. Spectrum of the excitation pulse should cover the frequency band of interest. The too narrow pulse will influence the convergence property and the time-domain signature may become meaningless due to the serious numerical dispersion.
4. The boundary of the dielectric objects requires special treatment regardless of the object shapes due to the singularity of the tangential component.
5. Plane wave excitation implementation is selected to be either scattering formulation or scattering/total formulation [1]. The scattering formulation is more accurate in the incident wave than the scattering/total formulation, especially for the large space problems; however, it is more time-consuming due to its reality is in the 3-D space.
6. Far-field calculation is realized by the near-to-far-field transformation. The downsampling in the space and time domain will significantly reduce the simulation time and memory requirement. The surface for near field calculation can be one surface and two surfaces [6]. In the one surface case, both the magnetic and electric fields are averaged to the selected surface. However, in the two surface case, the electric and magnetic fields are averaged to the two different surfaces. The far field using the two surfaces is usually more accurate than using the one surface.
7. The direct output parameters of the FDTD method are electric and magnetic fields, which have one-half cell shift in the space and one-half time step shift in the time domain. Ignoring this shift will cause the approximation error at higher frequencies.
8. Parallel processing techniques [7] are divided into the method based on OpenMP [8] or MPI [9]. OpenMP is designed to use the multiple cores inside one CPU or multiple CPUs that share the same host name. However, MPI allows us to use the distributed resources, but we need to develop the FDTD code using the MPI library.
9. Hardware acceleration techniques include multiple core processor, VALU [10], GPU [11], and FPGA [12]. The FDTD code developed using OpenMP is simple and can use all the compute cores inside the multiple core processor. VALU acceleration requires the developers be familiar with the SSE instruction set. GPU acceleration requires the developers to be familiar with the CUDA development kit and a separate GPU card. FPGA acceleration requires the developers to have some hardware and circuit knowledge.

Besides these factors, there are a lot of techniques in the FDTD method that are important for us to use the FDTD method to solve the practical problems, such as

conformal techniques, subgridding techniques, nonuniform mesh technique, ADI techniques [13], lumped element modeling, dispersive medium simulation [14], periodic structure simulation, nonlinear material simulation, body of revolution technique, metamaterial simulation techniques [15], and so on.

7.2 FDTD CODE DEVELOPMENT

The performance of the FDTD code will be different for the different programmers even if they use the same language. In any case, the key is for the developers to fully understand the data structure inside the memory. For example, the data of a 3-D array is continuous in the X-Y plane. The update loop in FORTRAN language should be written in the following way to reach better code performance:

```
Do k=0, Nz
  Do j=0, Ny
    Do i=0, Nx
      E_update;
    End do
  End do
End do
```

The update loop in C/C++ language should be written in the following way to reach the better code performance:

```
For (i=0; i<= Nx; i++) {
  For (j=0; j<= Ny, j++) {
    For (k=0; k<=Ny; k++) {
      E_update;
    }
  }
}
```

Considering the parallel processing and hardware acceleration, the update loop of electric fields should be placed in one loop. In this way, we can reuse the magnetic fields for the electric field update. For example, in C/C++ language,

```
For (i=0; i<= Nx; i++) {
  For (j=0; j<= Ny, j++) {
    For (k=0; k<=Ny; k++) {
      Ex_update;
      Ey_update;
      Ez_update;
    }
  }
}
```

In addition, considering the large amount of memory and memory price, we would rather achieve the best performance that may require increasing the memory usage. In any case, we should reduce the number of multiplications and divisions to the minimum inside the three layer loops.

Assemble language has a higher performance compared to FORTRAN and C/C++, but the code will be complex and not easy to debug. Both SIMD and SSE require developers to understand the hardware architecture.

7.3 FDTD SIMULATION TECHNIQUES

FDTD simulation technique is a bridge between the FDTD method and the engineering applications. The FDTD simulation technique will tell us how to use FDTD method to solve the practical problems. The main context of the simulation technique is how we can use the limited resource to solve one problem in a reasonable time frame. We summarize the main points in the simulation technique as follows:

1. *Mesh design.* In principle, the smaller the cell size, the more accurate the result. However, we cannot afford the price caused by the smaller cell size. For example, the simulation complexity is inversely proportional to h^4 , where h is the cell size. We have to find a balance between the simulation time, memory usage, and accuracy.

In practical applications, nonuniform mesh is helpful to balance the simulation time, memory usage, and accuracy. The proper ratio of adjacent cells is important to reach the best code performance. The subgridding is also a solution to reduce the simulation and memory usage, however the explicit differential technique is usually late time unstable, and the implicit technique has a low efficiency.

Actually, the mesh design is the trickiest part in the FDTD simulation. The important points in complex objects or interface of the different materials may significantly affect the simulation results. We define them as the key points during the mesh design. These points are detectable during the mesh generation by checking the geometrical shapes and material distribution. If they are too close to each other, we cannot use the too small cell size to discretize the computational domain limited to the computer resource, and we need to decide the most important key points manually and force the mesh passing these selected points.

2. *Boundary condition.* Application of the boundary condition is a relatively simple job in the FDTD simulation since there exist several typical boundary conditions in the FDTD method such as absorbing boundary,

PEC and PMC boundary, and periodic boundary. For the regular computational domains where the length, height, and width of the computational domain are equal or very close, we only need to set a proper white space between the objects and the domain boundary. The problem will become complex when the domain shape is irregular; for example, if the width and length of the domain are much larger than its height, the small white space in the height direction may result in the late time instability.

For the symmetric structures, we can use PEC or PMC to reduce the computational domain to half, quarter, or one-eighth of the original domain size if the excitation satisfies certain requirements.

3. *Excitation pulse.* The frequently used excitation pulse includes the pure Gaussian and differential Gaussian. The integral of the pure Gaussian pulse with respect to time is larger than zero, namely, the pure Gaussian excitation will move the pure charge from one pole to another one during the simulation, which will generate the DC component in the result time signature. However, if the port constructs a loop in some ways, the charge cannot be accumulated after the excitation disappears. Since the pure Gaussian contains more low frequency components, we usually select the pure Gaussian pulse as the excitation source when the frequency band of interest is very low. In contrast to the pure Gaussian pulse, we use the differential Gaussian pulse in most applications.
4. *Excitation.* In some cases, the excitation port has a clear definition and we know exactly how to excite it. But, in some cases, there exist several ways to excite an excitation port. A simple example is a patch antenna fed by a microstrip line. There are at least three ways to excite the excitation port: (1) we can use a match load to terminate the port first and then add a voltage excitation between the ground plane to the microstrip line, (2) we can terminate the microstrip line at the port at the absorbing boundary if the ground plane is infinity and then use a voltage or current to excite the port, and (3) we can treat it as a port that is excited by a TEM mode. The mode can be extracted by using different frequency-domain techniques. For the output ports we can follow a similar idea to extract the port voltage and current.
5. *Thin PEC sheet.* The conformal techniques for PEC and dielectric objects are different, so selecting the ideal material type can significantly improve

the FDTD performance. Dielectric conformal technique is realized through averaging the dielectric parameters. However, the PEC conformal technique requires modifying the field update procedure. For a thin PEC sheet, if the skin depth is important, we need to use at least two cells inside the thin PEC to describe the field variation. Otherwise, we can ignore its thickness and treat it as the infinitely thin PEC structure so that a coarse mesh can be employed.

7.4 HARDWARE PLATFORM

No matter how the FDTD code is developed, its performance must be realized on a better hardware platform. Here we present several guidelines on how to select the simulation hardware platforms. Generally speaking, a faster computer with the maximum memory is desired for the FDTD simulation. However, we frequently have to make a choice between the performance and cost.

The CPU speed must match the amount of memory. If too much memory is installed inside a computer, though we can simulate large problems, the simulation time may be beyond what we can afford. If we install a small amount of memory inside a powerful computer, the computer cannot be used to solve the large problems limited to the memory.

Multiple core processor and multiple CPUs have become a trend in hardware platform development [16, 17]. Multiple cores inside one computer can be fully used by OpenMP (multicore CPU), MPI (multiple CPUs), or a combination.

Multiple CPU computers today are developed based on the NUMA architecture, which allows us to improve the memory performance. To use the NUMA system, we usually need to modify the configuration file.

If we develop the FDTD code based on the Qt technique, the FDTD code will be cross platforms, namely, we do not need to modify the FDTD code when we change the operating system. The operating system can be Windows or Linux, and can reach similar performances of OpenMP and MPI.

The network speed in a cluster must match with the computer speed and number of computing nodes. The network is still the bottleneck today in the high performance cluster. For the parallel FDTD code, the latency of a network is one of the most important factors since the data exchanging happens at each time step. Besides the network performance, the memory bandwidth and hard disc performance are also very important factors. Namely, to achieve a better cluster performance, the performance of the network, CPU, memory, and hard drive must be well matched.

References

- [1] A. Taflové and S. Hagness, (eds.), *Computational Electromagnetics: The Finite-Difference Time-Domain Method*, 3rd ed., Norwood, MA: Artech House, 2005.
- [2] J. Young, "A Higher Order FDTD Method for EM Propagation in a Collisionless Plasma," *IEEE Transactions on Antennas and Propagation*, Vol. 44, No. 9, 1996, pp. 1283–1289.
- [3] J. Berenger, "A Perfectly Matched Layer Medium for the Absorption of Electromagnetic Waves," *Journal of Computational Physics*, Vol. 114, October 1994, pp. 185–200.
- [4] J. Roden and S. Gedney, "Convolution PML (CPML): An Efficient FDTD Implementation of the CFS-PML for Arbitrary Medium," *Microwave and Optical Technology Letters*, Vol. 27, No. 5, 2000, pp. 334–339.
- [5] R. Courant, K. Friedrichs, and H. Lewy, "Über die partiellen Differenzengleichungen der mathematischen Physik," *Mathematische Annalen*, Vol. 100, No. 1, 1928, pp. 32–74.
- [6] T. Martin, "An Improved Near-to-Far Zone Transformation for the Finite-Difference Time-Domain Method," *IEEE Transactions on Antennas and Propagation*, Vol. 46, No. 9, September 1998, pp. 1263–1271.
- [7] W. Yu, et al., *Parallel Finite-Difference Time-Domain Method*, Norwood, MA: Artech House, 2006.
- [8] <https://computing.llnl.tutorials/openMP>.
- [9] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, 2nd ed., Cambridge, MA: MIT Press, 1999.
- [10] W. Yu, et al., "New Development of Parallel Conformal FDTD Method in Computational Electromagnetic Engineering," *IEEE Antennas and Propagation Magazine*, June 2010.
- [11] J. P. Durbano, et al., "Implementation of Three-Dimensional FPGA-Based FDTD Solvers: An Architectural Overview," *Proceedings of the 11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, 2003.
- [12] A. Elsherbeni and V. Demir, *The Finite Difference Time Domain Method for Electromagnetics: With MATLAB Simulations*, Raleigh, NC: Scitech Publishing, 2008.
- [13] Z. F. Chen, Z. Zhang, and J. Toward, "The Development of a Three-Dimensional Unconditionally Stable Finite Difference Time-Domain Method," *IEEE Transactions on Microwave Theory and Techniques*, September 2009.
- [14] J. Young and R. Nelson, "A Summary and Systematic Analysis of FDTD Algorithms for Linearly Dispersive Media," *IEEE Antennas Propagation Magazine*, Vol. 43, No. 1, February 2001, pp. 61–77.
- [15] Y. Hao and R. Mittra, *FDTD Modeling of Metamaterials: Theory and Applications*, Norwood, MA: Artech House, 2009.
- [16] <http://www.intel.com>.
- [17] <http://www.amd.com>.

Appendix A

Antenna Power and Efficiency

Suppose an excitation source has an electric current density \vec{J} and a magnetic current density \vec{M} ; we calculate the power supplied to an antenna system when it is connected to an antenna system. If the electric and magnetic fields inside the source region V are \vec{E} and \vec{H} , respectively, the power supplied by the excitation source to the system can be expressed as [1, 2]:

$$P_{incident} = \frac{1}{2} \text{Re} \left(\iiint_V (\vec{E}^* \cdot \vec{J} + \vec{M}^* \cdot \vec{H}) dv \right) \quad (\text{A.1})$$

where $P_{incident}$ is the total incident power that the excitation source has when it is connected to an antenna. This total power includes the dissipation power on the source resistance and accepted power. The accepted power represents that the pure power passes the input port of antenna regardless of match or mismatch at the input port. For a lossless antenna, the accepted power should be equal to the radiation power. The matched radiation efficiency of an antenna, namely, accepted efficiency, is defined as:

$$E_{accepted} = \frac{P_{radiation}}{P_{accepted}} \quad (\text{A.2})$$

where $P_{radiation} = \frac{1}{2} \text{Re} \left(\oint_S (\vec{E} \times \vec{H}) \cdot d\vec{s} \right)$ is defined on the Huygens' surface, and the accepted power is expressed as:

$$P_{accepted} = P_{incident} - \frac{1}{2} \text{Re} \left(\iiint_V \sigma |E|^2 dv \right) \quad (\text{A.3})$$

where the second term in (A.3) is the dissipation power on the source internal resistance, as shown in Figure A.1.

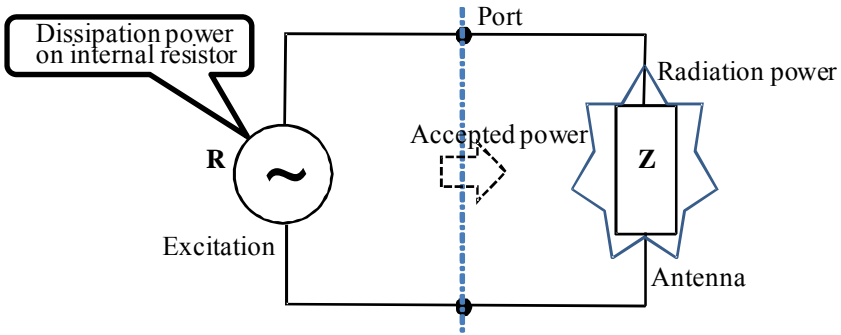


Figure A.1 Power distribution defined in an antenna system.

In the practical applications, we need to know how much power the excitation source delivers to the antenna originally, and how much of it is reflected back at the input port. Suppose the power delivered to the antenna port, called forward power, is $P_{forward} = P_{accepted} + P_{reflected}$, as shown in Figure A.2.

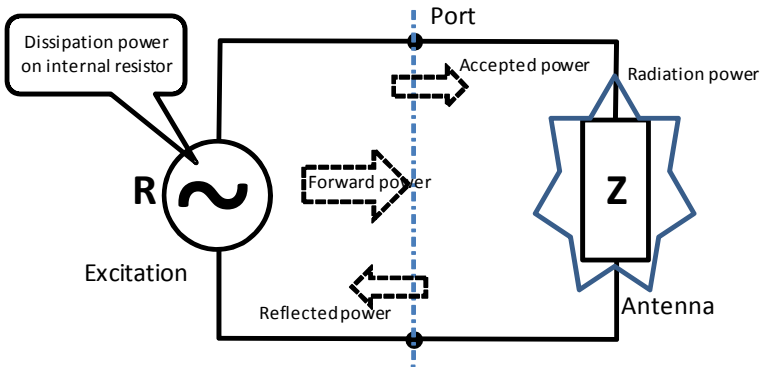


Figure A.2 Explanation of accepted power, forward power, and reflected power in an antenna system.

According to the definitions of S-parameters, for a lossless system we have [3]:

$$|S_{11}|^2 = \frac{P_{reflected}}{P_{forward}}$$

$$|S_{21}|^2 = \frac{P_{accepted}}{P_{forward}}$$

$$|S_{11}|^2 + |S_{21}|^2 = 1$$

The mismatched radiation efficiency of an antenna, called forward efficiency, is defined as:

$$\begin{aligned} E_{forward} &= \frac{P_{radiated}}{P_{forward}} = \frac{P_{radiated}}{P_{forward}} \frac{P_{accepted}}{P_{accepted}} \\ &= \frac{P_{radiated}}{P_{accepted}} \frac{P_{accepted}}{P_{forward}} = E_{accepted} (1 - |S_{11}|^2) \end{aligned} \quad (A.4)$$

Besides the forward and accepted efficiencies of an antenna, if the dissipation power on the source resistance is taken into account, the incident efficiency is expressed as:

$$E_{radiated} = \frac{P_{radiated}}{P_{incident}} \quad (A.5)$$

We can easily verify the relationship between the power in terms of circuit parameters and electromagnetic field.

$$P_{accepted} = \frac{1}{2} \text{Re}(V^* I)$$

where V and I are voltage and current measured at the input port of the antenna.

For a multiport excitation antenna system, the pure accepted power by antenna is the accepted power measured at the input port minus the transmitted power measured at the output ports. We first define an intermediate variable $P'_{accepted}$:

$$P'_{accepted} = P_{incident} - P_{dissipation} \quad (A.6)$$

which is the power accepted at the input port. The forward power propagating toward the input port is defined as:

$$P_{forward} = P_{reflected} - P'_{accepted} \quad (A.7)$$

Using the definition of S_{11} ,

$$|S_{11}|^2 = \frac{P_{reflected}}{P_{forward}} \quad (A.8)$$

we have:

$$P_{forward} = \frac{P'_{accepted}}{1 - |S_{11}|^2} \quad (A.9)$$

The pure accepted power of the antenna system, as shown in Figure A.3, is defined as:

$$P_{accepted} = P_{forward} - P_{reflected} - \sum_{i=2}^n P_{trans}^i \quad (A.10)$$

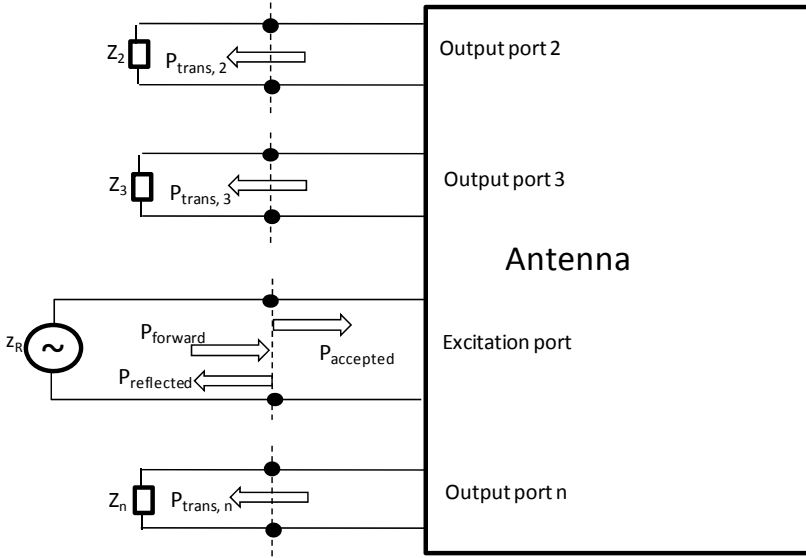


Figure A.3 Power relationship in a multiport antenna system.

Using (A.9) and the definition of S_{i1} ,

$$|S_{i1}|^2 = \frac{P_{trans}^i}{P_{forward}} \quad (A.11)$$

we have:

$$P_{accepted} = P'_{accepted} \frac{1 - |S_{11}|^2 - \sum_{i=2}^n |S_{i1}|^2}{1 - |S_{11}|^2} \quad (A.12)$$

The accepted efficiency is defined as:

$$E_{accepted} = \frac{P_{radiated}}{P_{accepted}} = \frac{P_{radiated}}{P'_{accepted}} \frac{1 - |S_{11}|^2}{1 - |S_{11}|^2 - \sum_{i=2}^n |S_{i1}|^2} \quad (A.13)$$

where $P'_{accepted}$ is known and can be calculated using (A.6). The value of $P'_{accepted}$ is zero when the S_{11} equals 1 and the radiation efficiency will be 0. The efficiency normalized to the forward power is defined as:

$$E_{forward} = \frac{P_{radiated}}{P_{forward}} = \frac{P_{radiated}}{P'_{accepted}} (1 - |S_{11}|^2) \quad (A.14)$$

The relationship between $E_{forward}$ and $E_{accepted}$ can be expressed as:

$$E_{forward} = E_{accepted} \left(1 - |S_{11}|^2 - \sum_{i=2}^n |S_{i1}|^2 \right) \quad (A.15)$$

References

- [1] R. Harrington, *Time-Harmonic Electromagnetic Fields*, New York: IEEE Press, 2001.
- [2] C. Balanis, *Advanced Engineering Electromagnetics*, New York: John Wiley & Sons, 1995.
- [3] D. M. Pozar, *Microwave Engineering*, 2nd ed., New York: John Wiley & Sons, 1998.

Appendix B

Active Reflection Coefficient

An antenna port and port parameters are described in Figure B.1. The port voltage and current are expressed as [1, 2]:

$$V = \sqrt{Z_0}(V^+ + V^-) \quad (\text{B.1a})$$

$$I = \sqrt{Z_0}(V^+ - V^-) \quad (\text{B.1b})$$

The variables in (B.1a) and (B.1b) are defined in Figure B.1.

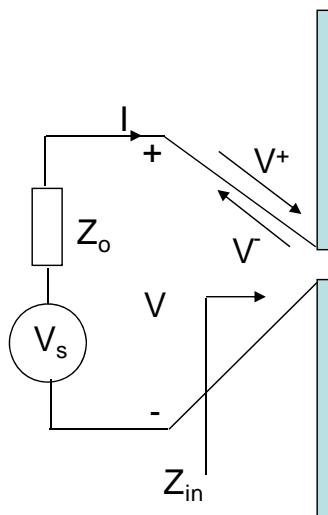


Figure B.1 Antenna feed port and the defined port parameters.

The input and reflected voltage at the input port are expressed as:

$$V^+ = \frac{V + Z_0 I}{2\sqrt{Z_0}} \quad (\text{B.2a})$$

$$V^- = \frac{V - Z_0 I}{2\sqrt{Z_0}} \quad (\text{B.2b})$$

The return loss of antenna can be expressed as:

$$S_{11} = \frac{V/\sqrt{Z_0} - I\sqrt{Z_0}}{V/\sqrt{Z_0} + I\sqrt{Z_0}} \quad (\text{B.3})$$

The voltage and current in (B.3) are total voltage and current measured at the input port, and Z_0 is the characteristic impedance of the antenna feed port.

For an antenna array problem (see Figure B.2), the incident and reflected voltages, V^+ and V^- , are defined at the terminals of each antenna element. The element feed ports are characterized by the $N \times N$ scattering matrix whose elements are defined as:

$$S_{nm} = \left. \frac{V_n^-}{V_m^+} \right|_{V_m^+ = 0, \text{ for } k \neq m} \quad (\text{B.4})$$

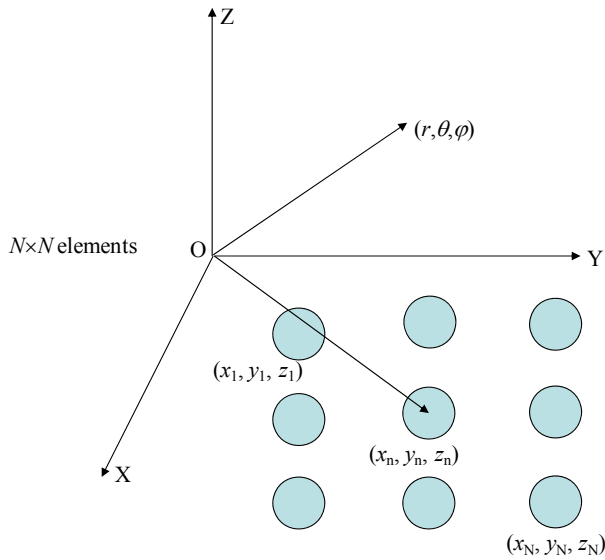


Figure B.2 $N \times N$ antenna array and the parameters used in the formulation derivation.

The total terminal voltage at the n th element is then:

$$V_n = V_n^+ + V_n^- = V_n^+ + \sum_{m=1}^N S_{nm} V_m^+ \quad (\text{B.5})$$

The field radiated by a single element can be expressed as:

$$E_n(r, \theta, \varphi) = V_n F_n(\theta, \varphi) e^{-jku_n} \frac{e^{-jkr}}{r} \quad (\text{B.6})$$

where V_n is the terminal voltage and $F_n(\theta, \varphi)$ represents the dominate polarization of the element pattern, and u_n is the element location parameter and can be expressed as:

$$u_n = x_n \sin \theta \cos \varphi + y_n \sin \theta \sin \varphi + z_n \cos \theta \quad (\text{B.7})$$

It is worthwhile to mention that the active element pattern is derived by the n th element while other elements are terminated by the matched loads. The total radiated field of the antenna array is given by:

$$\begin{aligned} E^a(r, \theta, \varphi) &= \frac{e^{-jkr}}{r} \sum_{n=1}^N V_n F_n(\theta, \varphi) e^{-jku_n} \\ &= \frac{e^{-jkr}}{r} \sum_{n=1}^N V_n^+ (1 + \Gamma_n) F_n(\theta, \varphi) e^{-jku_n} \end{aligned} \quad (\text{B.8})$$

In the full excitation phased array configuration, scanning to the angle (θ_0, φ_0) requires that the incident voltage at each element be phased as:

$$V_n = A_n e^{-jku_n(\theta_0, \varphi_0)} \quad (\text{B.9})$$

The resulting radiated field from the scanned array is then given by:

$$E^a(r, \theta, \varphi) = \frac{e^{-jkr}}{r} \sum_{n=1}^N \left(V_n = A_n e^{-jku_n(\theta_0, \varphi_0)} + \sum_{m=1}^N S_{nm} A_n e^{-jku_n(\theta_0, \varphi_0)} \right) F_n(\theta, \varphi) e^{-jku_n} \quad (\text{B.10})$$

The gain of the scanned array is then given by:

$$\begin{aligned}
 G^a(\theta, \varphi) &= \frac{4\pi r^2 |E^a(r, \theta, \varphi)|^2}{P_{inc}} \\
 &= \frac{4\pi}{\sum_{n=1}^N A_n^2} \left| \sum_{n=1}^N A_n^2 (1 + \Gamma_n) F_n(\theta, \varphi) e^{-jk(u_n(\theta, \varphi) - u_n(\theta_0, \varphi_0))} \right|^2
 \end{aligned} \tag{B.11}$$

where $P_{inc} = \sum_{n=1}^N A_n^2$. This gain is usually called the *realized* gain since the mismatch is considered. The active reflection coefficient seen at the element n of the fully excited array is given as:

$$\Gamma_n(\theta_0, \varphi_0) = \frac{V_n^-}{V_n^+} = \frac{\sum_{m=1}^N S_{nm} A_m e^{-jku_n(\theta_0, \varphi_0)}}{A_n e^{-jku_n(\theta_0, \varphi_0)}} \tag{B.12}$$

If equal power ($A_n = A_m$) is applied to each element terminal, we have then:

$$\Gamma_n(\theta_0, \varphi_0) = \frac{V_n^-}{V_n^+} = \sum_{m=1}^N S_{nm} A_m e^{-jk(u_m(\theta_0, \varphi_0) - u_n(\theta_0, \varphi_0))} \tag{B.13}$$

References

- [1] D. M. Pozar, "A Relation Between the Active Input Impedance and the Active Element Pattern of a Phased Array," *IEEE Transactions on Antennas and Propagation*, Vol. 51, No. 9, September 2003, pp. 2486–2489.
- [2] D. M. Pozar, "The Active Element Pattern," *IEEE Transactions on Antennas and Propagation*, Vol. 42, No. 8, August 1994, pp. 1176–1178.

Appendix C

Total Active Reflection Coefficient

An n -port antenna array is an $(N+2)$ -port network if considering the radiation as two ports for two polarizations. A complete S-parameter matrix should be an $(N+2) \times (N+2)$ matrix. However, the two ports corresponding to two polarizations cannot be handled as a regular port. According to the literature [1], we define a total active reflection coefficient for the multiport antenna array.

For a desired port excitation, the total active reflection coefficient (TARC) of the antenna is defined as the square root of the available power generated by all excitation minus the radiated power, divided by the available power as follows [1, 2]:

$$\Gamma_a^t = \sqrt{\frac{P_{available} - P_{radiated}}{P_{available}}} \quad (C.1)$$

The incident power, radiation power, and dissipation power are known, and the accepted power is equal to the incident power subtracting the dissipation power. According to the definition above, the available power is the forward power, but the forward is unknown in a multiport antenna array. For a lossless antenna, TARC can be calculated by using S-parameter matrix. For a given excitation $[a]$, the Γ_a^t , as a function of frequency, is:

$$\Gamma_a^t = \sqrt{\frac{\sum_{i=1}^N |b_i|^2}{\sum_{i=1}^N |a_i|^2}} \quad (C.2)$$

where

$$S_p = \begin{bmatrix} S_{11} & \dots & S_{1N} \\ \vdots & \ddots & \vdots \\ S_{N1} & \dots & S_{NN} \end{bmatrix} \quad (\text{C.3})$$

and

$$[b] = [S_p][a] \quad (\text{C.4})$$

We do not need to define TARC as a complex number because the phase reference plane does not have any physical meaning for a multiport antenna. TARC is a real number between zero and one. When the TARC equals zero, all the delivered power is radiated and when it equals one, all the power is either reflected back or goes to the other ports.

References

- [1] M. Manteghi and Y. Rahmat-Samii, "Broadband Characterization of the Total Active Reflection Coefficient of Multiport Antennas," *IEEE Antennas and Propagation Symposium*, 2003, pp. 20–23.
- [2] D. M. Pozar, *Microwave Engineering*, 2nd ed., New York: John Wiley & Sons, 1998.

Appendix D

MEG and ECC

The performance of a mobile phone is determined by both the antenna quality and the environment in which the mobile device is embedded. Both the mean effective gain (MEG) and envelope correlation coefficient (ECC) depend on the distribution of the incoming power and the orientation of the antenna relative to this environment [1, 2].

D.1 PROPAGATION ENVIRONMENTS

To calculate the ECC and the MEG, we need to take into consideration the incident radio wave. The propagation effects are defined as the power spectrum of the polarized incoming waves $P_\theta(\Omega)$ and $P_\varphi(\Omega)$, which can be expressed as the statistical distribution functions [2, 3]:

$$P_\theta(\Omega) = P_\theta(\theta, \varphi) = P_\theta(\theta)P_\varphi(\varphi) \quad (\text{D.1})$$

$$P_\varphi(\Omega) = P_\varphi(\theta, \varphi) = P_\varphi(\theta)P_\theta(\varphi) \quad (\text{D.2})$$

where, $P_\theta(\theta)$ and $P_\varphi(\varphi)$ are the angular density functions in the elevation plane, and $P_\theta(\theta)$ and $P_\varphi(\varphi)$ are the angular density functions in the azimuth plane. The commonly used angular functions include uniform, Gaussian, Laplacian, and elliptical patterns. To determine the coefficients, the angular functions are satisfied with the normalization:

$$\int_0^{2\pi} \int_0^\pi P_\theta(\theta, \varphi) \sin \theta d\theta d\varphi = 1 \quad (\text{D.3})$$

$$\int_0^{2\pi} \int_0^\pi P_\varphi(\theta, \varphi) \sin \theta d\theta d\varphi = 1 \quad (\text{D.4})$$

D.2 ECC

The independent fading signals incoming to the receiver of the system can be computed using the general expression of ρ_e when using the 3-D far-field pattern as the following equation:

$$\rho_e = \frac{N}{D_1 D_2} \quad (\text{D.5})$$

where,

$$N = \left| \int_0^{2\pi} \int_0^\pi XPRE_{\theta 1}(\theta, \varphi) E_{\theta 2}^*(\theta, \varphi) (P_\theta(\theta, \varphi) + P_\varphi(\theta, \varphi)) \sin \theta d\theta d\varphi \right|^2$$

$$D_1 = \left| \int_0^{2\pi} \int_0^\pi XPRE_{\theta 1}(\theta, \varphi) (E_{\theta 1}^*(\theta, \varphi) P_\theta(\theta, \varphi) + E_{\varphi 1}^*(\theta, \varphi) P_\varphi(\theta, \varphi)) \sin \theta d\theta d\varphi \right|$$

$$D_2 = \left| \int_0^{2\pi} \int_0^\pi XPRE_{\theta 2}(\theta, \varphi) (E_{\theta 2}^*(\theta, \varphi) P_\theta(\theta, \varphi) + E_{\varphi 2}^*(\theta, \varphi) P_\varphi(\theta, \varphi)) \sin \theta d\theta d\varphi \right|$$

D.3 MEG

MEG is a statistical measurement of the antenna gain in a mobile environment [2]. In a particular environment, MEG is defined as the power received by an antenna including the contribution of the radiation power pattern, the antenna total efficiency, and the propagation effects, as in the following equation:

$$MEG = \left| \int_0^{2\pi} \int_0^\pi \left(\frac{XPR}{1 + XPR} G_\theta(\theta, \varphi) P_\theta(\theta, \varphi) + \frac{XPR}{1 + XPR} G_\varphi(\theta, \varphi) P_\varphi(\theta, \varphi) \right) \sin \theta d\theta d\varphi \right|$$

#

where $G_\theta(\theta, \varphi)$ and $G_\varphi(\theta, \varphi)$ are the antenna power gain patterns in the θ - and φ -directions, respectively. Usually, MEG is normalized in the following way [1]:

$$1 = \left| \int_0^{2\pi} \int_0^\pi (G_\theta(\theta, \varphi) + G_\varphi(\theta, \varphi)) \sin \theta d\theta d\varphi \right|$$

For the uniform model, the environment functions are defined as follows:

$$P_\theta(\theta) = \frac{1}{4\pi}, \quad P_\varphi(\theta) = \frac{1}{4\pi}$$

$$P_{\theta}(\varphi) = 1, \quad P_{\varphi}(\theta) = 1$$

For the Gaussian uniform model, the environment functions are defined as follows:

$$P_{\theta}(\theta) = A_{\theta} e^{-\frac{(\theta - (\pi/2 - m_v))^2}{2\sigma_v^2}}, \quad P_{\varphi}(\theta) = A_{\varphi} e^{-\frac{(\theta - (\pi/2 - m_H))^2}{2\sigma_H^2}}$$

$$P_{\theta}(\varphi) = 1, \quad P_{\varphi}(\varphi) = 1$$

For the Laplacian uniform model, the environment functions are defined as follows:

$$P_{\theta}(\theta) = A_{\theta} e^{-\frac{\sqrt{2}(\theta - (\pi/2 - m_v))^2}{\sigma_v}}, \quad P_{\varphi}(\theta) = A_{\varphi} e^{-\frac{\sqrt{2}(\theta - (\pi/2 - m_H))^2}{\sigma_H}}$$

$$P_{\theta}(\varphi) = 1, \quad P_{\varphi}(\varphi) = 1$$

For the elliptical model, the environment functions are defined as follows:

$$P_{\theta}(\theta) = \sqrt{A_{\theta}} \frac{S_{\theta\theta}^2}{S_{\theta\theta}^2 + (\sin \theta)^2}, \quad P_{\varphi}(\theta) = \sqrt{A_{\varphi}} \frac{S_{\varphi\theta}^2}{S_{\varphi\theta}^2 + (\sin \theta)^2}$$

High directivity (1/8 of the sphere):

$$P_{\theta}(\varphi) = \sqrt{A_{\theta}} a_{\theta 0}, \quad P_{\varphi}(\varphi) = \sqrt{A_{\varphi}} (a_{\theta 0} - b_{\varphi} |\varphi|)$$

Low directivity (1/2 of the sphere):

$$P_{\theta}(\varphi) = \sqrt{A_{\theta}} \frac{S_{\theta\varphi}^2}{S_{\theta\varphi}^2 + (\sin \varphi)^2}, \quad P_{\varphi}(\varphi) = \sqrt{A_{\varphi}} \frac{S_{\varphi\varphi}^2}{S_{\varphi\varphi}^2 + (\sin \varphi)^2}$$

References

- [1] A. Diallo, et al., "Diversity Performance of Multiantenna Systems for UMTS Cellular Phones in Different Propagation Environments," *International Journal of Antennas and Propagation*, Vol. 2008, Article ID: 836050.
- [2] T. Taga, "Analysis for Mean Effective Gain of Mobile Antennas in Land Mobile Radio Environments," *IEEE Transactions on Vehicular Technology*, Vol. 39, No. 2, 1990, pp. 117–131.
- [3] G. F. Pederson and J. B. Andersen, "Handset Antennas for Mobile Communication Integration, Diversity and Performance," *Review of Radio Science*, Chapter 5, London, U.K.: Oxford University Press, 1999, pp. 119–137.

Appendix E

Lossy Dielectric Simulation Technique

In practical applications, if we do not mind the variation of dielectric loss with the frequency, we can specify the conductivity value in the FDTD method such as in most antenna problems. However, if the variation of dielectric loss cannot be ignored in the applications, we need to simulate the loss tangent in the FDTD method. Generally speaking, the input parameters of a dielectric material are permittivity and loss tangent, and both are constant in the selected frequency range. It is a well-known fact that the dispersion relationship in the frequency domain must be converted into the time domain in order to simulate it in the FDTD method [1, 2]. In this appendix, we introduce a simple but efficient technique to handle the loss tangent in the FDTD method [3].

We can use the approaches in the literature [1, 4] to extract the poles of the dispersive material using the Cauchy technique through the singular value decomposition technique. In this method, you need to provide the measurement dielectric parameters at more than two frequencies and go through a complex procedure.

A complex permittivity can be expressed as [3]:

$$\varepsilon = \varepsilon_0 (\varepsilon_r' - j\varepsilon_r'') \quad (\text{E.1})$$

where ε_r' and ε_r'' are real constants. While this specification is not causal, it is a good approximation to the behavior of many real materials. A causal expression can be expressed as:

$$\varepsilon = \varepsilon_0 \left(\varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + j\omega\tau_0} - \frac{j\sigma'}{\omega\varepsilon_0} \right) \quad (\text{E.2})$$

where ε_∞ is the infinite frequency permittivity, ε_s is the static permittivity, and τ_0 is the relaxation time.

For a given permittivity and loss tangent, we can calculate the values ε_r' and $\varepsilon_r'' = \varepsilon_r' \cdot \delta$ [5], and then we get a set of nonlinear equations:

$$K_1 \varepsilon_r' = \varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + (2\pi f_{\min} \tau_0)^2} \quad (\text{E.3a})$$

$$K_2 \varepsilon_r'' = \frac{(\varepsilon_s - \varepsilon_\infty) 2\pi f_{\min} \tau_0}{1 + (2\pi f_{\min} \tau_0)^2} + \frac{\sigma'}{2\pi f_{\min} \tau_0} \quad (\text{E.3b})$$

$$K_3 \varepsilon_r' = \varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + (2\pi f_{\min} \tau_0)^2} \quad (\text{E.3c})$$

$$K_4 \varepsilon_r'' = \frac{(\varepsilon_s - \varepsilon_\infty) 2\pi f_{\min} \tau_0}{1 + (2\pi f_{\min} \tau_0)^2} + \frac{\sigma'}{2\pi f_{\min} \tau_0} \quad (\text{E.3d})$$

For $K_1 = K_2 = K_3 = K_4 = 1$, the equations above will be ill posed and have no solution. However, we can try the K 's values that are very close to 1 to get a group of meaningful parameters, namely, ε_s , ε_∞ , τ_0 , and σ' . A subroutine `hybrd1(fcn, n, x, fvec, tol, info, wa, lwa)` of MINPACK [6] is used to calculate the K_1, K_2, K_3 , and K_4 . The function pointer `fcn` points to the equations above. The n is the number of equations, and it is equal to 4 here. The unknown x is an array whose elements are ε_s , ε_∞ , τ_0 , and σ' . The `fvec` is the value of the equations when the array x is used as the input. The `tol` is the tolerance. The `wa` is a work array of length `lwa`. `lwa` is a positive integer input variable not less than $(n*(3*n+13))/2$.

References

- [1] A. Taflov and S. Hagness, (eds.), *Computational Electromagnetics: The Finite-Difference Time-Domain Method*, 3rd ed., Norwood, MA: Artech House, 2005.
- [2] W. Yu, et al., *Parallel Finite-Difference Time-Domain Method*, Norwood, MA: Artech House, 2006.
- [3] R. Luebbers, "Lossy Dielectrics in FDTD," *IEEE Transactions on Antennas and Propagation*, Vol. 41, No. 11, November 1993, pp. 1685–1687.
- [4] K. Kottapalli, T. Sarkar, and Y. Hua, "Accurate Computation of Wide-Band Response of Electromagnetic Systems Utilizing Narrow-Band Information," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 39, No. 4, April 1991, pp. 682–688.
- [5] C. Balanis, *Advanced Engineering Electromagnetics*, New York: John Wiley & Sons, 1995.
- [6] <http://www.netlib.org/minpack>.

Appendix F

Circular Polarization Decomposition

A circular polarization wave can be split into the summation of a left and a right polarization wave. The co- and cross-polarization levels are important parameters that need to be estimated when designing circularly polarized antennas. For the more general case of an elliptical polarization, the axial ratio (AR) and the circular polarization index (CPI) are often employed as indirect measures of cross polarization. However, sometimes the co- and cross-polarization levels are needed for circularly polarized antennas, and this information cannot be readily obtained from the existing EM simulation tools. In this appendix, we describe an approach for computing the co- and cross-polarization levels of circularly polarized antennas.

In general, the fields radiated by an antenna are elliptically polarized, which can be resolved into a pair of orthogonal, circular polarized components at each point on the radiation sphere. The two components are E_L and E_R , whose senses are left- and right-handed rotations, respectively. Obviously, the dominant one corresponds to the copolarization, or the primary polarization that the antenna is designed to transmit (or receive). The other one is the cross-polarization, which is orthogonal to the copolarization and should have a low intensity as compared to the primary polarization. Thus, the instantaneous field of an elliptically polarized wave, traveling in the (θ, φ) direction, can be written as [1, 2]:

$$\vec{E}(\theta, \varphi, t) = \vec{E}_L(\theta, \varphi, t) + \vec{E}_R(\theta, \varphi, t) \quad (\text{F.1})$$

This instantaneous field can also be written as:

$$\vec{E}(\theta, \varphi, t) = \hat{a}_\theta E_\theta(\theta, \varphi, t) + \hat{a}_\varphi E_\varphi(\theta, \varphi, t) \quad (\text{F.2})$$

where the $E_\theta(\theta, \varphi, t)$ and $E_\varphi(\theta, \varphi, t)$ are two orthogonal components of the far field in the θ - and φ -directions. An arbitrary circularly polarized field can be represented as a summation of two orthogonal linearly polarized components in the θ - and φ -directions. Thus, we can write

$$E_\theta = E_{L\theta} + E_{R\theta} \quad (\text{F.3a})$$

$$E_\varphi = E_{R\varphi} + E_{L\varphi} \quad (\text{F.3b})$$

where $E_{R\varphi}$, $E_{L\varphi}$, $E_{R\theta}$, and $E_{L\theta}$ are complex and represent the components of $E_L(\theta, \varphi)$ and $E_R(\theta, \varphi)$ in the θ - and φ -directions, respectively. Since $E_{R\varphi} = E_{R\theta}e^{-j\pi/2}$ and $E_{L\varphi} = E_{L\theta}e^{j\pi/2}$, we can solve (F-3) to get:

$$E_{L\theta} = \frac{1}{\sqrt{2}} \left(E_\theta + E_\varphi e^{-j\frac{\pi}{2}} \right) \quad (\text{F.4a})$$

$$E_{R\theta} = \frac{1}{\sqrt{2}} \left(E_\theta + E_\varphi e^{j\frac{\pi}{2}} \right) \quad (\text{F.4b})$$

Next, we can write:

$$\text{Re}[E_{L\theta}] = \frac{1}{\sqrt{2}} [E_{\theta m} \cos(\omega t - \beta r + \psi_\theta) + E_{\varphi m} \sin(\omega t - \beta r + \psi_\varphi)] \quad (\text{F.5a})$$

$$\text{Re}[E_{R\theta}] = \frac{1}{\sqrt{2}} [E_{\theta m} \cos(\omega t - \beta r + \psi_\theta) - E_{\varphi m} \sin(\omega t - \beta r + \psi_\varphi)] \quad (\text{F.5b})$$

where $E_{\theta m}$ and $E_{\varphi m}$ are the amplitude of E_θ and E_φ ; ψ_θ and ψ_φ are the phases of E_θ and E_φ , respectively. We now expand (F.5) and define $E_L = E_{L\theta} = |\text{Re}(E_{L\theta})|_{\max}$ and $E_R = E_{R\theta} = |\text{Re}(E_{R\theta})|_{\max}$. For the general case, we have:

$$|E_L| = \frac{1}{\sqrt{2}} \sqrt{E_{\theta m}^2 + E_{\varphi m}^2 - 2E_{\theta m}E_{\varphi m} \sin(\psi_\theta - \psi_\varphi)} \quad (\text{F.6a})$$

$$|E_R| = \frac{1}{\sqrt{2}} \sqrt{E_{\theta m}^2 + E_{\varphi m}^2 + 2E_{\theta m}E_{\varphi m} \sin(\psi_\theta - \psi_\varphi)} \quad (\text{F.6b})$$

If $|E_L| > |E_R|$, it is a left-handed elliptically polarized wave, with co-polarization corresponding to $|E_L|$ and crosspolarization corresponding to $|E_R|$. On the other hand, if $|E_R| > |E_L|$, the reverse is true. The phases of $\bar{E}_L(\theta, \varphi; t)$ and $\bar{E}_R(\theta, \varphi; t)$ have no effects on the basic shape of the ellipse, but only on the tilt of the major axis of the ellipse relative to the coordinate axis.

References

- [1] D. Kraus, *Antennas*, New York: McGraw-Hill, 1988.
- [2] Z. Li, et al., "Analysis of Large Circularly Polarized Antenna Array by Using a Parallelized FDTD Code Running on a High Performance Cluster," *IEEE Antennas and Propagation Symposium*, 2009.

Appendix G

Vector Fitting Technique

In most frequency-domain methods, we need to solve the problem one frequency at each time, namely, the solutions are only at the discrete frequencies. It is necessary for us to have a robust interpolation technique to generate a smooth curve from the discrete points. In this appendix, our objective is to calculate the unknown poles a_j , residues c_j , and the constant term d in the expression:

$$h(s) \approx \sum_{j=1}^N \frac{C_j}{S - a_j} + d \quad (\text{G.1})$$

where $h(s)$ is a vector or a scalar. To explain the method, we now assume that $h(s)$ is a scalar. The summation limit N is the order of the approximation, assumed to be known.

We recall that the difficulty in fitting $h(s)$ is due to the (unknown) poles a_j , which appear in the denominator, thus causing (G.1) to become a nonlinear problem in the unknowns. However, if the poles had been known, then (G.1) would have been linear in the unknowns c_j , d , which could then have easily been calculated by solving a linear, least-squares problem. Vector fitting solves (G.1) in two steps. Each step amounts to solving a linear system of the same form as (G.1), but with known poles. The vector fitting technique includes two steps.

Step 1: Pole identification.

Instead of fitting $h(s)$ directly, $h(s)$ is multiplied with an unknown rational function $\theta(s)$ of order N :

$$\theta(s) \approx \sum_{j=1}^N \frac{\hat{C}_j}{S - \bar{a}_j} + 1 \quad (\text{G.2})$$

The poles of $\theta(s)$ are assigned a set of initial values a_j , which span in an almost arbitrary way the domain of interest. We then postulate that the function $\theta(s) h(s)$ can be fitted with the same set of poles a_j as $\theta(s)$:

$$\theta(s)h(s) \approx \sum_{j=1}^N \frac{\tilde{C}_j}{s - \bar{a}_j} + \tilde{d} \quad (\text{G.3})$$

Combining (G.2) and (G.3) gives:

$$\left(\sum_{j=1}^N \frac{\hat{C}_j}{s - \bar{a}_j} + 1 \right) h(s) \approx \sum_{j=1}^N \frac{\tilde{C}_j}{s - \bar{a}_j} + \tilde{d} \quad (\text{G.4})$$

Equation (G.4) is linear in the unknowns \hat{C}_j , \tilde{C}_j , and \tilde{d} , which are calculated by solving a linear least-squares equation of the form $Ax = b$. From (G.4) we get:

$$h(s) = \frac{\sum_{j=1}^N \frac{\tilde{C}_j}{s - \bar{a}_j} + \tilde{d}}{\sum_{j=1}^N \frac{\hat{C}_j}{s - \bar{a}_j} + 1} = \frac{\prod_{j=1}^N \frac{(s - \tilde{z}_j)}{(s - \bar{a}_j)}}{\prod_{j=1}^N \frac{(s - \hat{z}_j)}{(s - \bar{a}_j)}} = \frac{\prod_{j=1}^N (s - \tilde{z}_j)}{\prod_{j=1}^N (s - \hat{z}_j)} \quad (\text{G.5})$$

It is seen from (G.5) that the initial poles a_j cancel out, and that the zeros \hat{z}_j of $\theta(s)$ become the poles of $h(s)$! The zeros of $\theta(s)$ are calculated by input/output interchange from its state equation obtained from (G.2).

Step 2: Residue identification.

With the poles of $h(s)$ known from Step 1, (G.1) is solved with respect to the unknown residues C_j and constant term d by solving a linear least-squares problem of the type $Ax = b$. Steps 1 and 2 may have to be repeated several times with the new poles as starting poles, in order for the method to converge. As convergence is achieved, $\theta(s)$ becomes unity (i.e., all C_j become zero). With a good set of starting poles, we usually need fewer than five iterations for our applications. Thus, vector fitting solves (G.1) as follows.

1. Choose a set of starting poles $\bar{a}_1 \dots \bar{a}_N$.

2. Solve (G.4) with respect to \hat{C}_j , \tilde{C}_j , and \tilde{d} .
 3. Calculate the poles \bar{a}_1 \bar{a}_N of $h(s)$ as the zeros of $\theta(s)$. $\theta(s)$ is given by (G.2).
 4. Solve (G.1) with respect to C_j and d .
 5. If necessary, repeat procedures 1–4 with the new poles as starting poles.
- Details regarding the implementation are given in [1–6].

References

- [1] A. Semlyen and B. Gustavsen, "Vector Fitting by Pole Relocation for the State Equation Approximation of Nonrational Transfer Matrices," *Circuits System Signal Process*, Vol. 19, No. 6, 2000, pp. 549–566.
- [2] B. Gustavsen and A. Semlyen, "Combined Phase and Modal Domain Calculation of Transmission Line Transients Based on Vector Fitting," *IEEE Transactions on Power Delivery*, Vol. 3, No. 2, April 1998, pp. 596–604.
- [3] B. Gustavsen and A. Semlyen, "Simulation of Transmission Line Transients Using Vector Fitting and Modal Decomposition," *IEEE Transactions on Power Delivery*, Vol. 13, No. 2, April 1998, pp. 605–614.
- [4] B. Gustavsen and A. Semlyen, "Application of Vector Fitting to State Equation Representation of Transformers for Simulation of Electromagnetic Transients," *IEEE Transactions on Power Delivery*, Vol. 13, No. 3, July 1998, pp. 834–842.
- [5] B. Gustavsen and A. Semlyen, "Calculation of Transmission Line Transients Using Polar Decomposition," *IEEE Transactions on Power Delivery*, Vol. 13, No. 3, July 1998, pp. 855–862.
- [6] B. Gustavsen and A. Semlyen, "Rational Approximation of Frequency-Domain Responses by Vector Fitting," *IEEE Transactions on Power Delivery*, Vol. 14, No. 3, July 1999, pp. 1052–1061.

Appendix H

Partially Symmetric Problem Simulation

If both the problem geometry and excitation source are symmetric, we can use the proper PEC or PMC boundary to reduce the problem size to half, one-quarter, or one-eighth of the original problem. The selection of the PEC or PMC boundary at each side is determined by the polarization of the excitation source [1, 2]. The basic idea is described in Figure H.1.

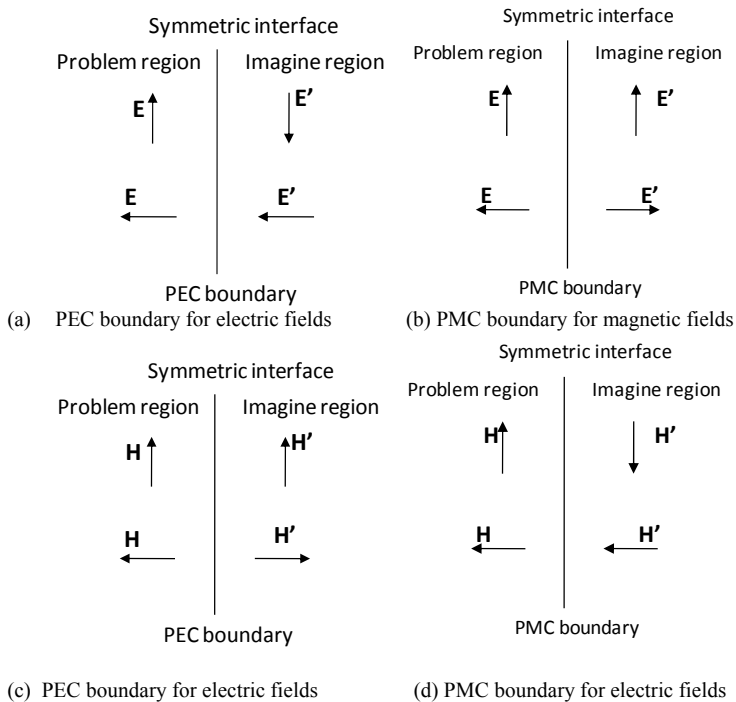


Figure H.1 Relationship between the boundary condition and excitation: (a) electric field for PEC, (b) electric field for PMC, (c) magnetic field for PEC, and (d) magnetic field for PMC.

Since the electric and magnetic fields are half cell offset in the FDTD mesh definition, the PEC and PMC boundaries are also half cell offset. However, the position of the PMC boundary varies with the cell size that is not desired for the practical problems. We can apply the PMC boundary on the edge of FDTD cell using the following idea:

In Figure H.2, the electric fields on the the PMC boundary are calculated by using the magnetic fields inside the simulation area and the corresponding images. If the problem structure is symmetric but the excitation source is not symmetric, for example, one case is shown in Figure H.3.

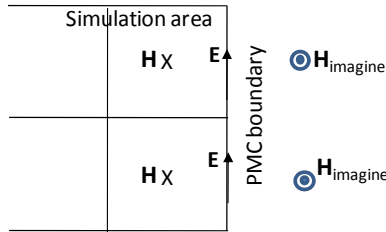


Figure H.2 Idea for PMC boundary implementation.

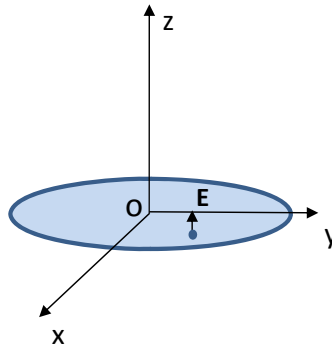


Figure H.3 Symmetric structure with asymmetric excitation.

If we apply the PEC or PMC boundary to reduce the problem size, the excitation source is also imaged, and hence, there are four excitations in the original problem domain, as shown in Figure H.4. In order to obtain the solution of the original problem, we need to simulate the problem (only one-quarter of the original problem) four times in which the PEC and PMC boundaries should be changed properly. For example, the boundary should be PEC and PMC in the first simulation, PEC and PEC in the second simulation, PMC and PMC in the third simulation and PMC and PEC in the fourth simulation. The near-field result, such as electric and magnetic fields, voltage, and current, is equal to the summation divided by a factor of 4. The far-zone field can be calculated by the vector

summation of each component. To calculate the directivity and gain, we need to calculate the radiation power of each case.

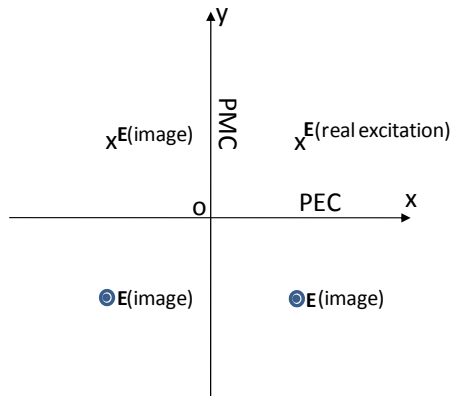


Figure H.4 Boundary and imaged excitation sources.

References

- [1] W. Yu, et al., *Parallel Finite-Difference Time-Domain Method*, Norwood, MA: Artech House, 2006.
- [2] W. Yu, et al., "New Development of Parallel Conformal FDTD Method in Computational Electromagnetic Engineering," *IEEE Antennas and Propagation Magazine*, Vol. 53, No. 3, 2011.

Appendix I

Time-Domain Reflectometry (TDR)

Signal integrity is a growing priority as digital system designers pursue ever-higher clock and data rates in computer, communications, video, and network systems. At today's high operating frequencies, anything that affects a signal rise time, pulse width, timing, jitter, or noise content can impact reliability at the system level. To ensure signal integrity, it is necessary to understand and control impedance in the transmission environment through which the signals travel. Mismatches and variations can cause reflections that decrease signal quality as a whole.

Time-domain reflectometry (TDR) [1, 2] indicates the reflections that result from a signal travelling through a transmission environment of some kind: a circuit board trace, a cable, a connector, and so on. The excitation source sends a narrow Gaussian pulse through the medium and compares the reflections from the unknown transmission environment to those produced by a standard impedance.

The TDR display is the voltage waveform that returns when a fast step signal is propagated down a transmission line. The resulting waveform is the combination of the incident step and reflections generated when the step encounters impedance variations.

The mathematical foundation of TDR measurements is elementary but important. Fundamentally, TDR value is based on a series of impedance ratios and are described in terms of a reflection coefficient, ρ . The coefficient ρ is the ratio of the reflected pulse amplitude to the incident pulse amplitude:

$$\rho = \frac{V_{reflected}}{V_{incident}} \quad (I.1)$$

where $V_{incident}$ is the integral of the incident pulse with respect to time and $V_{reflected}$ is the integral of the reflected pulse with respect to time. The excitation pulse in FDTD simulation must be the narrow Gaussian that ensures there is no overlapping between the incident pulse and reflected pulse.

For a fixed termination Z_L , ρ can also be expressed in terms of the transmission line characteristic impedance, Z_0 and the load impedance Z_L .

$$\rho = \frac{V_{reflected}}{V_{incident}} = \frac{(Z_L - Z_0)}{(Z_L + Z_0)} \quad (I.2)$$

Now that we have the formulas, if we plug in numbers representing a matched load, a short circuit, and an open load, we can see that ρ has a range of values from +1 to -1, with 0 representing a matched load.

When Z_L is equal to Z_0 , the load is matched. $V_{reflected}$, the reflected wave, is equal to 0 and ρ is 0. There are no reflections:

$$\rho = \frac{V_{reflected}}{V_{incident}} = \frac{0}{V} = 0 \quad (I.3)$$

A Z_L reading of zero (0) implies a short circuit. The reflected wave is equal to the incident wave, but opposite in polarity. As seen below, the reflected wave negates part of the incident wave. The ρ value is -1.

$$\rho = \frac{V_{reflected}}{V_{incident}} = \frac{-V}{V} = -1 \quad (I.4)$$

When Z_L is infinite, an open circuit is implied. The reflected wave is equal to the incident wave and of the same polarity. As seen below, the reflected wave reinforces part of the incident wave. The ρ value is +1.

$$\rho = \frac{V_{reflected}}{V_{incident}} = \frac{V}{V} = 1 \quad (I.5)$$

From (I.2), the time-domain impedance is expressed as:

$$Z_L = Z_0 \cdot \frac{1 + \rho}{1 - \rho} \quad (I.6)$$

where Z_0 and ρ are characteristic impedance and TDR. The capacitive and inductive discontinuities can be recognized in the TDR plot, as shown in Figure I.1.

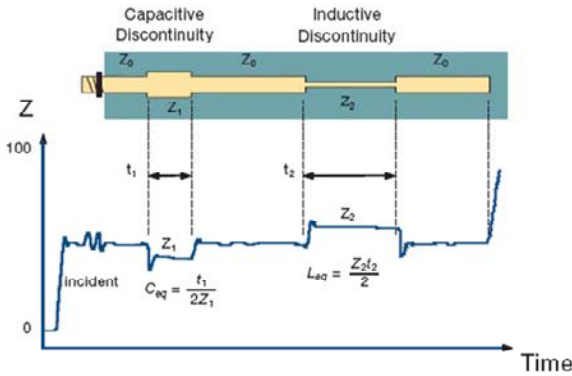


Figure I.1 Capacitive discontinuity and inductive discontinuity in the TDR plot.

References

- [1] Tektronix Application note, http://www2.tek.com/cmsreplive/tirep/3846/55W_14601_2_2008.09.04.12.32.15_3846_EN.pdf.
- [2] Tektronix Application note, http://www2.tek.com/cmsreplive/tirep/4873/85W_19888_0_LowRes_2010.04.15.17.32.48_4873_EN.pdf.

Appendix J

S-Parameter Extraction

The S-parameter is one of most important parameters in antennas, microwave circuits, EMC/EMI, and signal integrity. In this appendix we introduce the extraction techniques of the S-parameters for different types of applications.

J.1 MATCHED S-PARAMETER

For a multiple-port network system, we need to terminate all ports using the match loads and excite one port each time to calculate the S-parameter matrix. To calculate the S-parameters at the match status, we need to measure at least four voltages at the excitation port and one voltage at each output port. Without knowing the port impedance, we can calculate the S-parameters using the following procedure [1]:

1. Measure four voltages at the excitation port along the feed structure V_1, V_2, V_3 , and V_4 .
2. Assume the measured total voltage at the excitation port at position r includes two waves, namely, incident and reflected waves, $V_{incident} = V_{in}e^{-j\beta r}$ and $V_{reflected} = V_{ref}e^{j\beta r}$.
3. Four measured voltages satisfy the following equations:

$$\begin{aligned} V(r_1, f) &= V_{in}e^{-j\beta_{in}r_1} + V_{ref}e^{j\beta_{ref}r_1} \\ V(r_2, f) &= V_{in}e^{-j\beta_{in}r_2} + V_{ref}e^{j\beta_{ref}r_2} \\ V(r_3, f) &= V_{in}e^{-j\beta_{in}r_3} + V_{ref}e^{j\beta_{ref}r_3} \\ V(r_4, f) &= V_{in}e^{-j\beta_{in}r_4} + V_{ref}e^{j\beta_{ref}r_4} \end{aligned} \tag{J.1}$$

where $V(r_{1,2,3,4}, f)$ is a complex number measured at r_1, r_2, r_3 , and r_4 along the feed structure, respectively. The variables V_{in} , V_{ref} , β_{in} , and β_{ref} are the real

- numbers (assuming the loss is ignored) and the unknowns that will be solved for.
4. Solving the equations above using the Prony's method to obtain the incident and reflected wave information. However, if we directly solve the equations (J.1), we cannot guarantee the relationship: $\beta_{in}=\beta_{ref}$.
 5. The modified Prony's method is described as follows:

$$\begin{aligned}
 V_0 &= V_{in} + V_{ref} \\
 V_1 &= V_{in}Z_1 + V_{ref}Z_2 \\
 V_2 &= V_{in}Z_1^2 + V_{ref}Z_2^2 \\
 V_3 &= V_{in}Z_1^3 + V_{ref}Z_2^3
 \end{aligned} \tag{J.2}$$

where $V(r_1, f) = V_0$, $V(r_2, f) = V_1$, $V(r_3, f) = V_2$, and $V(r_4, f) = V_3$. We assume that $r_1=0$, $r_2= r$, $r_3= 2r$, and $r_4= 3r$. The root of (J.2) satisfies $a_2 + a_1Z - Z^2 = 0$ and a_1 and a_2 are determined by:

$$\begin{aligned}
 a_2V_0 + a_1V_1 - V_2 &= 0 \\
 a_2V_1 + a_1V_2 - V_3 &= 0 \\
 a_2 &= -1
 \end{aligned} \tag{J.3}$$

where the criterion $a_2=-1$ is due to the criterion $Z_1Z_2=1$, namely, $\beta_{in}=\beta_{ref}$. The experiments demonstrated that the solution from the modified Prony's method is better than the original Prony's method for the lower frequencies. We only need to measure one voltage at each output port to calculate the transmission coefficients.

J.2 NORMALIZED S-PARAMETER

A two-port network system is shown in Figure J.1 [2].

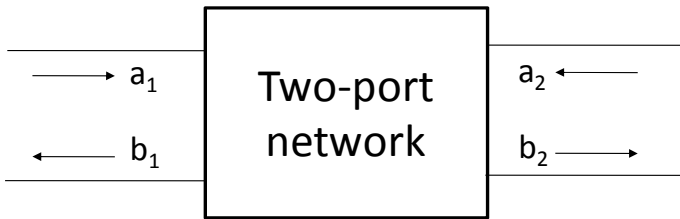


Figure J.1 Two-port network showing incident waves (a_1 , a_2) and reflected waves (b_1 , b_2) used in S-parameter definitions.

It is convenient to assume that the reference impedance Z_i is positive and real, and all variables and parameters will be referenced to a single positive real impedance, Z_0 . The independent variables a_1 and a_2 are normalized incident voltages, as follows:

$$a_1 = \frac{V_1 + I_1 \sqrt{Z_0}}{2\sqrt{Z_0}} \quad (\text{J.4a})$$

$$a_2 = \frac{V_2 + I_2 \sqrt{Z_0}}{2\sqrt{Z_0}} \quad (\text{J.4b})$$

$$b_1 = \frac{V_1 - I_1 \sqrt{Z_0}}{2\sqrt{Z_0}} \quad (\text{J.4c})$$

$$b_2 = \frac{V_2 - I_2 \sqrt{Z_0}}{2\sqrt{Z_0}} \quad (\text{J.4d})$$

The S-parameters s_{11} , s_{22} , s_{21} , and s_{12} are expressed as:

$$s_{11} = \left. \frac{b_1}{a_1} \right|_{a_2=0} = \frac{V_1 - I_1 \sqrt{Z_0}}{V_1 + I_1 \sqrt{Z_0}} \quad (\text{J.5a})$$

$$s_{21} = \left. \frac{b_2}{a_1} \right|_{a_2=0} = \frac{V_2 - I_2 \sqrt{Z_0}}{V_1 + I_1 \sqrt{Z_0}} \quad (\text{J.5b})$$

$$s_{22} = \left. \frac{b_2}{a_2} \right|_{a_1=0} = \frac{V_2 - I_2 \sqrt{Z_0}}{V_2 + I_2 \sqrt{Z_0}} \quad (\text{J.5c})$$

$$s_{12} = \left. \frac{b_1}{a_2} \right|_{a_1=0} = \frac{V_1 - I_1 \sqrt{Z_0}}{V_2 + I_2 \sqrt{Z_0}} \quad (\text{J.5d})$$

Another advantage of S-parameters springs from the simple relationship between the variables a_1 , a_2 , b_1 , and b_2 , and various power waves:

$|a_1|^2$: Power incident on the input of the network, or, power available from a source impedance Z_0 .

$|b_1|^2$: Power reflected from the input port of the network, or power available from a Z_0 source minus the power delivered to the input of the network.

$|a_2|^2$: Power incident on the output of the network, or, power reflected from the load

$|b_1|^2$: Power reflected from the output port of the network, or, power incident on the load, or, power that would be delivered to a Z_0 load.

The previous four equations show that s-parameters are simply related to power gain and mismatch loss, quantities that are often of more interest than the corresponding voltage functions:

$$|s_{11}|^2 = \frac{\text{Power reflected from the network input}}{\text{Power incident on the network input}}$$

$$|s_{22}|^2 = \frac{\text{Power reflected from the network output}}{\text{Power incident on the network output}}$$

$$\begin{aligned} |s_{21}|^2 &= \frac{\text{Power delivered to a } Z_0 \text{ load}}{\text{Power available from } Z_0 \text{ source}} \\ &= \text{Transducer power gain with } Z_0 \text{ load and source} \end{aligned}$$

$$|s_{12}|^2 = \text{Reverse transducer power gain with } Z_0 \text{ load and source}$$

J.3 MODE S-PARAMETER

For the waveguide and coaxial structures, the excitation and output are usually the field distribution inside the structure. To calculate the S-parameters for this type of problem, we need to define a mode voltage and current that are used to derive the S-parameters [1].

$$\begin{aligned} V(r, f) &= \int_S \vec{E}(r', f) \cdot \vec{h}(\vec{r}', f) \cdot d\vec{s} \\ I(r, f) &= \int_S \vec{e}(r', f) \cdot \vec{H}(\vec{r}', f) \cdot d\vec{s} \end{aligned} \tag{J.6}$$

where $\vec{E}(r', f)$ and $\vec{e}(r', f)$ are the measured aperture electric field pattern and the normalized mode pattern, and $\vec{H}(r', f)$ and $\vec{h}(r', f)$ are the measured aperture magnetic field pattern and the normalized mode pattern. The normalized mode patterns $\vec{e}(r', f)$ and $\vec{h}(r', f)$ are obtained by the analytic approach for the rectangular and circular waveguides.

$$Z = \sqrt{\frac{V(r, f) \cdot V'(r, f)}{I(r, f) \cdot I'(r, f)}} \quad (\text{J.7})$$

where $V'(f)$ and $I'(f)$ are the variations of the mode voltage and current in the propagation direction, and can be expressed as:

$$\begin{aligned} V'(r, f) &= \frac{V(r_2, f) - V(r_1, f)}{r_2 - r_1} \\ I'(r, f) &= \frac{I(r_2, f) - I(r_1, f)}{r_2 - r_1} \end{aligned} \quad (\text{J.8})$$

In the formula (J.6), the aperture field distributions $\vec{E}(r', f)$ and $\vec{H}(r', f)$ are calculated in the FDTD simulation through the Fourier transform. However, $\vec{e}(r', f)$ and $\vec{h}(r', f)$ are the input parameters that generated through either the analytic approach for the waveguide with the rectangular or circular cross section [3], or the numerical method such as FDFD to extract the mode pattern [4].

References

- [1] A. Taflove and S. Hagness, (eds.), *Computational Electromagnetics: The Finite-Difference Time-Domain Method*, 3rd ed., Norwood, MA: Artech House, 2005.
- [2] <http://www.sss-mag.com/pdf/an-95-1.pdf>.
- [3] C. Balanis, *Advanced Engineering Electromagnetics*, New York: John Wiley & Sons, 1995.
- [4] W. Yu, et al., *Parallel Finite-Difference Time-Domain Method*, Norwood, MA: Artech House, 2006.

Appendix K

Debye Model Construction

In practical applications, the material manufacturers usually provide a parameter list that shows the variation of dispersive material parameters with the frequencies, as shown in Table K.1. We need to fit them into an analytic formulation that can be transformed to the time-domain expression through Fourier transformation, and then the material can be simulated in the FDTD simulation for the broadband results.

Table K.1 Dispersive Material Parameter List

Frequency	Relative Permittivity	Loss Tangent
ω_1	$\varepsilon_r'(\omega_1)$	δ_1
ω_2	$\varepsilon_r'(\omega_2)$	δ_2

To handle the dispersive medium in the FDTD simulation, we can convert the model in the listed format in Table K.1 into the Debye model, which has been proved having the time-domain expression. The typical Debye model is expressed as follows [1, 2]:

$$\begin{aligned}
 \varepsilon &= \varepsilon_0 \left(\varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + j\omega\tau_0} - \frac{j\sigma'}{\omega\varepsilon_0} \right) \\
 &= \varepsilon_0 \left(\varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + \omega^2\tau_0^2} \right) - j\varepsilon_0 \left(\frac{(\varepsilon_s - \varepsilon_\infty)\omega\tau_0}{1 + \omega^2\tau_0^2} + \frac{\sigma'}{\omega\varepsilon_0} \right) \\
 &= \varepsilon_0 \left(\varepsilon_\infty + \frac{\Delta\varepsilon}{1 + \omega^2\tau_0^2} \right) - j\varepsilon_0 \left(\frac{\Delta\varepsilon\omega\tau_0}{1 + \omega^2\tau_0^2} + \frac{\sigma'}{\omega\varepsilon_0} \right)
 \end{aligned} \tag{K.1}$$

where ε_∞ is the infinite frequency permittivity, ε_s is the static permittivity, σ' is the conductivity, τ_0 is the relaxation time, and $\Delta\varepsilon = \varepsilon_s - \varepsilon_\infty$. The four equations

are required to solve for the four unknowns, ε_∞ , $\Delta\varepsilon$, τ_0 and σ' in (K.1). Using the relationships:

$$\varepsilon = \varepsilon_0 (\varepsilon_r' - j\varepsilon_r'') = \varepsilon_0 (\varepsilon_r' - j\varepsilon_r' \delta) \quad (\text{K.2})$$

We can get the following four equations:

$$\varepsilon_r'(\omega_1) = \varepsilon_\infty + \frac{\Delta\varepsilon}{1 + \omega_1^2 \tau_0^2} \quad (\text{K.3a})$$

$$\varepsilon_r''(\omega_1) = \frac{\Delta\varepsilon \omega_1 \tau_0}{1 + \omega_1^2 \tau_0^2} + \frac{\sigma'}{\omega_1 \varepsilon_0} \quad (\text{K.3b})$$

$$\varepsilon_r'(\omega_2) = \varepsilon_\infty + \frac{\Delta\varepsilon}{1 + \omega_2^2 \tau_0^2} \quad (\text{K.3c})$$

$$\varepsilon_r''(\omega_2) = \frac{\Delta\varepsilon \omega_2 \tau_0}{1 + \omega_2^2 \tau_0^2} + \frac{\sigma'}{\omega_2 \varepsilon_0} \quad (\text{K.3d})$$

Solving (K.3), we have:

$$\tau_0 = \frac{\varepsilon_r''(\omega_1) - \varepsilon_r''(\omega_2)}{\omega_1 \varepsilon_r''(\omega_1) - \omega_2 \varepsilon_r''(\omega_2)} \quad (\text{K.4a})$$

$$\Delta\varepsilon = \frac{(\varepsilon_r''(\omega_1) - \varepsilon_r''(\omega_2))(1 + \omega_1^2 \tau_0^2)(1 + \omega_2^2 \tau_0^2)}{(\omega_2^2 - \omega_1^2) \tau_0^2} \quad (\text{K.4b})$$

$$\varepsilon_\infty = \varepsilon_r'(\omega_1) - \frac{\Delta\varepsilon}{1 + \omega_1^2 \tau_0^2} \quad (\text{K.4c})$$

$$\sigma' = \omega_1 \varepsilon_r''(\omega_1) \varepsilon_0 - \frac{\Delta\varepsilon \omega_1^2 \tau_0 \varepsilon_0}{1 + \omega_1^2 \tau_0^2} \quad (\text{K.4d})$$

where $\varepsilon_r'(\omega_1)$, $\delta_1 = \varepsilon_r''(\omega_1)/\varepsilon_r'(\omega_1)$, $\varepsilon_r'(\omega_2)$, and $\delta_2 = \varepsilon_r''(\omega_2)/\varepsilon_r'(\omega_2)$ are the material parameters provided by the manufacturers. For example, the parameters of material FR4 is listed in Table K.2. Using (K.4) gets the values of the parameters in the Debye model, which is shown in Table K.3.

Table K.2 FR4 Material Parameter List

Frequency	Relative Permittivity	Loss Tangent
0.1 GHz	4.0	0.013
24 GHz	3.8	0.025

Table K.3 Debye Model Parameters for FR4 Material

ϵ_{∞}	ϵ_s	τ_0	σ'
3.755	4.0	0.0879 ns	4.414e-5

References

- [1] J. Young and R. Nelson, "A Summary and Systematic Analysis of FDTD Algorithms for Linearly Dispersive Media," *IEEE Antennas Propagation Magazine*, Vol. 43, No. 1, February 2001, pp. 61–77.
- [2] A. Taflove and S. Hagness, (eds.), *Computational Electromagnetics: The Finite-Difference Time-Domain Method*, 3rd ed., Norwood, MA: Artech House, 2005.

Appendix L

Geometry Mapping Technique

The project modeling is not related to the FDTD method directly; however, it plays a critical role in the FDTD simulation [1, 2]. In this appendix, we introduce a way that a complex model, for example, a cylinder FSS or cone FSS structure, can be mapped from a simple planar FSS structure, as shown in Figure L.1.

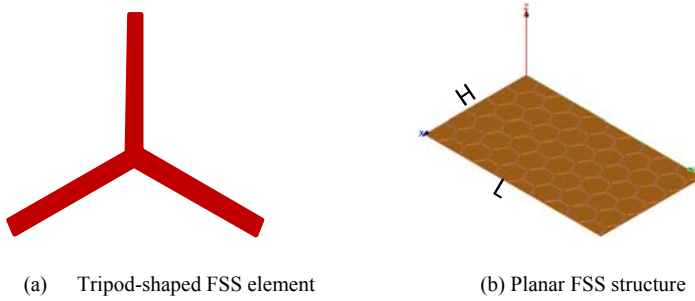


Figure L.1 A planar FSS configuration (a) tripod-shaped FSS element and (b) planar FSS structure.

The width and length of the planar FSS structure are H and L , respectively (see Figure L.1(b)). The axis of cylindrical FSS structure that we will construct is along the x -direction. The radius of cylinder is r , as shown in Figure L.2.

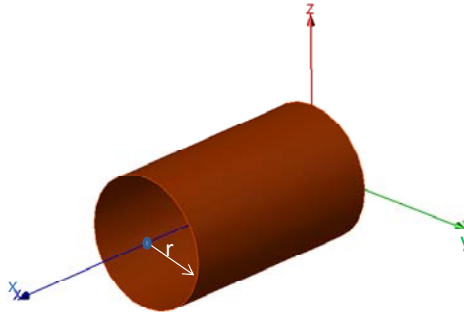


Figure L.2 Cylinder on which the designed FSS surface will be mounted.

A cylinder shown in Figure L.2 (along the x -direction) can be expressed as in this format:

$$z = r \cos(\varphi) \quad (\text{L.1a})$$

$$y = r \sin(\varphi) \quad (\text{L.1b})$$

$$x = h \quad (\text{L.1c})$$

where (x_0, y_0) is a point on the planar FSS sheet and h is the height of the point (x, y) . Using the relationship:

$$r = \frac{L}{2\pi}, \quad \varphi = 2\pi \frac{x_0}{L}, \quad h = y_0$$

(L.1) can be expressed as:

$$z = \frac{L}{2\pi} \cos\left(2\pi \frac{x_0}{L}\right) \quad (\text{L.2a})$$

$$y = \frac{L}{2\pi} \sin\left(2\pi \frac{x_0}{L}\right) \quad (\text{L.2b})$$

$$x = y_0 \quad (\text{L.2c})$$

Using (L.2), we can map the planar FSS structure to a cylindrical FSS structure. If the constructed cylinder is along the z -direction, we need to switch the variables x and z in (L.1). For example, if the planar FSS structure is located in the XOY plane, the width H and length L are taken to be 50 mm and 86.6 mm, respectively. We can get the cylindrical FSS structure, as shown in Figure L.3.

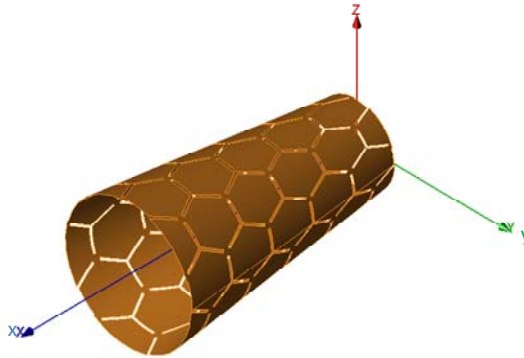


Figure L.3 Constructed cylindrical FSS structure.

Next, we show you how to map a planar FSS structure in the XOY plane to a conical FSS structure, as shown in Figure L.4. To this end, we need to know the geometry information marked in Figure L.4 for the FSS structure mapping.

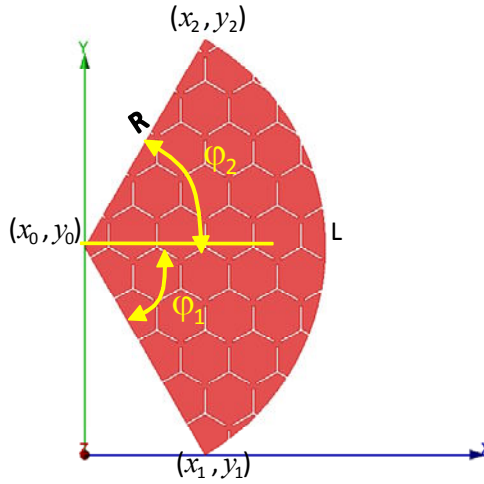


Figure L.4 Sector planar FSS structure.

The sector FSS structure that will be constructed is shown in Figure L.5. The radius of bottom circle of the cone is shown in Figure L.5.

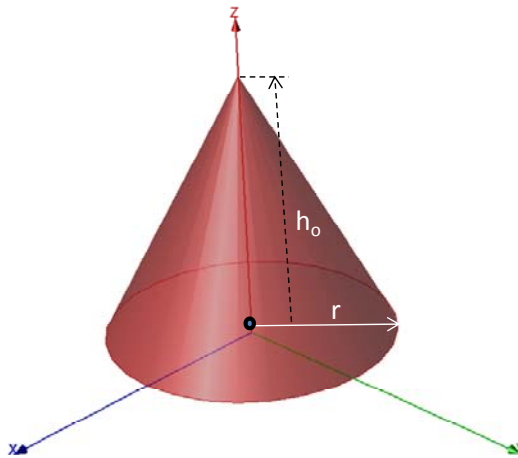


Figure L.5 Cone which the designed FSS surface will be mounted on.

Express the intermediate variables r , h_0 , and φ as follows [1, 2]:

$$r = \frac{L}{2\pi} \quad (\text{L.3a})$$

$$h_0 = \sqrt{R^2 - r^2} \quad (\text{L.3b})$$

$$\varphi = \left[\arctan\left(\frac{y - y_0}{x - x_0}\right) - \varphi_1 \right] \frac{2\pi}{\varphi_2 - \varphi_1} \quad (\text{L.3c})$$

where, $\varphi_1 = \arctan\left(\frac{y_1 - y_0}{x_1 - x_0}\right)$, $\varphi_2 = \arctan\left(\frac{y_2 - y_0}{x_2 - x_0}\right)$. The mapping relationship is expressed the following format:

$$x = r \cos(\varphi) \frac{h_0 - h}{h_0} \quad (\text{L.4a})$$

$$y = r \sin(\varphi) \frac{h_0 - h}{h_0} \quad (\text{L.4b})$$

$$z = h \quad (\text{L.4c})$$

For example, if the conical FSS structure is located in the XOY plane, the geometry parameters are listed in Table L.1. We can get the conical FSS structure, as shown in Figure L.6.

Table L.1 Geometry Information of Cone FSS Structure

x_0	0 mm
y_0	43.3 mm
x_1	25 mm
y_1	0 mm
x_2	25 mm
y_2	86.6 mm
R	50 mm
L	116.35 mm
R	18.459 mm
H	46.468 mm
φ_2	1.047
φ_1	-1.047
φ	$0 \sim 2\pi$

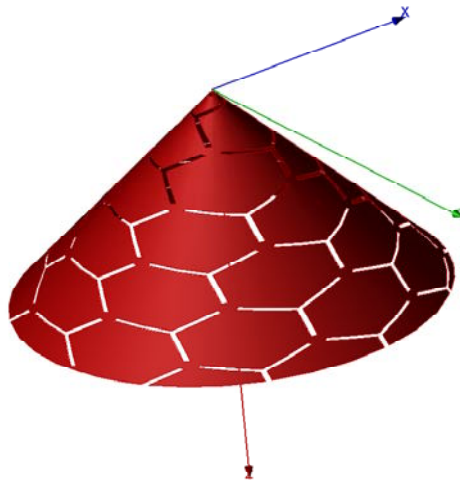


Figure L.6 Constructed conical FSS structure.

Finally, we map a planar FSS structure in the XOY plane to a circular frustum FSS structure, as shown in Figure L.7. To this end, we need to know the geometry information marked in Figure L.8 for the FSS structure mapping.

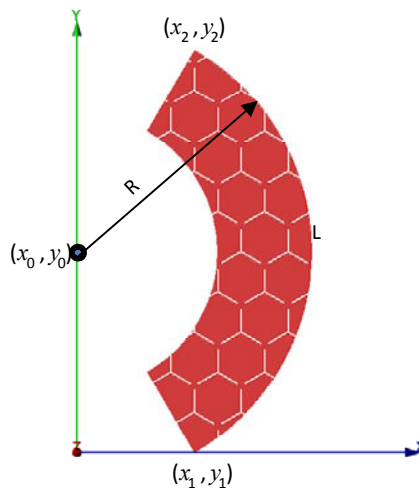


Figure L.7 Sector planar FSS structure.

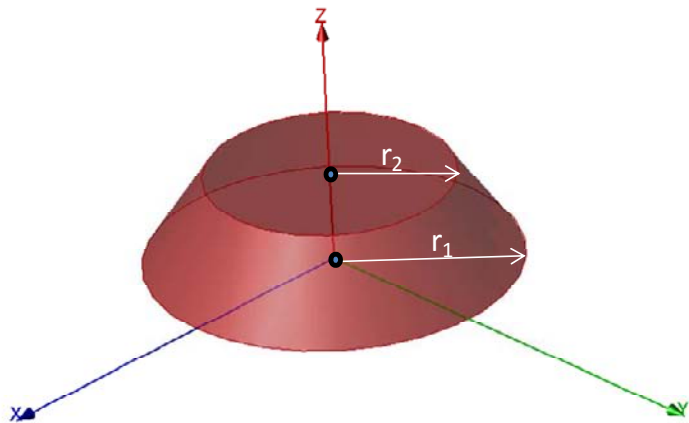


Figure L.8 Cone structure on which the designed FSS surface will be mounted.

For example, if the planar FSS structure is located in the X0Y plane, the geometry parameters are listed in Table L.2. We can get the conical FSS structure, as shown in Figure L.9.

Table L.2 Geometry Information of Cone FSS Structure	
x_0	0 mm
y_0	43.3 mm
x_1	25 mm
y_1	0 mm
x_2	25 mm
y_2	86.6 mm
R	50 mm
L	116.35 mm
R_1	16.6389 mm
R_2	11.075 mm
H	46.468 mm
φ_2	1.055
φ_1	-1.055
φ	$0 \sim 2\pi$

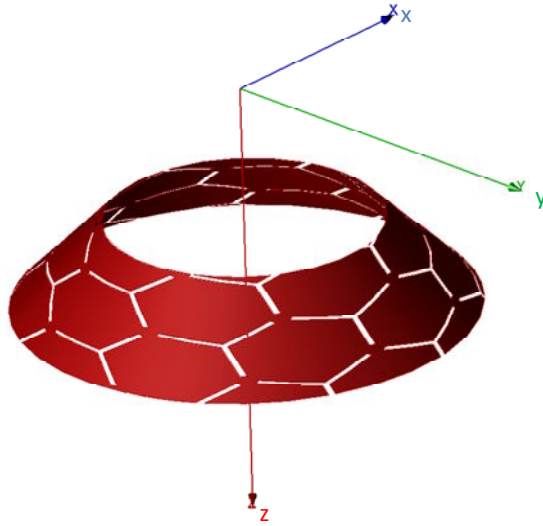


Figure L.9 Constructed conical FSS structure.

When we do the geometry mapping, the ill-conditioned cases may occur if the geometry parameters are selected improperly. Sometimes, to avoid the ill-conditioned geometry such as in the case divided by zero, we may cut a small circle around the point (x_0, y_0) or set the angle φ from 0° to 359° . Sometimes, we may need to split the transform procedure into two steps to get the desired pattern. For example, to get a conical FSS structure, we need to do the first transform using the following formulas:

$$x = \cos((\arctan((y-43.3)/x)+1.0455)/1.0455*\pi)*16.6389*\sqrt{x*x+(y-43.3)*(y-43.3)}/50$$

$$y = \sin((\arctan((y-43.3)/x)+1.0455)/1.0455*\pi)*16.6389*\sqrt{x*x+(y-43.3)*(y-43.3)}/50$$

$$z = (1-\sqrt{x*x+(y-43.3)*(y-43.3)})/50$$

The conical FSS structure is the desired shape in the z -direction. The second transform using the following formula will allow us to get the desired conical FSS pattern.

$$\begin{aligned} x &= x \\ y &= y \\ z &= z*47.15 \end{aligned}$$

References

- [1] W. Chinn, *First Concepts of Topology: The Geometry of Mappings of Segments, Curves, Circles, and Disks*, New York: Random House, 1966.
- [2] R. Kurazume, M. Wheeler, and K. Ikeuchi, *Mapping Textures on 3D Geometric Model Using Reflectance Image*, New York: Wiley-Interscience, 2005.

Appendix M

PC Cluster Optimization

As benefits from the MPI, OpenMP, and high-performance network devices, the parallel processing techniques [1, 2] have been applied in many software packages including most commercial electromagnetic software and home-grown electromagnetic simulation codes. However, there still exist two major challenging issues in the parallel processing applications. First, a user-friendly interface including the project design and data postprocessing through a Web browser is the most important, and we have to ensure the data and cluster security when the users can be access to the cluster through the Internet. Second, for the required performance, the cost should be reduced to an affordable level. Next, we use a simple example to demonstrate the performance of parallel conformal FDTD method on the different hardware platforms that are installed with different network systems, as shown in Table M.1.

Table M.1 Configuration of Different Hardware Platforms

	InfiniBand (Bandwidth: 10 GB, Latency: 2.6 μ m)	Gigabit Ethernet (Bandwidth: 1 GB, Latency: 130 μ m)	Gigabit Ethernet (Bandwidth: 1 GB, Latency: 130 μ m)
CPU type	Intel Q6700, 2.66 GHz	Intel Q6700, 2.66 GHz	Intel Q6700, 2.66 GHz
Operating system	Linux	Linux	Linux
Network card	InfiniBand card	Gigabit card	Gigabit card
Number of nodes	8	8	8
Acceleration	6	4.8	1.4
Comments	InfiniBand network system	Optimized gigabit network	Regular gigabit network

For example, if we use DELL computers with the integrated or independent Gigabit network card and Gigabit Ethernet switch (<http://www.dell.com>), the performance of the parallel FDTD code is shown as the curve “Regular Gigabit Ethernet” in Figure M.1. If we select an ASUS computer (<http://www.asus.com>) installed with D-Link Gigabit network card and Ethernet (<http://www.dlink.com>), the performance of the parallel FDTD code is shown as the curve “Optimized Gigabit Ethernet” in Figure M.1. The InfiniBand network system gets much better performance when the number of nodes becomes larger. It is observed from Figure M.1 that we can reach only slightly less optimal performance than the InfiniBand network system using the optimized Gigabit Ethernet network system at a much lower cost. The acceleration factor in Figure M.1 is related to the performance of the parallel FDTD code on one quad core computer. For example, if the simulation time of the parallel FDTD code is 100 seconds on one computer and 50 seconds on two computers for the same problem, the acceleration factor of the parallel FDTD code on the computer cluster is 2.

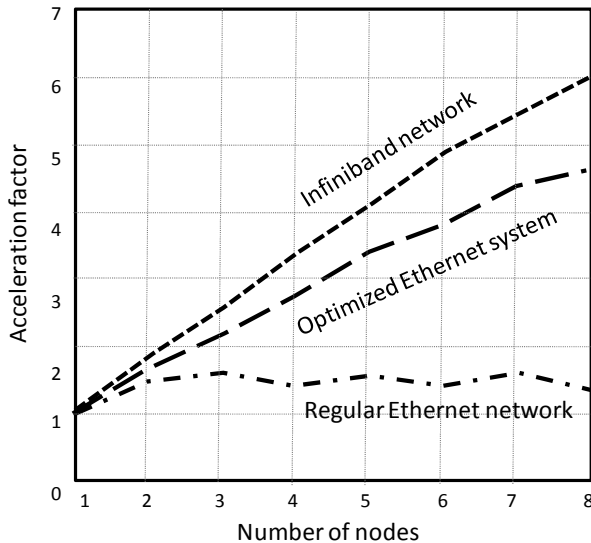


Figure M.1 The performance of parallel FDTD code on the different cluster.

References

- [1] M. Oguchi, et al., “Optimizing Protocol Parameters to Large Scale PC Cluster and Evaluation of its Effectiveness with Parallel Data Mining,” *Proceeding of the Seventh International Symposium on High Performance Distributed Computing*, 1998.
- [2] W. Yu, et al., “Performance Analysis of Parallel FDTD Method on the Different Platforms,” *Microwave and Optical Technology Letters*, Vol. 50, No. 9, September 2008, pp. 2465–2467.

About the Authors



Wenhua Yu joined the Department of Electrical Engineering at the Pennsylvania State University in 1996, where he is the associate director of the Electromagnetic Communication Lab. He received a Ph.D. in electrical engineering from the Southwest Jiaotong University in 1994. Dr. Yu worked at the Beijing Institute of Technology as a postdoctoral research associate and has published seven books related to the FDTD method, parallel processing techniques, the software development technique, and simulation techniques. Dr. Yu has published more than 150 technical papers and four book chapters. He founded the company Computer and Communication Unlimited (www.2comu.com) and serves as its president and CEO. He is a senior member of the IEEE. His research interests include computational electromagnetic methods, software development techniques, parallel processing techniques, and the simulation and design of antennas and microwave circuits.



Xiaoling Yang is a research associate in the Electromagnetic Communication Lab at the Pennsylvania State University. He received a B.S. and an M.S. in communication and mathematics from Tianjin University in 2001 and 2004, respectively. He has published three books related to the FDTD method, parallel processing techniques, the software development technique, and simulation techniques. He has published more than 20 technical papers. His research interests include numerical methods, visual languages, and software development.



Yongjun Liu received a B.S. and an M.S. in electrical engineering from Tsinghua University in 1991, and from the Communication University of China in 1994, respectively. He has worked at Beijing Broadcasting Institute as an associate professor. He currently works at the Electromagnetic Communication Lab at the Pennsylvania State University. He has published four books related to the FDTD method, parallel processing techniques, the software development technique, and simulation techniques. He has published more than 20 technical papers. His research interests include computational electromagnetics and its visualization techniques and visual languages.



Raj Mittra is a professor in the Electrical Engineering Department at the Pennsylvania State University. He is also the director of the Electromagnetic Communication Lab, which is affiliated with the Communication and Space Sciences Lab of the Electrical Engineering Department. Prior to joining Penn State, Dr. Mittra was a professor in electrical and computer engineering at the University of Illinois at Urbana-Champaign. Dr. Mittra is also the president of RM Associates, a consulting organization that provides services to industrial and governmental organizations, both in the United States and abroad. He is a life fellow of the IEEE and a past president of the Antenna and Propagation Society, and he has served as the editor of the *Transactions of the Antennas and Propagation Society*. He won the Guggenheim Fellowship Award in 1965, the IEEE Centennial Medal in 1984, the IEEE Millennium Medal in 2000, and the IEEE/APS Distinguished Achievement Award in 2002. He has been a visiting professor at Oxford University, Oxford, England, and at the Technical University of Denmark, Lyngby, Denmark.

Dr. Mittra has published over 1,000 technical papers and more than 40 books or book chapters on various topics related to electromagnetics, antennas, microwaves, and electronic packaging. He also has three patents on communication antennas. He has advised over 80 Ph.D. students and 80 M.S. students and has mentored approximately 50 postdoctoral research associates and visiting scholars in the EMC Laboratories at the University of Illinois and Penn State.



Akira Muto works in the Advanced EMC Technology Development Section, EDA Design Technology Solution Department, MONO-ZUKURI Technology Division, Production Group, at Sony. He worked for Corporate R&D Advanced Material Laboratories as a signal integrity engineer at the Sony Corporation, Japan, from 2001 to 2010. He was a visiting scholar at the Electromagnetic Communication Lab at the Pennsylvania State University from February 2009 to August 2010. He received a B.S. and an M.S in electrical and computer engineering from Yokohama National University, Japan, in 1995 and 2001, respectively. His research interests include hardware design, EM simulation, and high-performance computing.

Index

10-gigabit	44	cylindrical FSS	239
3-D SAR distribution.....	139		
absorbing boundary	82	dielectric sphere	108
accepted efficiencies of antenna	197	dipole	51
accepted power	196	dispersive medium	82
active reflection coefficient	201	dissipation power	195, 196
alternative direction implicit.....	5		
AMD.....	40	electromagnetic band gap	127
		electromagnetic band gap structure...	127
banded microwave connector	60	envelope correlation coefficient.....	207
Beowulf class clusters	41		
bistatic radar cross section	108	fast multiple method	13
bow-tie antennas	82	FDTD performance.....	33, 34
buried pipe	83	FEKO	90
		finite element method	13
cache optimization.....	25	finite integration technique	15
card reader	179	finite-difference time-domain	1
Cartesian coordinate system	2	flat reflector antenna	171
Category 5	45	Floating point unit.....	20
Category 6	45	forward efficiency.....	195
cell phone antenna	117	Fourier transform.....	3
cell phone model.....	147	four-way UWB power divider	157
central processing unit.....	98	FR4	236
chip model	64	Frequency selective surface	239
circular polarization.....	213		
cloaking structure	183	GEMS	91
cluster setup	247	GEMS workstation	92
comb-shaped element	153	Gigabit Ethernet.....	43
computer aided design model	50	graphics processing unit.....	98
cone FSS.....	244	ground penetrating radar	81
coupling.....	70		
Courant condition	6	hardware platform.....	92
CST	90	HFSS	89
		high gain antenna.....	159

high performance cluster	42	periodic boundary condition	5
high Q systems	140	periodic structure	78, 80
IEEE Standards 1528	131	personal computer cluster	41
incident efficiency	197	portable batch system	41
incident power	195	prefetch	26
InfiniBand	44	Prony's method	230
Intel	40	pseudospectrum time-domain	5
intermediate cluster	34		
		radar cross section	108
linear operator	12	reflector antenna	165
loss tangent	211		
		SAM head model	147
match loads	66	scalar formulation	14
Maxwell's curl equations	2	signal integrity	174
Maxwell's equations	4	single instruction and multiple data	20
mean effective gain	207	slot coupling	69
mesh generation technique	47	S-parameter	229
message passing interface	41	specific absorption rate	151
method of moments	11	streaming SIMD extensions	21
microwave connector	84	Sun grid engine	41
microwave filter	72	symmetric multiprocessing	28
multiresolution time domain	5		
		task parallelization and bundling	26
nonuniform memory architecture	28	terminator	102
		time convolution PML	8
open multiple processing	40	time-domain impedance	226
		time-domain reflectometry	86, 225
parabolic reflector antenna	159	total active reflection coefficient	205
parallel processing technique	37	two-port antenna	67
partial symmetric problem	165		
patch antenna	92	vector arithmetic logic unit	20
patch antenna array	74	vector fitting	217
perfect electric conductor	5	vector formulation	14
perfect electric conductor boundary	7	Vivaldi antenna	55, 99
perfect electric conductor shell	49		
perfect magnetic conductor boundary ...	7	waveguide filter	141
perfectly matched layer	7		
periodic boundary condition	126	Yee's scheme	1

Recent Titles in the Artech House Electromagnetics Series

Tapan K. Sarkar, Series Editor

Advanced FDTD Methods: Parallelization, Acceleration, and Engineering Applications, Wenhua Yu, et al.

Advances in Computational Electrodynamics: The Finite-Difference Time-Domain Method, Allen Taflove, editor

Analysis Methods for Electromagnetic Wave Problems, Volume 2, Eikichi Yamashita, editor

Analytical and Computational Methods in Electromagnetics, Ramesh Garg

Analytical Modeling in Applied Electromagnetics, Sergei Tretyakov

Applications of Neural Networks in Electromagnetics, Christos Christodoulou and Michael Georgiopoulos

CFDTD: Conformal Finite-Difference Time-Domain Maxwell's Equations Solver, Software and User's Guide, Wenhua Yu and Raj Mittra

The CG-FFT Method: Application of Signal Processing Techniques to Electromagnetics, Manuel F. C  tedra, et al.

Computational Electrodynamics: The Finite-Difference Time-Domain Method, Second Edition, Allen Taflove and Susan C. Hagness

Electromagnetic Waves in Chiral and Bi-Isotropic Media, I. V. Lindell, et al.

Engineering Applications of the Modulated Scatterer Technique, Jean-Charles Bolomey and Fred E. Gardiol

Fast and Efficient Algorithms in Computational Electromagnetics, Weng Cho Chew, et al., editors

Fresnel Zones in Wireless Links, Zone Plate Lenses and Antennas, Hristo D. Hristov

Grid Computing for Electromagnetics, Luciano Tarricone and
Alessandra Esposito

High Frequency Electromagnetic Dosimetry,
David A. Sánchez-Hernández, editor

*Iterative and Self-Adaptive Finite-Elements in Electromagnetic
Modeling*, Magdalena Salazar-Palma, et al.

Numerical Analysis for Electromagnetic Integral Equations,
Karl F. Warnick

Parallel Finite-Difference Time-Domain Method, Wenhua Yu, et al.

Practical Applications of Asymptotic Techniques in Electromagnetics,
Francisco Saez de Adana, et al.

Quick Finite Elements for Electromagnetic Waves, Giuseppe Pelosi,
Roberto Coccioli, and Stefano Selleri

*Understanding Electromagnetic Scattering Using the Moment
Method: A Practical Approach*, Randy Bancroft

Wavelet Applications in Engineering Electromagnetics,
Tapan K. Sarkar, Magdalena Salazar-Palma, and Michael C. Wicks

For further information on these and other Artech House titles,
including previously considered out-of-print books now available
through our In-Print- Forever® (IPF®) program, contact:

Artech House Publishers

685 Canton Street

Norwood, MA 02062

Phone: 781-769-9750

Fax: 781-769-6334

e-mail: artech@artechhouse.com

Artech House Books

16 Sussex Street

London SW1V 4RW UK

Phone: +44 (0)20 7596 8750

Fax: +44 (0)20 7630 0166

e-mail: artech-uk@artechhouse.com

Find us on the World Wide Web at: www.artechhouse.com
