

MULTILEVEL MULTIPOLE-FREE FAST ALGORITHM
FOR ELECTROMAGNETIC SCATTERING PROBLEMS IN LAYERED MEDIA

BY

MICHAEL ANDREW SAVILLE

B.S., Texas A&M University, 1997
M.S., Air Force Institute of Technology, 2000

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2006

Urbana, Illinois

Report Documentation Page			Form Approved OMB No. 0704-0188		
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 29 SEP 2006		2. REPORT TYPE N/A		3. DATES COVERED -	
4. TITLE AND SUBTITLE Multilevel Multipole-Free Fast Algorithm For Electromagnetic Scattering Problems In Layered Media				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Illinois at Urbana-Champaign				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited					
13. SUPPLEMENTARY NOTES The original document contains color images.					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 160	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

MULTILEVEL MULTIPOLE-FREE FAST ALGORITHM FOR ELECTROMAGNETIC SCATTERING PROBLEMS IN LAYERED MEDIA

Michael Andrew Saville, Ph.D.
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, 2006
Weng Cho Chew, Advisor

A multilevel multipole-free algorithm is presented for solving electromagnetic scattering problems in the vicinity of a half space or layered medium. By replacing the multipole expansion in the fast inhomogeneous plane wave algorithm (FIPWA) with a multipole-free expansion, this new algorithm is simpler to derive and retains $\mathcal{O}(N \log N)$ scaling in memory and processing time. To develop this new algorithm, known as the multipole-free fast inhomogeneous plane wave algorithm (MF-FIPWA), error control is established for arbitrary accuracy.

In addition, comparison of the memory usage and simulation time is presented for FIPWA and MF-FIPWA for moderate to large scale problems. Various alternate approaches to implementing MF-FIPWA are discussed in terms of how the fast algorithms set up translation matrices and where gains can be made. Finally, details of the advantages of using non-uniform sampling are provided. Results show 30% savings in memory usage and up to 20% savings in computing the matrix-vector product.

To my wife and children.

ACKNOWLEDGMENTS

I sincerely thank Professor Weng Cho Chew for his superb teaching, continuous encouragement, and generous support throughout these challenging three years. A scholar and a gentleman, I have learned much more from him than just electromagnetics research. I also thank Professors Jianming Jin, Andreas Cangellaris, and Michael Oelze for their teaching and kind discussions about my research. I am very grateful to have these esteemed faculty members on my committee.

Many people inside and outside the university system had an impact on my success. To Professor Jose Shutte-Aine, thank you for all the support with the CCEM cluster. To Professor Minh Do, thank you for your advice during the morning bus rides to campus and your help with understanding some finer points of signal processing. I give many thanks to the visiting professors and postdoctoral researchers for sharing their expertise and friendship: visiting Professors Levant Gurel, Jukka Sarvas, Shinichiro Ohnuki, and postdoctoral researchers Dr. I-Ting Chiang and Dr. Meisong Tong. I thank the previous graduates of Prof. Chew: AFIT Professor Michael Hastriter, Dr. Yunhui Chu, Dr. Lijun Jiang, and Dr. Alaeddin Aydiner, for answering so many questions as I returned to the rigors of school, and for being solid examples of excellent researchers.

To my office mates: Matthew Delaquil, Greg Sorenson, and Clayton Davis, I thank you for always offering great insight into the research problems, and for always being a source of encouragement. I sincerely thank my lunch time pal, sounding board for ideas, and gracious computer problem solver: Andrew Hesford. To my fellow colleagues in Prof. Chew's group:

Phil Atkins, Maokun Li, Yuan Liu, Zhiguo Qian, Lin Sun, Jie Xiong, so many times you were willing to discuss ideas with me; you are all such wonderful people that it would be my great honor to work with you again. Also I thank several people outside of our research group, Wayne Weaver and Gouliang Zhang, my “qual” study group, and Jun Meng for helping me build 64-bit applications on the CCEM cluster, and sorting through interpolation of spheres.

In addition I would like to thank the author of FIPWA, Dr. Bin Hu, for his help and patience while I asked many questions to understand the algorithm. I also sincerely thank my friends and neighbors in Champaign who often took an interest in my work, shared their technical expertise, and gave useful perspective. Many times, that is all I needed to take a step forward.

Most of all, I thank my Lord Jesus Christ, and my loving wife and beautiful children for all their love, patience, encouragement, and sacrifice while I pursued this degree. My words cannot express the love and gratitude I have for you.

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

TABLE OF CONTENTS

LIST OF FIGURES	xi
LIST OF TABLES.....	xiii
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.2 Fast Algorithms	2
1.3 Multipole-Free Fast Algorithm	4
1.4 Organization	4
CHAPTER 2 FAST INTEGRAL EQUATION SOLVER.....	5
2.1 Electric Field Integral Equation	5
2.2 Layered Media Dyadic Green's Function	6
2.3 FIPWA for Layered Media	8
2.3.1 Why FIPWA?	8
2.3.2 Factorizing the Green's function	8
2.3.3 Steepest descent path	11
2.3.4 Modified SDP	12
2.3.5 Multilevel implementation	14
2.4 Summary	16
CHAPTER 3 MODIFIED FORM OF FIPWA FOR LAYERED MEDIA..	17
3.1 Introduction	17
3.2 Formulation of the Plane-Wave Algorithm	18
3.3 Deriving the Steepest Descent Path	20
3.3.1 SDPs in lossy medium	22
3.3.2 SDPs in active medium	22
3.3.3 Steepest ascent and constant magnitude paths	24
3.3.4 Modified steepest descent path (M-SDP)	24
3.3.5 Behavior of the integrand along the M-SDP	26
3.4 Cost Analysis	29
3.5 Summary	30

CHAPTER 4 ERROR CONTROL	31
4.1 Introduction	31
4.2 SDP Truncation Error	32
4.3 Extrapolation Error	32
4.4 Control Regions	37
4.5 Summary	40
CHAPTER 5 MULTIPOLE-FREE ALGORITHM	41
5.1 Introduction	41
5.2 Formulation	43
5.2.1 Combined field integral equation	43
5.2.2 Multilevel multipole-free fast algorithm	45
5.2.3 Surface wave contributions and poles	46
5.3 Error Control	47
5.3.1 Integration along the SDPs	47
5.3.2 Interpolation and extrapolation of radiation patterns	50
5.3.3 Interpolation between levels	50
5.4 Numerical Results	51
5.4.1 Validating the multilevel multipole-free algorithm	51
5.4.2 Scaling and efficiency	53
5.4.3 Moderate sized complex target	54
5.5 Summary	56
CHAPTER 6 COMPARISON OF FIPWA AND MF-FIPWA	57
6.1 Introduction	57
6.2 Scope of Comparison	58
6.3 3-D translators for layered medium	59
6.3.1 Matrix representation of the translator	59
6.3.2 FIPWA 3-D O2I-SM translator	61
6.3.3 MF-FIPWA 3-D O2I-SM translator	62
6.3.4 Integration differences between FIPWA and MF-FIPWA	62
6.3.5 Summary of integration differences	65
6.4 Construction Method for O2I-SM Translator	65
6.4.1 Scaling with problem size	65
6.4.2 Efficient construction method	66
6.5 Numerical Results	69
6.5.1 Accuracy of spectral Green's function expansion	70
6.5.2 Accuracy of solution	72
6.5.3 Efficiency in setup and matrix solver	75
6.5.4 Mixed-form algorithm	79
6.6 Conclusions on MF-FIPWA Versus FIPWA	80

CHAPTER 7 IMPLEMENTATION AND OPTIMIZATION	81
7.1 Introduction	81
7.2 Architecture of the Fast Algorithm	82
7.3 Data Structures	86
7.3.1 Translation matrices	86
7.3.2 Error control modules	88
7.3.3 Proposed translator class for the FMA library	88
7.4 Numerical Methods	90
7.4.1 Numerical integration	90
7.4.2 Interpolation and extrapolation	97
7.5 Error Control Methodology	100
7.5.1 Global error control	101
7.5.2 Local error control	102
7.6 Applied Error Control in MF-FIPWA	103
7.6.1 Error control according to bandwidth	103
7.6.2 Error control according to the inner SDP	104
7.6.3 Error control according to the outer SDP Γ_β	105
7.7 Summary	105
CHAPTER 8 TESTING AND DEBUGGING	106
8.1 Introduction	106
8.2 Technical Aspects	106
8.2.1 Code compatibility	106
8.2.2 Code limitations	108
8.3 Debugging Methods	109
8.3.1 Comparison by Green's function	109
8.3.2 Comparison by plane wave direction	113
8.4 Summary	115
CHAPTER 9 NONUNIFORM SAMPLING OF θ-ϕ PLANE	116
9.1 Introduction	116
9.2 Memory and CPU Costs of the Fast Algorithm	117
9.3 Uniform and Nonuniform Sampling	120
9.3.1 Uniform sampling	120
9.3.2 Nonuniform sampling	120
9.4 FIPWANOS Versus FIPWA	121
9.4.1 Memory cost of FIPWANOS for smallest box size of $\lambda/10$	122
9.4.2 Memory cost of FIPWANOS for smallest box size of $\lambda/5$	123
9.5 Summary	126
CHAPTER 10 CONCLUSIONS	127
10.1 Conclusions	127
10.2 Future Work	128

APPENDIX A SPECIAL INTERPOLATION FUNCTION.....	129
APPENDIX B COMPARATIVE RESULTS OF SCALING STUDY.....	132
REFERENCES.....	143
AUTHOR'S BIOGRAPHY.....	147

LIST OF FIGURES

FIGURE	PAGE
2.1 Scattering of impenetrable object by incident electric and magnetic fields. . .	6
2.2 Total field due to direct and reflected fields.	8
2.3 Bounding box and subdivision.	9
2.4 Steepest descent path (SDP).	12
2.5 Instability caused by SDPs that cross SAPs.	13
2.6 Modified steepest descent path (M-SDP).	14
2.7 Multilevel subdivision of cubes.	15
3.1 Steepest descent paths for 2-D translator.	20
3.2 Particle separation and group association.	21
3.3 Quadratic mapping from the complex s -plane to the complex α -plane. . . .	22
3.4 Family of steepest descent paths in lossy media.	23
3.5 Family of steepest descent paths in active media.	23
3.6 SDP, SAP, CMP for lossy media.	25
3.7 SDP, SAP, CMP for active media.	25
3.8 Group parameters used to define the M-SDP.	26
3.9 Normalized integrand of Eq. (3.5) for small groups in lossy medium.	27
3.10 Normalized integrand of Eq. (3.5) for small groups in active medium.	27
3.11 Normalized integrand of Eq. (3.5) for large groups in lossy medium.	28
3.12 Normalized integrand of Eq. (3.5) for large groups in active medium.	28
4.1 Interpolation error of small group in lossy medium with large separation. . .	34
4.2 Interpolation error of small group in active medium with large separation. . .	34
4.3 Extrapolation from real axis to MSDP in lossy media.	36
4.4 Extrapolation from real axis to MSDP in active media.	36
4.5 Error control of small groups by group separation in lossy media.	37
4.6 Control regions for small boxes.	38
4.7 Control regions for large boxes.	39
5.1 Illustration of how k_ρ depends on $k \sin \theta$	48
5.2 Illustration of the shallow and deep evanescent regions of paths I and III. . .	49
5.3 Validation scattering case.	52
5.4 Comparison of MF-FIPWA with FIPWA.	52
5.5 Error of FIPWA and MF-FIPWA relative to the full matrix solution.	53

5.6	Scaling of memory and processing time for matrix-vector product.	54
5.7	VFY-218 above lossy earth, $\epsilon = (6.5, 0.6)$	55
5.8	Bistatic radar cross section of VFY-218 above lossy earth.	55
6.1	The 3-D translator is illustrated as a sampled function on the unit sphere. .	60
6.2	Integration paths used to construct O2I-layer translator.	67
6.3	Comparison of bistatic solutions when smallest box size is $\lambda/10$	73
6.4	Comparison of bistatic solutions when smallest box size is $\lambda/5$	73
6.5	Relative error between scattering solutions of MF-FIPWA and FIPWA. . . .	74
6.6	Scaling of global relative error.	75
6.7	Setup time for O2I-SM translation matrix by box size.	79
6.8	Comparison of memory cost when the smallest box size is $\lambda/10$	80
7.1	Top level diagram of MF-FIPWA under the MLFMA paradigm.	83
7.2	Setting up the O2I-layer translation matrix for FIPWA and MF-FIPWA. . .	84
7.3	Proposed translator class for FMA library.	89
7.4	Distribution of quadrature points from quadratic map.	97
8.1	Debugging the stability problem.	114
B.1	Bistatic scattering and relative error of sphere at 0.103 GHz.	133
B.2	Bistatic scattering and relative error of sphere at 0.297 GHz.	134
B.3	Bistatic scattering and relative error of sphere at 0.943 GHz.	135
B.4	Bistatic scattering and relative error of sphere at 1.481 GHz.	136
B.5	Bistatic scattering and relative error of sphere at 2.099 GHz.	137
B.6	Bistatic scattering and relative error of sphere at 2.959 GHz.	138
B.7	Bistatic scattering and relative error of sphere at 0.206 GHz.	139
B.8	Bistatic scattering and relative error of sphere at 0.594 GHz.	140
B.9	Bistatic scattering and relative error of sphere at 1.887 GHz.	141
B.10	Bistatic scattering and relative error of sphere at 2.963 GHz.	142

LIST OF TABLES

TABLE	PAGE
3.1 Definition of loss/gain ratio $\tan \delta$	20
6.1 List of scattering examples and set up parameters.	70
6.2 RMS error of O2I-SM translator by box size.	71
6.3 MF-FIPWA construction cost of O2I-SM matrices when $a = \lambda/10$	76
6.4 FIPWA construction cost of O2I-SM matrices when $a = \lambda/10$	76
6.5 MF-FIPWA construction cost of O2I-SM matrices when $a = \lambda/5$	77
6.6 FIPWA construction cost of O2I-SM matrices when $a = \lambda/5$	78
7.1 Gauss-Laguerre quadrature with 15 points.	92
7.2 Integration error of large boxes in 2-D FIPWA for $\tan \delta = 0$	94
7.3 Integration error of large boxes in 2-D FIPWA for $\tan \delta = \frac{\pi}{6}$	94
7.4 Integration error of large boxes in 2-D FIPWA for $\tan \delta = -\frac{\pi}{6}$	94
7.5 Integration error of small boxes in 2-D FIPWA for $\tan \delta = 0$	96
7.6 Integration error of small boxes in 2-D FIPWA for $\tan \delta = \frac{\pi}{6}$	96
7.7 Integration error of small boxes in 2-D FIPWA for $\tan \delta = -\frac{\pi}{6}$	96
9.1 Distribution of memory for FIPWA.	118
9.2 Distribution of computation time for FIPWA.	119
9.3 Cost of uniform sampling for smallest box size of $\lambda/10$	122
9.4 Cost of nonuniform sampling for smallest box size of $\lambda/10$	123
9.5 Percentage of reduction for smallest box size of $\lambda/10$	124
9.6 Cost of uniform sampling for smallest box size of $\lambda/5$	125
9.7 Cost of nonuniform sampling for smallest box size of $\lambda/5$	125
9.8 Percentage of reduction for smallest box size of $\lambda/5$	126

MULTILEVEL MULTIPOLE-FREE FAST ALGORITHM FOR ELECTROMAGNETIC SCATTERING PROBLEMS IN LAYERED MEDIA

Michael Andrew Saville, Ph.D.
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, 2006
Weng Cho Chew, Advisor

A multilevel multipole-free algorithm is presented for solving electromagnetic scattering problems in the vicinity of a half space or layered medium. By replacing the multipole expansion in the fast inhomogeneous plane wave algorithm (FIPWA) with a multipole-free expansion, this new algorithm is simpler to derive and retains $\mathcal{O}(N \log N)$ scaling in memory and processing time. To develop this new algorithm, known as the multipole-free fast inhomogeneous plane wave algorithm (MF-FIPWA), error control is established for arbitrary accuracy.

In addition, comparison of the memory usage and simulation time is presented for FIPWA and MF-FIPWA for moderate to large scale problems. Various alternate approaches to implementing MF-FIPWA are discussed in terms of how the fast algorithms set up translation matrices and where gains can be made. Finally, details of the advantages of using non-uniform sampling are provided. Results show 30% savings in memory usage and up to 20% savings in computing the matrix-vector product.

CHAPTER 1

INTRODUCTION

1.1 Background

Electromagnetics applications play a vital role in the world today. The importance of this field ranges from the common convenience of wireless phones and networks, to the critical need of global communication and defense systems. Key to the design of these technologies is simulation of electromagnetic wave propagation, radiation, and scattering.

Maxwell's equations, the governing equations for electromagnetic phenomena, are elegantly concise, but are, by no means, simple to solve. While the concise form¹ traditionally includes only four vector differential equations, most real-world problems are not solvable without the aid of computer science. To solve real problems, the differential equations are cast into a large number of linear equations and computed simultaneously. It might appear that the problem is easily solved; however, advanced mathematics and computational skills are often needed to complete the simulation.

While there are mature mathematical and numerical techniques for solving differential equations, the underlying physics introduce additional challenges. In many instances, the frequency of interest makes the size of the problem impractical to solve, even with a computer. For example, to evaluate the radiation pattern of an installed aircraft antenna at X-band, the number of matrix equations easily exceeds 10 million. Whether differential or integral

¹Credited to Oliver Heaviside [1].

equation solvers are used, the computation requires an enormous amount of computer memory and number of floating point operations. Additionally, many advancing technologies and problems have even more sophisticated needs. Of particular interest to national defense are the buried object and foliage penetration problems, whereby one desires to use electromagnetic sensing to observe, monitor, and track a target that is embedded in the earth or masked by forest. These problems are difficult to solve because they involve complicated environments and immense sizes of over 100 million unknowns.

In recent years, the effort of solving large problems has been eased by the invention of fast algorithms [2–8]. These are varied, but are recognizable by the efficiency $\mathcal{O}(N \log N)$ with which they solve the electromagnetics problem. Fast algorithms have been successfully applied to both differential and integral equation solvers, but the integral equation method is well-suited for many scattering and remote sensing applications. The number of unknowns can be greatly reduced as compared to the differential solvers, and no artificial boundary is needed to truncate the problem domain. However, the integral equation has its own difficulty, and that is the need for a Green’s function. Once the Green’s function is known, or can be computed, the problem can be efficiently solved with the fast algorithm.

1.2 Fast Algorithms

The notion of a fast algorithm is implied by the computational complexity, or efficiency, of the algorithm, where *fast* is anything better than $\mathcal{O}(N^2)$. In terms of the integral equation solver, the method of moments (MOM) [9] is routinely employed to solve electromagnetic (EM) scattering and radiation problems. In this method, the scatterer, or radiator, is first discretized into a finite set of radiating elements. The Green’s function describes how each element radiates EM fields due to an unknown excitation current. In solving for the unknown current elements, each current element must interact with every other. For a very large problem, this entails a matrix of enormous size and a capable matrix solver. Using iterative

matrix solvers, such as the conjugate gradient method, up to two matrix-vector products must be computed during each iteration at a cost of $\mathcal{O}(N^2)$. This is still prohibitive to large-scale problems, and this is where the fast algorithms contribute. In particular, the multilevel fast multipole algorithm (MLFMA) [8] and the fast inhomogeneous plane wave algorithm (FIPWA) [10] both solve the integral equation formulation by accelerating the matrix-vector products in the matrix solver. In this work, the phrase *multilevel fast algorithm* is used to describe the broader class of fast algorithms that use a multilevel approach [11] to achieve $\mathcal{O}(N \log N)$ complexity. The multilevel approach will be discussed more in Chapter 2.

Today, different categories exist for fast algorithms that are based on integral equations, but each is formulated by expanding the Green's function with multipoles, plane-waves, or both [8, 10, 12], and then forming a diagonal Green's-function operator. Essentially, the expansion is achieved by replacing the spatial, free-space Green's function with a spectral representation. Instead of computing the numerous interactions individually, aggregates are computed by first treating neighboring elements as a group, which has a radiation pattern. The radiation pattern is represented in the spectral (plane-wave) domain so that it can be translated to a receiving group with a limited number of plane waves. This limited number equates to a diagonal form of the Green's function. Finally, the receiving group disseminates the incoming radiated field to its respective elements. In this process, only a single translation is needed for many interactions, and it is computed efficiently because of the diagonal form.

The particular expansion of the Green's function is what makes each fast algorithm unique. Most solutions for free space problems use the fast multipole algorithm. Yet, several solutions to the low-frequency, free-space scattering problem use purely plane-wave based approaches. For example, in [13], the Green's function is formed as a mix of propagating and evanescent plane waves. These approaches achieve $\mathcal{O}(N \log N)$ cost in processing and memory because they use the multilevel paradigm. However, the methods for layered-media problems still rely on multipole expansions [14–16]. The Green's function for layered media

is complicated, because its closed form solution is in integral form, i.e., Sommerfeld integrals. The monopole integrand of the Sommerfeld integral is expanded into multipoles to yield a diagonal-form Green's function. In the formulation of FIPWA for the layered media problem, the Sommerfeld integrals are accelerated with the method of steepest descent [1], and the Green's function is set into diagonal form with interpolation and extrapolation. While FIPWA used multipoles, the multipoles are not needed and can be replaced with the plane-wave expansion of 2-D FIPWA. The details are reserved for Chapter 2.

1.3 Multipole-Free Fast Algorithm

The focus of this research is to develop a pure plane-wave, or multipole-free, fast algorithm for layered media applications. By replacing the Green's function expansion, or translator, of the original FIPWA with a multipole-free expansion, a simpler translator is achieved. The significance of the multipole-free expansion is first seen in the simplicity of the derivation and computation. Second, the use of plane waves makes it easier, and potentially less expensive, to control the error due to low-frequency breakdown. Finally, this new algorithm is studied for suitability to large scale problems, such as scattering of a military vehicle over lossy earth.

1.4 Organization

Chapters 2–3 present the ground work of the new algorithm. Chapter 4 demonstrates error control of the 2-D translator that is used in the new algorithm and Chapter 5 demonstrates the multipole-free algorithm for canonical and complex targets. Chapter 6 presents a comparison of the multipole and multipole-free forms of FIPWA. Chapter 7 discusses the implementation and optimization. Chapter 8 provides notes on debugging and testing the new algorithm. Chapter 9 initiates a study of non-uniform sampling of the translator coordinates, and Chapter 10 presents brief conclusions.

CHAPTER 2

FAST INTEGRAL EQUATION SOLVER

2.1 Electric Field Integral Equation

Given the differential, time-harmonic form of Maxwell's equations, it is straightforward to derive the vector wave equation,

$$\nabla \times \nabla \times \mathbf{E}(\mathbf{r}) - \omega^2 \mu \epsilon \mathbf{E}(\mathbf{r}) = i\omega \mu \mathbf{J}(\mathbf{r}), \quad (2.1)$$

where $\mathbf{J}$ represents the electric sources and $\mathbf{E}$ represents the unknown electric field. Note that the time convention $e^{-i\omega t}$ is used and suppressed. The free space scattering by a perfect electric conductor (PEC), shown in Fig. 2.1, can be solved by computing the electric field in (2.1) with the electric-field, surface integral equation [17],

$$\frac{4\pi i}{k\eta} \hat{\mathbf{n}} \times \mathbf{E}^{inc}(\mathbf{r}) = \hat{\mathbf{n}} \times \int_S dS \overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}'), \quad \mathbf{r} \in S_0, \quad (2.2)$$

where $\overline{\mathbf{G}} = (\bar{\mathbf{I}} - \frac{1}{k^2} \nabla \nabla')$ $g_0(\mathbf{r}, \mathbf{r}')$ is the free-space dyadic Green's function, and $g_0(\mathbf{r}, \mathbf{r}') = \frac{e^{i\mathbf{k} \cdot (\mathbf{r} - \mathbf{r}')}}{|\mathbf{r} - \mathbf{r}'|}$. The currents on the target are unknown, but can be solved with the Moment Method [9], also called the Method of Moments (MOM).

For the surface integral equation in MOM, $\mathbf{J}(\mathbf{r})$ is routinely expanded with the Rao, Wilton, Glisson (RWG) basis functions [18]. Next, the boundary condition, $\hat{\mathbf{n}} \times \mathbf{E}^{inc} = \hat{\mathbf{n}} \times \mathbf{E}^{scat}$, $\mathbf{r}$ on the surface S_0 , is enforced with a set of testing functions to form the matrix

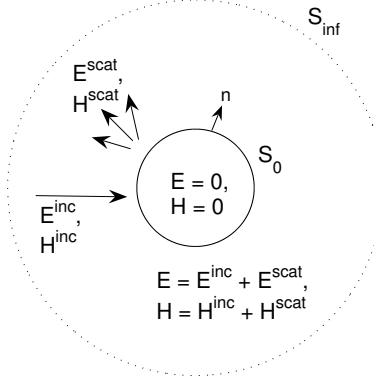


Figure 2.1: Scattering of impenetrable object by incident electric and magnetic fields.

equation

$$\mathbf{V} = \bar{\mathbf{Z}} \cdot \mathbf{I}. \quad (2.3)$$

In (2.3), $\mathbf{V} = \langle \mathbf{\Lambda}(\mathbf{r}), \hat{\mathbf{n}} \times \mathbf{E}^{inc}(\mathbf{r}) \rangle$, $\langle \mathbf{\Lambda}(\mathbf{r}), \bar{\mathbf{G}}(\mathbf{r}, \mathbf{r}'), \mathbf{\Lambda}(\mathbf{r}') \rangle$, and $\mathbf{\Lambda}$ is the RWG basis/testing function. The integrals are represented with the inner product notation $\langle \cdot, \cdot \rangle$.

Clearly, the current vector $\mathbf{I}$ is solved by inverting $\bar{\mathbf{Z}}$. However, this is extremely expensive, with a cost of $\mathcal{O}(N^3)$. Hence, iterative solvers are used, but they are also too expensive for large problems. In general, the fast algorithms accelerate the matrix-vector product in the iterative solver. The particular fashion of constructing the matrix-vector product also alleviates the need to form $\bar{\mathbf{Z}}$ explicitly. The multilevel fast algorithms achieve a solution for $\mathbf{I}$ with $\mathcal{O}(N \log N)$ cost in processing time and $\mathcal{O}(N \log N)$ cost in memory.

2.2 Layered Media Dyadic Green's Function

The Green's function for the layered-media problem is more costly to compute than the free-space Green's function because Sommerfeld integrals are involved. To find the layered-media dyadic Green's function, the spatial Green's function is evaluated in the spectral domain.

Using the E_z, H_z formulation in [1], the spectral Green's function for planarly layered media can be derived as

$$g(\mathbf{r}, \mathbf{r}') = \frac{i}{2} \int_{SIP} dk_\rho \frac{k_\rho}{k_{1z}} H_0^{(1)}(k_\rho \rho) \left[e^{ik_{1z}|z|} + \tilde{R}(k_{1z}, \dots, k_{Nz}) e^{ik_{1z}(z+2d_1)} \right], \quad (2.4)$$

where $\tilde{R}(k_{1z}, \dots, k_{Nz})$ is the generalized reflection coefficient for the N -layer medium, $k_{iz} = \sqrt{k_i^2 - k_\rho^2}$, d_1 is the height of the interface, and SIP is the Sommerfeld integration path.

While the above is not the Image Theorem, the contribution of the reflection is treated as though the source comes from an image, and is weighted by the appropriate reflection coefficient. Figure 2.2 shows the source and image paths, and clarifies the purpose of the generalized reflection coefficient. $\tilde{R}$ is evaluated by recursively computing the Fresnel reflection coefficients, and thus constitutes the reflections from N planar, complex homogeneous layers.

Letting $d_1 = 0, z > 0$, and generalizing the coefficients in the integrand, (2.4) is expressed more succinctly as

$$\begin{aligned} g(\mathbf{r}, \mathbf{r}') &= g^d(\mathbf{r}, \mathbf{r}') + g^r(\mathbf{r}, \mathbf{r}'), \\ g^d(\mathbf{r}, \mathbf{r}') &= \int_{SIP} dk_\rho W^d(k_{1z}) H_0^{(1)}(k_\rho \rho) e^{ik_{1z}z}, \\ g^r(\mathbf{r}, \mathbf{r}') &= \begin{cases} \int_{SIP} dk_\rho W^{TM}(k_{1z}, \dots, k_{Nz}) H_0^{(1)}(k_\rho \rho) e^{ik_{1z}z}, \\ \int_{SIP} dk_\rho W^{TE}(k_{1z}, \dots, k_{Nz}) H_0^{(1)}(k_\rho \rho) e^{ik_{1z}z}, \end{cases} \end{aligned} \quad (2.5)$$

where $W^d(k_{1z}) = \frac{ik_\rho}{2k_{1z}}$, $W^{TM}(k_{1z}, \dots, k_{Nz}) = \frac{ik_\rho}{2k_{1z}} \tilde{R}^{TM}$, and $W^{TE}(k_{1z}, \dots, k_{Nz}) = -\frac{ik_\rho}{2k_{1z}} \tilde{R}^{TE}$.

The dyadic Green's function $\overline{\mathbf{G}}$ is symmetric [19], and each element in $\overline{\mathbf{G}}$ reduces to a combination of the components in (2.5). Hence, the individual components are not shown here. The next step is to apply the fast algorithm to accelerate computation of the matrix elements of (2.3), as well as the matrix-vector products.

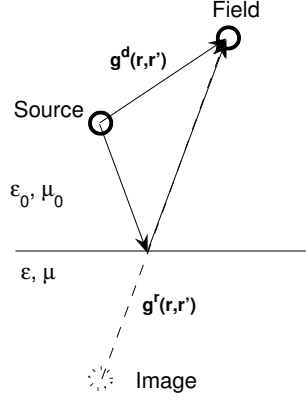


Figure 2.2: Total field due to direct and reflected fields.

2.3 FIPWA for Layered Media

2.3.1 Why FIPWA?

The fast inhomogeneous plane wave algorithm, FIPWA, is chosen as the fast algorithm for this research because it is based on a plane-wave approach [20], is adaptable to a multipole-free form, and is known to solve various layered-media scattering problems, such as scattering by an object over a lossy half-space, and scattering by a buried object [10, 14, 21]. The original formulation expanded part of the factorized Green's function with multipoles and is presented here to lay the groundwork for the modification in Chapter 3. Emphasis is given to the multilevel approach, as it is important to understanding efficiency in error control, as discussed in Chapter 5.

2.3.2 Factorizing the Green's function

Factorization begins by first discretizing the scatterer into N elements, and then grouping basis elements, or particles, according to proximity, as shown in Fig. 2.3. The top left displays

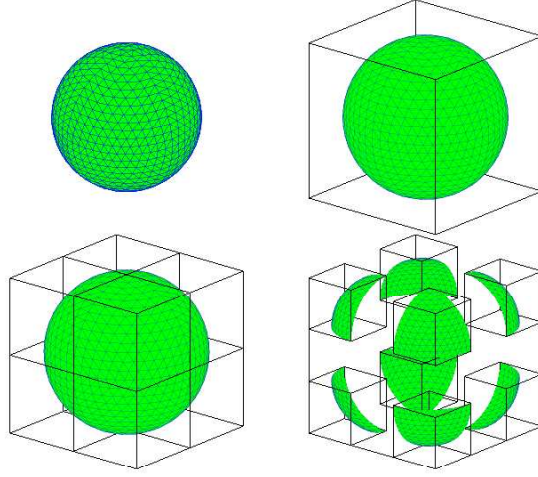


Figure 2.3: Bounding box and subdivision. Top left: A meshed sphere. Top right: The bounding cube around the meshed sphere. Bottom left: First partition into eight subcubes (level 1). Bottom right: An expanded view of the level 1 cubes with their respective particles.

a simple sphere; the top right shows the bounding box. Starting from the bounding box, each box is recursively divided into eight smaller boxes until the desired number of levels is reached. For clarity, the bottom left and right show the first level subdivision, and how the particles are grouped according to their parent cube. In the multilevel approach, many levels are used, with hierarchy representing an upside-down tree. Hence, each box is associated with a different level in the tree. This will be discussed more in a later section, but for now, it begins to show how particles will be grouped together to form aggregate radiation patterns. As the Green's function is factorized, the radiation patterns will be distinguishable from the translator. Recalling that the matrix elements in (2.3) represent interactions between particles, the separation vector $\mathbf{r}_{ji}$ denotes a particular interaction pair. For the reflected interaction, $\mathbf{r}_{ji} = \hat{\mathbf{x}}(x_j - x_i) + \hat{\mathbf{y}}(y_j - y_i) + \hat{\mathbf{z}}(z_j + z_i)$. The vector is factored according to the parent cubes so that $\mathbf{r}_{ji} = \mathbf{r}_{jJ} + \mathbf{r}_{JI} + \mathbf{r}_{Ii}$. Typically, the Green's function is factored into three parts: radiation pattern, translator, receiving pattern. It is succinctly written as

$$g(r_{ji}) = \sum_{\Omega_s} \beta_{jJ}(\Omega_s) \cdot \mathcal{T}_{JI}(\Omega_s) \cdot \beta_{Ii}(\Omega_s), \quad (2.6)$$

where β_{Ii} represents the radiation pattern from cube I that contains particle i , $\mathcal{T}_{JI}$ represents the translation of the radiation pattern of cube I to far-field cube J , and β_{jJ} represents the receiving pattern and dissemination to a particle j in cube J .

The factored form in FIPWA is derived from (2.5) by changing variables from (k_ρ, k_z) to (θ, α) , and introducing the 2-D translator from the fast multipole algorithm (FMA) [17]. FMA is well documented in the references, so it is not derived here. Instead, the factored form of FIPWA is presented, where the FMA translator is used in the 3-D translator. Also, the term for the direct wave interaction can be expressed without using multipoles. Therefore, the following derivation is for the reflected wave part of the Green's function:

$$g^q(\mathbf{r}_j, \mathbf{r}_i) = \int_{SIP} d\theta W^q(\theta) H_0^{(1)}(k \sin \theta |\boldsymbol{\rho}_j - \boldsymbol{\rho}_i|) e^{ik \cos \theta (z_j + z_i)}, \quad (2.7)$$

$$= \int_{SIP} d\theta W^q(\theta) e^{ikz_{jJ} \cos \theta} \left[H_0^{(1)}(k \sin \theta |\boldsymbol{\rho}_j - \boldsymbol{\rho}_i|) e^{ikz_{JI} \cos \theta} \right] e^{ikz_{Ii} \cos \theta}, \quad (2.8)$$

$$\begin{aligned} &= \int_{SIP} d\theta W^q(\theta) e^{ikz_{jJ} \cos \theta} e^{ikz_{Ii} \cos \theta} \\ &\quad \times \frac{1}{2\pi} \int_0^{2\pi} d\alpha \sum_{p=-P}^P H_p^{(1)}(k \sin \theta \rho_{JI}) e^{-ip(\alpha - \phi_{JI} - \pi/2)} e^{ikz_{JI} \cos \theta} \\ &\quad \times e^{ik\rho_{jJ} \sin \theta \cos(\alpha - \phi_{jJ})} e^{ik\rho_{Ii} \sin \theta \cos(\alpha - \phi_{Ii})} \end{aligned} \quad (2.9)$$

$$\begin{aligned} &= \int_{SIP} d\theta \int_0^{2\pi} d\alpha e^{i\mathbf{k}(\theta, \alpha) \cdot \mathbf{r}_{jJ}} \\ &\quad \times \left(\frac{1}{2\pi} W^q(\theta) e^{ikz_{JI} \cos \theta} \mathcal{T}_{JI}(\theta, \alpha) \right) e^{i\mathbf{k}(\theta, \alpha) \cdot \mathbf{r}_{Ii}}, \end{aligned} \quad (2.10)$$

where q represents the TE, or TM component, and the multipole expansion is given by

$$\mathcal{T}_{JI}(\theta, \alpha) = \sum_{p=-P}^P H_p^{(1)}(k \sin \theta \rho_{JI}) e^{-ip(\alpha - \phi_{JI} - \pi/2)}.$$

For the direct interaction between two particles, (2.10) is exact when $P \rightarrow \infty$, but the error has been shown to be exponentially controllable when P is finite. However, to make the factorization efficient, the radiation and receiving patterns, represented by the exponential terms containing Ii and jJ , are sampled. These represent radiation and receiving patterns which are smooth in the far field, and efficiency comes from using the same samples for all

cubes of the same size. The sampled patterns are

$$e^{i\mathbf{k}(\theta,\alpha)\cdot\mathbf{r}_{jJ}}e^{i\mathbf{k}(\theta,\alpha)\cdot\mathbf{r}_{Ii}} = \sum_{\Omega_s} e^{i\mathbf{k}(\theta_s,\alpha_s)\cdot\mathbf{r}_{jJ}}e^{i\mathbf{k}(\theta_s,\alpha_s)\cdot\mathbf{r}_{Ii}}I(\theta_s,\alpha_s,\Omega_s), \quad (2.11)$$

where $I(\theta_s,\alpha_s,\Omega_s) = I_\theta(\theta - \theta_s)I_\alpha(\alpha - \alpha_s)$ represents the interpolation function from the sample set $\Omega_s = (\theta_s,\alpha_s)$.

Upon substituting the interpolation into (2.10), and swapping the order of summation and integration, the Green's function becomes factored as

$$\begin{aligned} g^q(\mathbf{r}_j, \mathbf{r}_i) &= \sum_{\Omega_s} e^{i\mathbf{k}(\theta_s,\alpha_s)\cdot\mathbf{r}_{jJ}}e^{i\mathbf{k}(\theta_s,\alpha_s)\cdot\mathbf{r}_{Ii}} \\ &\quad \times \left(\int_{SIP} d\theta \frac{1}{2\pi} W^q(\theta) e^{ikz_{JI} \cos \theta} \int_0^{2\pi} d\alpha \mathcal{T}_{JI}(\theta, \alpha) I(\theta_s, \alpha_s, \Omega_s) \right) \\ &= \sum_{\Omega_s} \beta_{jJ}(\Omega_s) \cdot \mathcal{T}_{JI}(\Omega_s) \cdot \beta_{Ii}(\Omega_s), \end{aligned} \quad (2.12)$$

where $\beta_{jJ}(\Omega_s) = e^{i\mathbf{k}(\theta_s,\alpha_s)\cdot\mathbf{r}_{jJ}}$, $\beta_{Ii}(\Omega_s) = e^{i\mathbf{k}(\theta_s,\alpha_s)\cdot\mathbf{r}_{Ii}}$ and the form of the 3-D FIPWA translator for layered media is

$$\mathcal{T}_{JI}(\Omega_s) = \int_{SIP} d\theta \int_0^{2\pi} d\alpha \frac{1}{2\pi} W^q(\theta) e^{ikz_{JI} \cos \theta} \mathcal{T}_{JI}(\theta, \alpha) I(\theta_s, \alpha_s, \Omega_s). \quad (2.13)$$

2.3.3 Steepest descent path

The 3D translator described by (2.13) is slow to compute, is very unstable, and is a dense operator. The slowness comes from the Sommerfeld integration path (SIP). The integrand is oscillatory on the SIP in the θ -plane, so the path of integration is very long. Instead of integrating on the SIP, the Cauchy Theorem and Jordan's Lemma are invoked to deform the SIP to the steepest descent path (SDP). Starting from the saddle point of the integrand in (2.13) ($\theta = 0$), the integrand has constant phase, i.e., it is not oscillatory, and it has exponential decay. It converges very fast.

Figure 2.4 shows the SDP for a single interaction. Also shown are the steepest ascent path (SAP) and constant magnitude path (CMP), which will help clarify the source of instability.

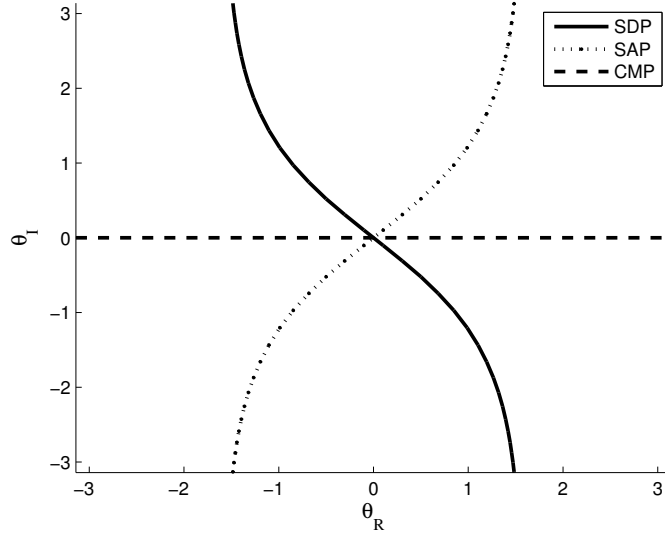


Figure 2.4: Steepest descent path (SDP). Also shown are the corresponding steepest ascent (SAP) and constant magnitude (CMP) paths.

A rigorous treatment of the steepest descent path is deferred to Chapter 3, where it is derived for complex media. Here, the SDP is presented aside from its explicit formula. In [21,22], this SDP is called the original SDP, but in this effort, it is called the fundamental SDP to distinguish it more clearly from the modified SDP that is presented next. Also, the saddle point for the fundamental SDP is referred to as the fundamental saddle point.

2.3.4 Modified SDP

The translator is used to translate all particles in cube I to cube J . As such, it must account for all possible interactions between particles in cubes I and J . When the fundamental path is used for all interactions (i.e., translation between the cube centers), it is possible that one SDP crosses the SAP of another. Figure 2.5 shows how this occurs and illustrates how to construct the modified SDP (M-SDP). The saddle point for any SDP occurs where the path crosses the real axis. This makes it easy to construct a single M-SDP that will prevent crossing a steepest ascent path.

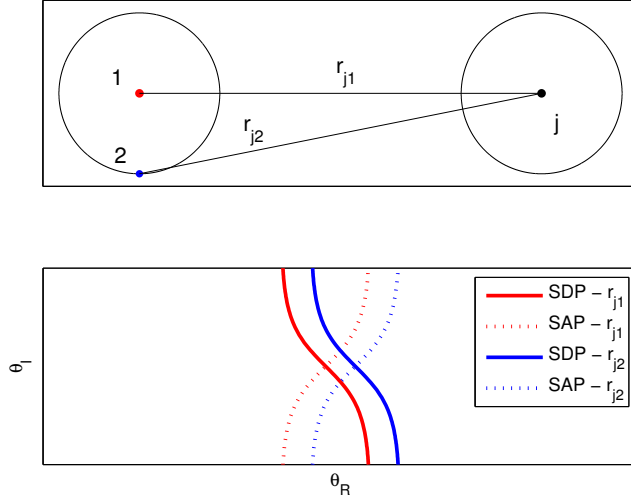


Figure 2.5: Instability caused by SDPs that cross SAPs. Top: two different interactions with particle j . Bottom: SDP and SAP for each interaction. The SDPs cross each other's SAP, causing instability in the numerical integration along the SDP.

Referring to the M-SDP, shown in Fig. 2.6, each section of the M-SDP is labeled as Path I, Path II, and Path III. This is not the exact SDP, so some error occurs. However, the error is controllable with proper numerical integration. Gauss-Laguerre rules are used on Paths I and III, while Gauss-Legendre rules are used on Path II. The length of Paths I and III are discussed in Chapter 4.

With the M-SDP, the 3-D FIPWA translator for layered media is

$$\mathcal{T}_{JI,3\text{DFIPWA}}(\Omega_s) = \int_{\Gamma_\theta} d\theta \int_0^{2\pi} d\alpha \frac{1}{2\pi} W^q(\theta) e^{ikz_{JI} \cos \theta} \mathcal{T}_{JI}(\theta, \alpha) I(\theta_s, \alpha_s, \Omega_s), \quad (2.14)$$

where Γ_θ and Γ_α are the respective, modified steepest descent paths.

The last point in Section 2.3.3, about the translator being dense, is also overcome by the M-SDP, because only a single integration path is used for all the particle interactions between cubes I and J . By using a single path of integration, only one set of samples are also needed. The samples $\theta_s \in [-\pi, \pi]$ and $\alpha_s \in [-\pi, \pi]$ represent N_s samples on the unit sphere. In this context, the radiation and receiving patterns form vectors of length N_s , and the translator in (2.14) is a diagonal matrix. This is the desired form for a fast algorithm.

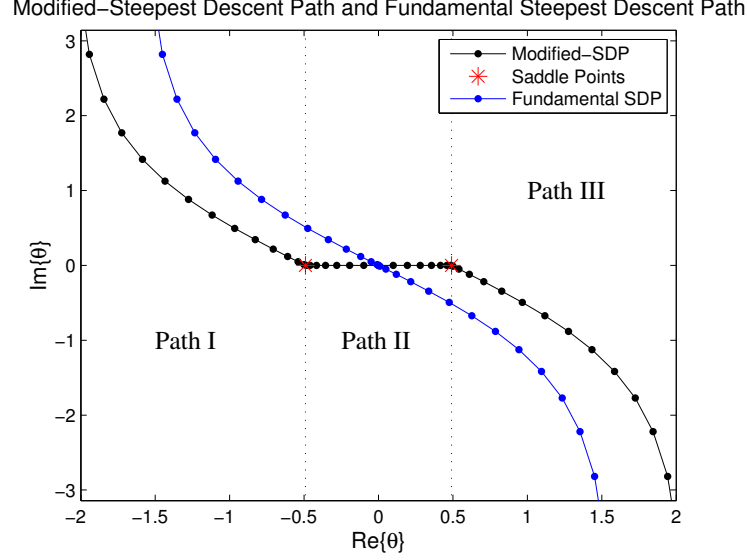


Figure 2.6: Modified steepest descent path (M-SDP). Saddle points are located where the SDP intersects the real axis. Hence, Paths I and III of the SDP are translated far enough from the fundamental saddle point so that they do not cross any SAPs. Path II is along the real axis between Paths I and III.

Finally, the interpolation functions were introduced before the SIP was deformed to the M-SDP. For θ on Path I or Path III, $I(\theta - \theta_s)$ functions as an extrapolation function. This is allowed because the radiation patterns are analytic in the complex plane, and analytically continuous. In this work, the term extrapolation will be used for both interpolation and extrapolation of the M-SDP to distinguish from the interpolation of the multilevel approach. Also, the fundamental SDP is not used, so the modified SDP will simply be referred to as the SDP.

2.3.5 Multilevel implementation

The 2-D multilevel implementation is represented in Fig. 2.7, where four levels are shown, and different levels are assigned to different box sizes. The level 0 box is the bounding box that encompasses the problem domain. The level 1 box is the first box division, and level 2 is the first level where boxes are separated far enough to use FIPWA. Hence, the multilevel approach translates child-level radiation patterns to parent-level patterns until the highest

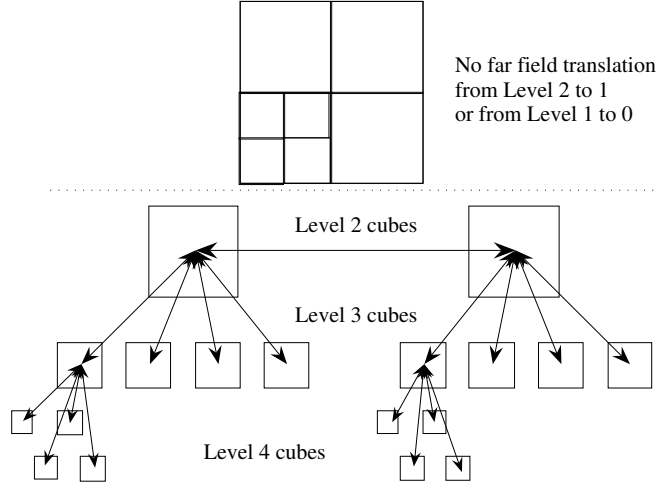


Figure 2.7: Multilevel subdivision of cubes. Shown is the 2-D representation. There are no translations from levels 2 to 1 or 1 to 0.

level in the tree is reached. This enables an efficient matrix-vector product, however, the children cannot be summed naïvely. The 3-D subdivision of the cube into eight smaller cubes was illustrated in Fig. 2.3, page 9. The number of samples used for each cube size is different because the bandwidth of the corresponding radiation pattern varies with the cube size. Thus, the patterns of smaller cubes are interpolated to larger cubes, and when traversing downward, the larger cubes are antinterpolated (transpose interpolation) to smaller cubes. The mathematical description in vector notation is [17]

$$g^q(\mathbf{r}_j, \mathbf{r}_i) = \boldsymbol{\beta}_{j_{l_{max}}}^t \cdot \bar{\mathbf{I}}_1 \cdot \bar{\boldsymbol{\beta}}_{J_{l_{max}} J_{l_{max}+1}} \cdot \bar{\mathbf{I}}_2 \cdots \bar{\boldsymbol{\beta}}_{J_{L+1} J_L} \cdot \mathcal{T}_{J_L I_L} \cdot \bar{\boldsymbol{\beta}}_{I_L I_{L+1}} \cdots \bar{\mathbf{I}}_2 \cdot \bar{\boldsymbol{\beta}}_{I_{l_{max}+1} I_{l_{max}}} \cdot \bar{\mathbf{I}}_1 \cdot \boldsymbol{\beta}_{I_{l_{max}} i}, \quad (2.15)$$

where $\bar{\boldsymbol{\beta}}_{uv}$ are the child to parent translations, $\bar{\boldsymbol{\beta}}_{uv}^t$ are the parent to child translations, L is the highest translation level, typically set to 2, l_{max} is the level of the smallest cubes, and $\bar{\mathbf{I}}_n, n = 1, 2, \dots$ are the interpolation matrices. Note that the transposes serve as the antinterpolation matrices.

The approximation, inherent in interpolation, introduces exponentially controllable error. This error is different from the error caused by interpolation and extrapolation to the SDP.

2.4 Summary

In this chapter, the scattering from a PEC object is presented in the form of the electric field integral equation and MOM. Next, the dyadic Green's function is shown explicitly for the layered media case. Finally, the overview of FIPWA establishes a baseline for the multipole-free modification that is applied in Chapter 3.

CHAPTER 3

MODIFIED FORM OF FIPWA FOR LAYERED MEDIA

3.1 Introduction

Hu et al. developed the fast inhomogeneous plane wave algorithm (FIPWA) [10, 14, 22, 23] as an alternative to the fast multipole algorithm (FMA) and multilevel fast multipole algorithm (MLFMA) (thoroughly discussed and cited in [17]). FIPWA evolves from the fast steepest descent path algorithm (FASDPA) developed by Michielssen and Chew [24], which was the first fast algorithm ($\mathcal{O}(N^{4/3} \ln N)$) to use the spectral integral representation of the Green's function, i.e., propagating and evanescent plane waves. FIPWA modified the steepest descent path (SDP) integrals in FASDPA with extrapolation of evanescent waves from propagating waves to achieve a diagonal translator. In doing so, FIPWA achieved $\mathcal{O}(N \log N)$ efficiency for various two- and three-dimensional scattering problems.

Recently, FIPWA has been adapted in various works to solve the well-known, low-frequency breakdown in free-space problems [13, 25–27]. Greengard et al. [25] adopted the plane-wave methods to overcome the low frequency breakdown. They used the spectral representation of the Green's function, and then computed the propagating and evanescent waves with different sets of inhomogeneous plane waves. FIPWA has also permeated other techniques that include both inhomogeneous plane waves and modified multipoles [13, 26, 27].

Sarvas and Wallén [27] provide an excellent summary of the various techniques along with their composition of the different methods.

To modify 3-D FIPWA for layered media into a purely plane-wave-based algorithm, the 2-D FMA translator is replaced with the 2-D FIPWA translator [22,23]. However, the original work on 2-D FIPWA was for free-space problems. In the substitution, the 2-D FIPWA entails complex media, which means additional SDPs must be carefully defined. With the modification, potential computational savings become immediately clear.

3.2 Formulation of the Plane-Wave Algorithm

The spectral Green's function for layered media consists of two principal parts: the direct interaction and the reflected interaction. The former has already been solved with free space FIPWA, and the latter can be derived as [1]

$$g^r(\mathbf{r}, \mathbf{r}') = \int_{\Gamma} d\theta W(k \sin \theta) e^{ikz \cos \theta} H_0^{(1)}(k\rho \sin \theta), \quad W(k_\rho) = \frac{i}{2} R^{\text{TM,TE}}(k_{1z}, k_{2z}), \quad (3.1)$$

where $k_{iz} = \sqrt{k_i^2 - k_{i\rho}^2}$. While the notion of a purely plane-wave driven, fast algorithm was first proposed in the fast steepest descent path algorithm [24], FIPWA was originally factored under the MLFMA paradigm, with the 2-D FMA translator in place of the 2-D Green's function. Using the basic formulation of 2-D FIPWA and letting $\mathbf{k}_\rho = k \sin \theta [\hat{\mathbf{x}} \cos(\alpha) + \hat{\mathbf{y}} \sin(\alpha)]$, and $\boldsymbol{\rho}_{ji} = \boldsymbol{\rho}_{jJ} + \boldsymbol{\rho}_{JI} + \boldsymbol{\rho}_{Ii}$, the 2-D translator is derived as

$$\begin{aligned} H_0^{(1)}(k_\rho \rho_{ji}) &= \frac{1}{\pi} \int_{\Gamma_\alpha} d\alpha e^{i\mathbf{k}_\rho(\alpha) \cdot \boldsymbol{\rho}_{jJ}} \cdot e^{i\mathbf{k}_\rho(\alpha) \cdot \boldsymbol{\rho}_{JI}} \cdot e^{i\mathbf{k}_\rho(\alpha) \cdot \boldsymbol{\rho}_{Ii}} \\ &= \sum_{s=1}^{N_s} e^{i\mathbf{k}_\rho(\alpha_s) \cdot \boldsymbol{\rho}_{jJ}} \cdot T(\alpha_s) \cdot e^{i\mathbf{k}_\rho(\alpha_s) \cdot \boldsymbol{\rho}_{Ii}}, \end{aligned}$$

where α_s are the sample points, and $T(\alpha_s)$ is either the FIPWA or FMA translator. The potential cost savings can be seen by comparing the 2D FMA and FIPWA translators.

$$T_{2\text{dfipwa}}(\alpha_s) = \frac{1}{\pi} \int_{\Gamma_\alpha} d\alpha e^{i\mathbf{k}_\rho(\alpha) \cdot \boldsymbol{\rho}_{JI}} I(\alpha - \alpha_s), \quad (3.2)$$

$$T_{2\text{dfma}}(\alpha_s) = \frac{1}{2\pi} \int_0^{2\pi} d\alpha \left(\sum_{p=-N_p}^{N_p} H_p^{(1)}(k_\rho \rho_{JI}) e^{-ip(\alpha_{JI} - \alpha - \pi/2)} \right) I(\alpha - \alpha_s), \quad (3.3)$$

where $I(\alpha - \alpha_s)$ is the interpolation or extrapolation function, $N_p = (kD) + 1.8d_0^{2/3}(kD)^{1/3}$, and d_0 is the number of digits of precision [17]. It is important to note that the interpolation function in (3.2) may be a global or local interpolation function. If it is global, then it must use all of the stored samples, which is computationally expensive. In practice, it is local, where only a few stored samples are used and the cost is greatly reduced.

From (3.1) it can be seen that as long as θ is on the complex SDP, $k_\rho = k \sin \theta$ is complex and can take on all values in the complex plane. This complicates the SDP integration in the α -plane. Figure 3.1 shows how the SDPs change as k_ρ changes quadrants in the spectral plane. The circle in the center of the figure represents the angular locus of possible values of $\hat{\mathbf{k}}_\rho$, and the subgraphs shown in each quadrant of the k_ρ -plane show the corresponding SDP and steepest ascent path (SAP).

Upon substituting (3.2) into (3.1), the plane wave translator for layered media is

$$\mathcal{T}_{JI,3\text{DMFIPWA}}(\Omega_s) = \int_{\Gamma_\theta} d\theta \int_{\Gamma_\alpha} d\alpha W^q(\theta) e^{ikz_{JI} \cos \theta} e^{i\mathbf{k}_\rho(\alpha) \cdot \boldsymbol{\rho}_{JI}} I(\theta_s, \alpha_s, \Omega_s), \quad (3.4)$$

where MF-FIPWA distinguishes the multipole-free, fast inhomogeneous plane-wave algorithm (MF-FIPWA) from the fast inhomogeneous plane-wave algorithm reviewed in Chapter 2.

Knowledge of the location of the SAP is critical to performing accurate numerical integration. However, before deriving the SDP for complex media, it is important to note that $\mathbf{k}_\rho$ varies in magnitude according to the value of θ . Essentially, each 2-D translator can have a different effective frequency. This point will be addressed in Section 5.2 of Chapter 5.

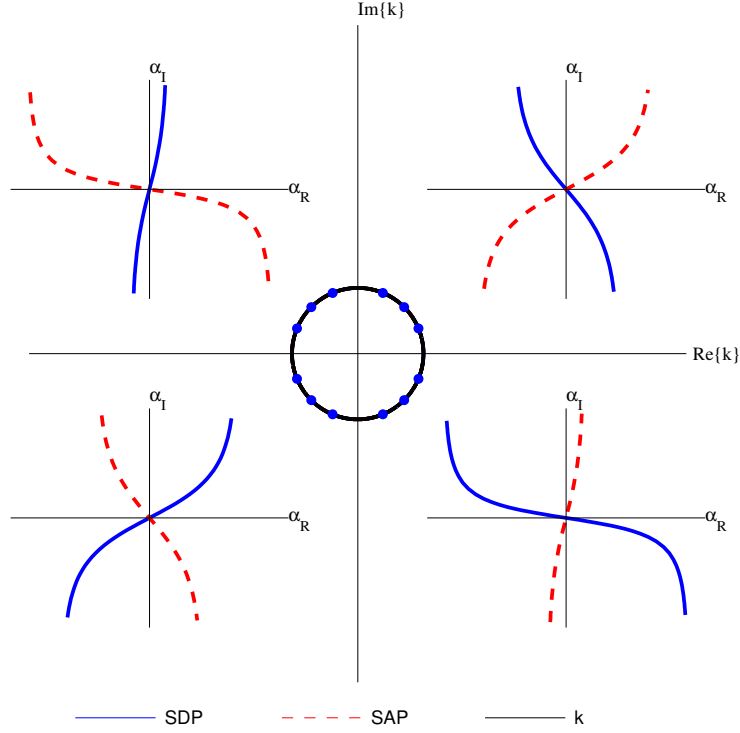


Figure 3.1: Steepest descent paths for 2-D translator.

3.3 Deriving the Steepest Descent Path

The SDP for a lossless background medium is a special case of the complex case, therefore the following is applicable to the SDPs in free space problems as well. The fundamental SDP is derived first, followed by the construction of the modified SDP. To simplify the 2-D notation, let $k = ke^{i\delta}$, where $\tan \delta = k_I/k_R \in [-1, 1]$. The last constraint is due to the continuity relations for the Bessel functions. Finally, the lossless, lossy, and active medium backgrounds are denoted according to Table 3.1.

Table 3.1: Definition of loss/gain ratio $\tan \delta$.

$\tan \delta$	Type of Medium
< 0	Active
$= 0$	Lossless
> 0	Lossy

The integral representation of the 2-D Green's function is given as [22, 23]

$$\begin{aligned} H_0^{(1)}(k\rho_{ji}) &= \frac{1}{\pi} \int_{\Gamma} d\alpha e^{i\mathbf{k}(\alpha) \cdot \boldsymbol{\rho}_{ji}}, \\ &= \frac{1}{\pi} \int_{\Gamma} d\alpha e^{i\mathbf{k}(\alpha) \cdot \boldsymbol{\rho}_{jJ}} e^{i\mathbf{k}(\alpha) \cdot \boldsymbol{\rho}_{JI}} e^{i\mathbf{k}(\alpha) \cdot \boldsymbol{\rho}_{Ii}}, \end{aligned} \quad (3.5)$$

where $\mathbf{k}(\alpha) = k(\hat{\mathbf{x}} \sin \alpha + \hat{\mathbf{y}} \cos \alpha)$, Γ is the steepest descent path in the complex α -plane, and the subscripts denote the group and particle orientations as shown in Fig. 3.2.

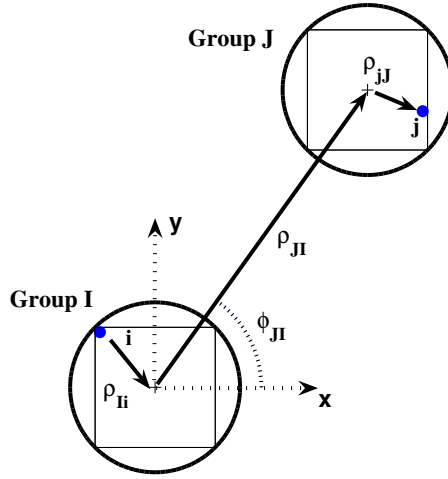


Figure 3.2: Particle separation and group association.

The fundamental SDP is derived from the translator segment, $e^{ik\rho_{JI} \cos(\alpha - \phi_{JI})}$, where ϕ_{JI} is defined in Fig. 3.2. Following the technique of [1], the fundamental SDP is found by expanding the exponent into a Taylor series and mapping α to the real s -axis as

$$ik \cos(\alpha - \phi_{JI}) = -s^2, \quad -\infty < s < \infty, \quad (3.6)$$

where $\alpha = \alpha_R + i\alpha_I$, and ϕ_{JI} is the angle $\boldsymbol{\rho}_{JI}$ makes with the x -axis. Thus, points on Γ are defined by

$$\alpha = \text{sgn}(s) \cos^{-1}(1 - s^2/ik) + \phi_{JI}. \quad (3.7)$$

Figure 3.3 displays points chosen in the s -plane and the corresponding points in the α -plane according to the quadratic mapping in (3.7).

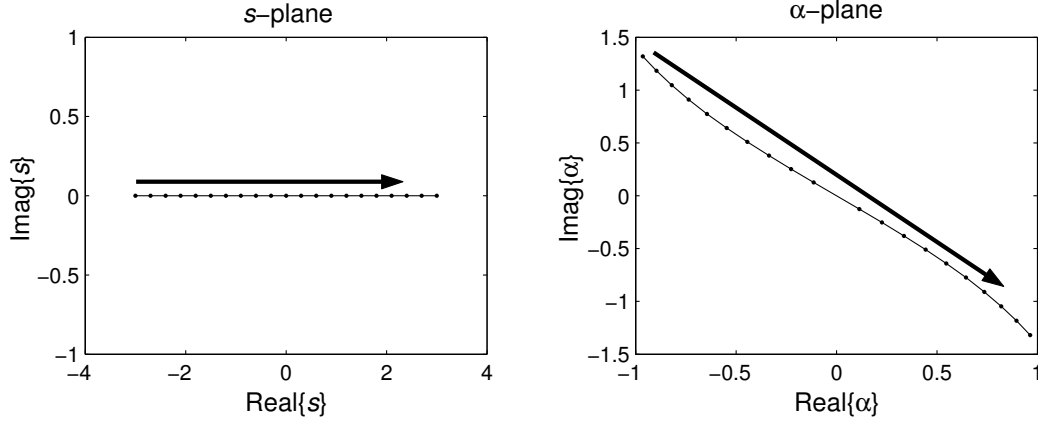


Figure 3.3: Quadratic mapping from the complex s -plane to the complex α -plane. Shown for (3.6) with $\tan \delta = 0.1$. Left: real values of mapping parameter in complex s -plane. Right: SDP in complex α -plane.

3.3.1 SDPs in lossy medium

Several SDPs for lossy medium are shown in Fig. 3.4. The solid black line denotes the SDP for lossless media. The SDPs revolve about the saddle point, centered at $\alpha = \phi_{JI}$, in the way a pinwheel revolves about its center pin. The curves rotate clockwise as $\tan \delta \rightarrow 1$. Under extreme loss (Fig. 3.4), the SDP becomes aligned with the imaginary α -axis. The red line shows the nearly vertical SDP that also passes through the saddle point. The saddle point is located at the origin of the α -plane. Under lossless conditions, the asymptotes of the SDP approach the lines $\alpha = \pm\pi/2$. However, in lossy media the asymptotes occur before $\alpha = |\pi/2|$. The dashed, vertical black line in Fig. 3.4 shows the asymptotes of the SDP when $\tan \delta \approx 0$.

3.3.2 SDPs in active medium

The SDP for the active medium case is shown in Fig. 3.5. It behaves similarly to the SDP for lossy media, but the revolution is counter-clockwise. As $\tan \delta \rightarrow -1$, the SDP becomes horizontally aligned with the real α -axis. In Chapter 4, this behavior will be shown to benefit FIPWA. Also, the vertical asymptotes occur past the lines $\alpha = \pm\pi/2$.

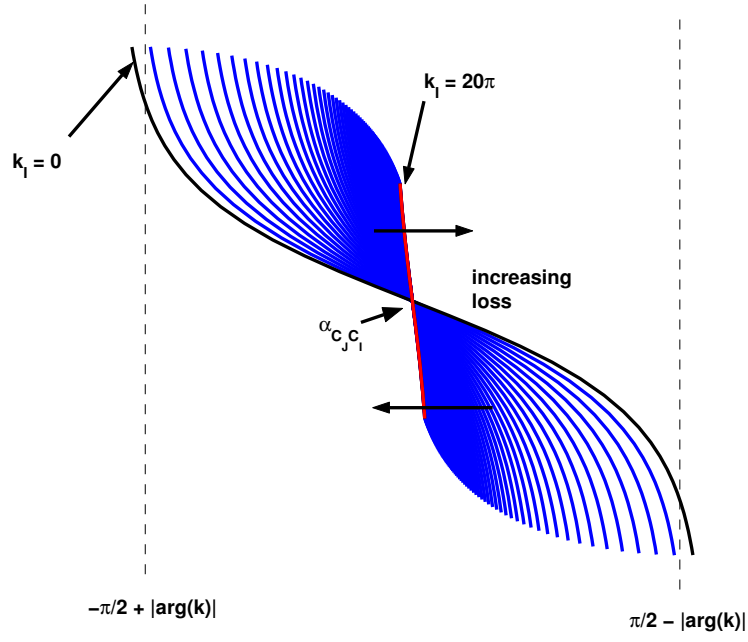


Figure 3.4: Family of steepest descent paths in lossy media.

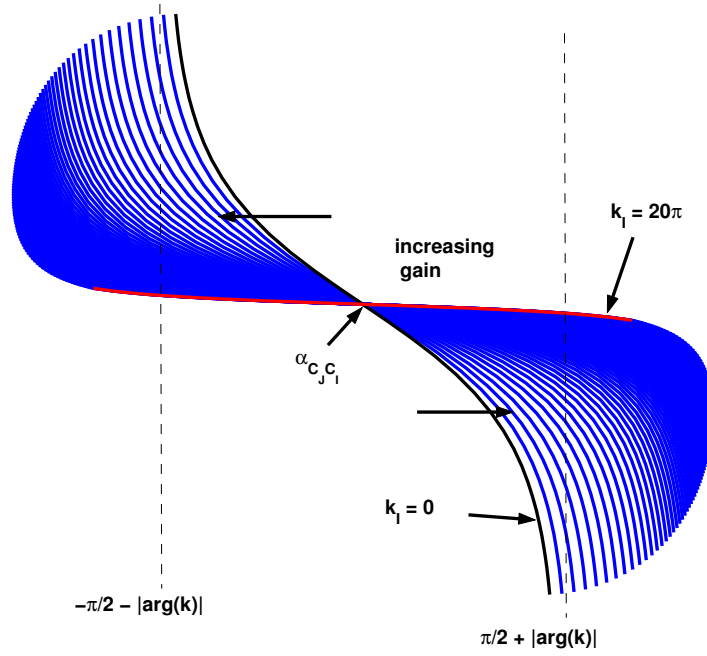


Figure 3.5: Family of steepest descent paths in active media.

3.3.3 Steepest ascent and constant magnitude paths

A more complete perspective of the complex α -plane is shown in Figs. 3.6 and 3.7. In these graphs, the SDP is shown for various loss/gain ratios along with the corresponding steepest ascent path (SAP) and constant magnitude path (CMP). The SAP and CMP also rotate in accordance with $\tan \delta$, and the modified steepest descent path (M-SDP) can still be constructed because the SAP, SDP, and CMP all pass through the saddle point. However, for the lossy medium case, the extrapolation degrades. The reasons why will be discussed in Chapter 4.

3.3.4 Modified steepest descent path (M-SDP)

As discussed in [21], the SDP is infinite and unique to the orientation of particles j and i . The M-SDP is used to designate the single contour of integration for all particle interactions between groups I and J . Referring to Fig. 2.6, page 14, an example M-SDP with the fundamental SDP is shown for lossless media; the M-SDPs in lossy and active media are constructed in similar fashion, but with consideration of the physical orientation of sending and receiving groups.

In Fig. 3.8, page 26, the real angle ϕ_0 is defined as $\phi_0 = \sin^{-1}(|\mathbf{k} \cdot \hat{D}D|/|\mathbf{k} \cdot \hat{L}L|) = \sin^{-1}(kD/kL)$, where D is the group size, and L is the group separation. Clearly, Path II changes length according to group size and separation.

To construct the M-SDP, Paths I and III are shifted by ϕ_0 . The complete SDP for arbitrary group size, separation, loss/gain, and truncation point is

$$\begin{aligned} \text{Path I: } \alpha &= -\cos^{-1}(1 - s^2/ik) - |\phi_0| + \phi_{JI}, \quad s < 0, \\ \text{Path II: } \alpha_{JI} - |\phi_0| &< \alpha_R < \alpha_{JI} + |\phi_0|, \\ \text{Path III: } \alpha &= \cos^{-1}(1 - s^2/ik) + |\phi_0| + \phi_{JI}, \quad s > 0, \end{aligned} \tag{3.8}$$

where $\phi_0 = \sin^{-1}(kD/kL)$, and s is on the real s -axis.

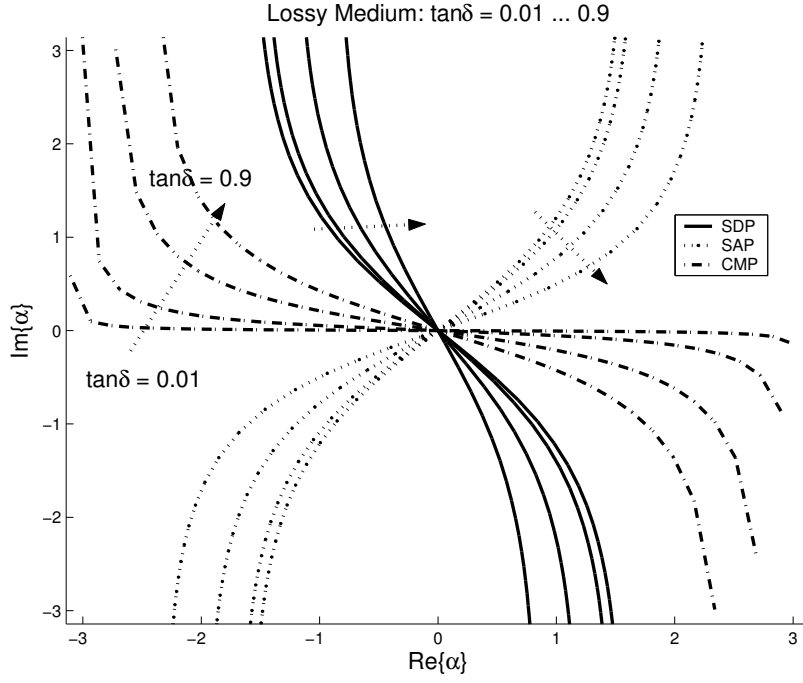


Figure 3.6: SDP, SAP, CMP for lossy media. Arrows indicate direction of increasing loss.

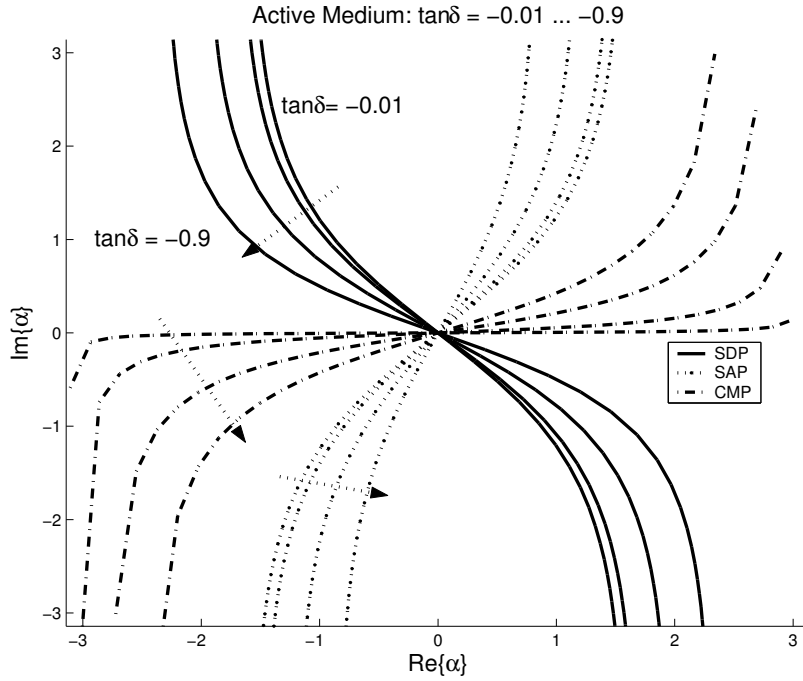


Figure 3.7: SDP, SAP, CMP for active media. Arrows indicate direction of increasing gain.

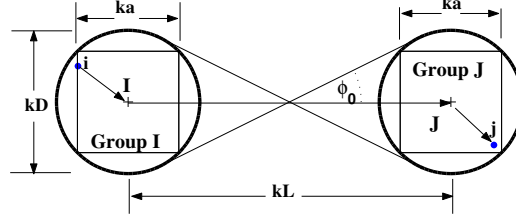


Figure 3.8: Group parameters used to define the M-SDP.

3.3.5 Behavior of the integrand along the M-SDP

Examples of the behavior of the integrand along the M-SDP are shown in Figs. 3.9–3.12, pages 27–28. The top graph in Fig. 3.9 shows the M-SDP in lossless (black line) and lossy media (colored lines) with group size $kD = 5$ and separation $kL = 10$ (one buffer box). The middle graphs show the normalized real and imaginary parts of the integrand along the M-SDP, and the bottom graph shows the normalized magnitude of the integrand. Note that the horizontal axis is the real part of α for alignment with the M-SDP in the top graph. Figure 3.10 shows a similar comparison for active media. The integrand of (3.5) decays exponentially fast, so paths I and III do not extend far from the real axis.

This is seen more clearly with similar comparisons for large groups. Figures 3.11 and 3.12 show the lossy and active cases, respectively, for group size $kD = 15$ and one buffer box separation. As seen by the M-SDP, the imaginary part of α , α_I , is less than 0.5, whereas, in the case of the small group with a single buffer box, $\alpha_I \approx 1$ (Fig. 3.9). In the active case (Fig. 3.12), paths I and III only extend 1-2 points away from the real axis.

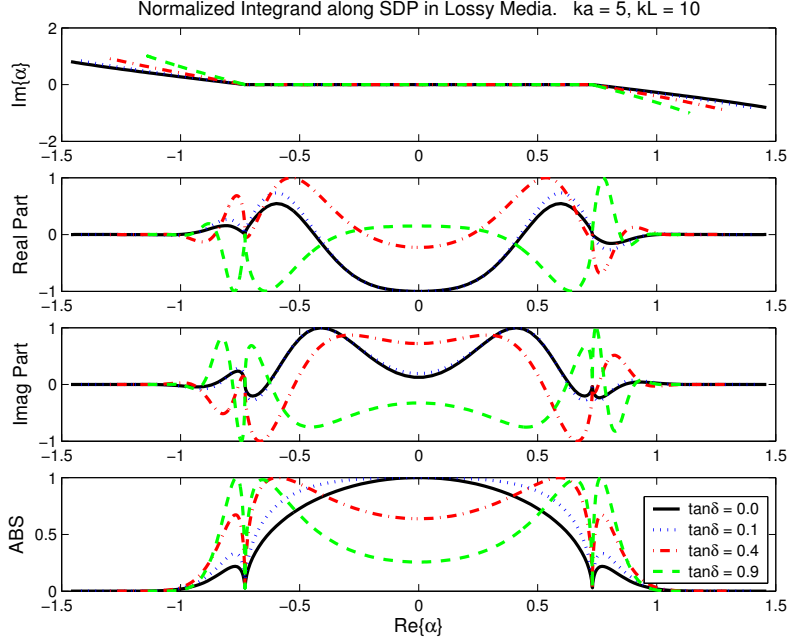


Figure 3.9: Normalized integrand of (3.5) along M-SDP for small groups in lossy medium. Shown for $kD = 5$ and small separation $kL = 10$ (one buffer box) in a lossy medium. Top: M-SDP. Middle: Real and imaginary parts. Bottom: Magnitude.

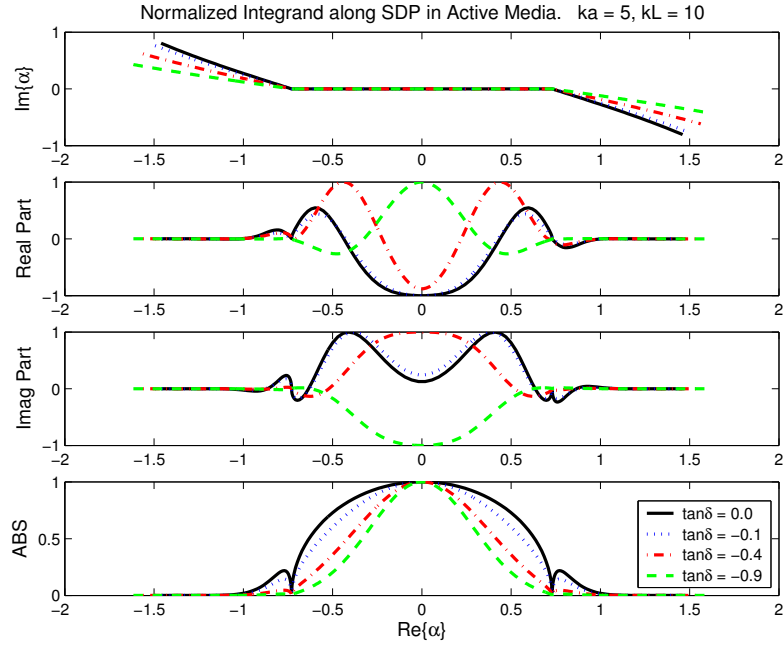


Figure 3.10: Normalized integrand of (3.5) along M-SDP for small groups in active medium. Shown for $kD = 5$ and one buffer box separation ($kL = 10$) in an active medium. Top: M-SDP. Middle: Real and imaginary parts. Bottom: Magnitude.

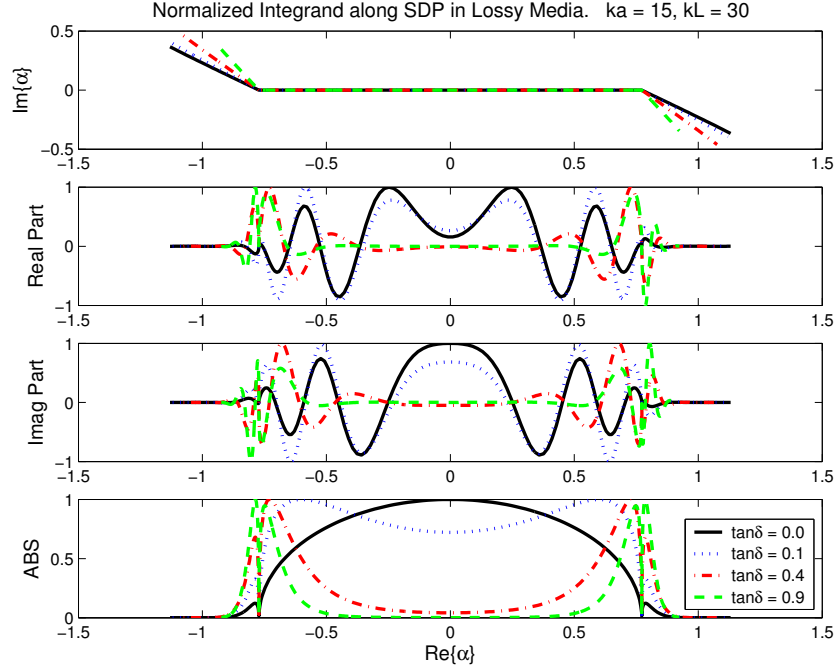


Figure 3.11: Normalized integrand of (3.5) along M-SDP for large groups in lossy medium. Shown for $kD = 15$ and small separation $kL = 30$ (one buffer box) in a lossy medium. Top: M-SDP. Middle: Real and imaginary parts. Bottom: Magnitude.

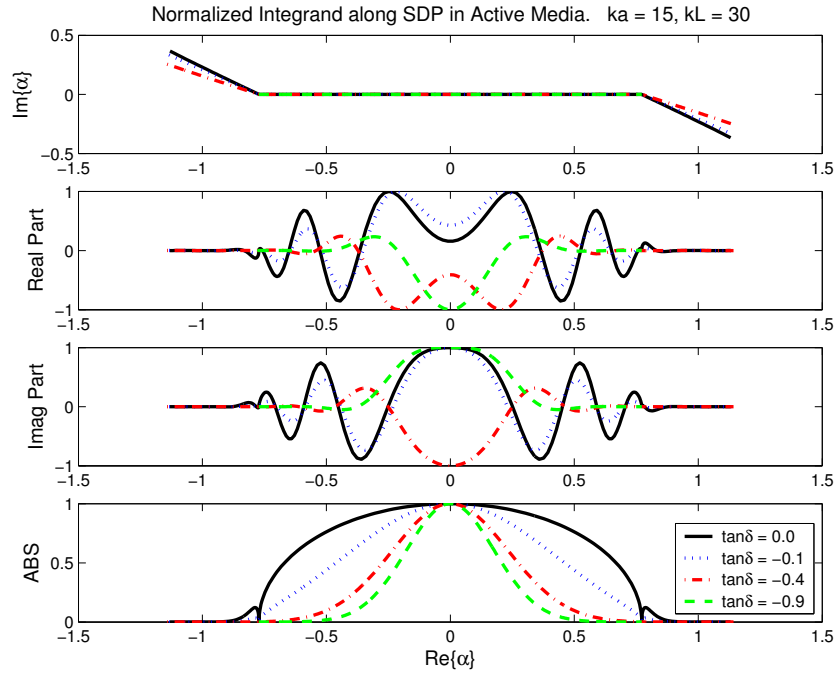


Figure 3.12: Normalized integrand of (3.5) along M-SDP for large groups in active medium. Shown for $kD = 15$ and one buffer box separation ($kL = 30$) in an active medium. Top: M-SDP. Middle: Real and imaginary parts. Bottom: Magnitude.

3.4 Cost Analysis

The plane-wave algorithm is easier to derive than the multipole algorithms, and in this section the cost is briefly compared to FIPWA. Given the 3-D translator $\mathcal{T}_{JI,3D}(\theta_s, \alpha_s) = W(\theta)e^{ikz_{JI} \cos \theta_s} \mathcal{T}_{JI,2D}(\theta_s, \alpha_s)$, where $\mathcal{T}_{JI,2D}(\theta_s, \alpha_s)$ can be the 2-D FMA or 2-D FIPWA translator. Upon comparing the two translators,

$$\mathcal{T}_{JI,2D,\text{fma}}(\theta_s, \alpha_s) = \frac{1}{2\pi} \int_0^{2\pi} d\alpha \sum_{p=-P}^P H_p^{(1)}(k \sin \theta_s \rho_{JI}) e^{-ip(\alpha - \phi_{JI} - \pi/2)} I(\alpha - \alpha_s), \quad (3.9)$$

$$\mathcal{T}_{JI,2D,\text{fipwa}}(\theta_s, \alpha_s) = \frac{1}{\pi} \int_{\Gamma_\phi} d\alpha e^{ik\rho_{JI} \sin \theta_s \cos(\alpha - \phi_{JI})} I(\alpha - \alpha_s), \quad (3.10)$$

it is clear to see that it is simpler and less expensive to construct $\mathcal{T}_{JI,3D}$ with $\mathcal{T}_{JI,\text{fipwa}}$. Upon discretizing the integrals,

$$\mathcal{T}_{JI,2D,\text{fma}}(\theta_s, \alpha_s) = \sum_q^{N_Q} \sum_{p=-P}^P w_q H_p^{(1)}(k \sin \theta_s \rho_{JI}) e^{-ip(\alpha_q - \phi_{JI} - \pi/2)} I(\alpha_q - \alpha_s), \quad (3.11)$$

and

$$\mathcal{T}_{JI,2D,\text{fipwa}}(\theta_s, \alpha_s) = \sum_q^{N_Q} w_q e^{ik\rho_{JI} \sin \theta_s \cos(\alpha_q - \phi_{JI})} I(\alpha_q - \alpha_s), \quad (3.12)$$

where N_Q is the number of quadrature points, and w_q is the quadrature weight.

For the multipole translator, P is derived from the refined excess bandwidth formula [17], so the integration is proportional to P . The bandwidth of the radiation pattern is $2P$, and following the Nyquist sampling theorem, $N_\alpha = 4P$. Also, the number of samples for interlevel interpolation is $N_s = 2P$, making the total complexity of constructing the 2-D multipole translator $N_{\text{fma}} = N_s N_Q (2P + 1) \approx (2P)^3$.

The multipole-free translator has $N_Q = N_I + N_{II} + N_{III}$, where the subscripts, I, II, III , denote the integration path in Fig. 2.6. Letting the interpolation function, $I(\cdot)$, equal the Dirichlet function as defined in [21], $I(\cdot)$ has a finite bandwidth of $2P$ [21]. The number of samples for interlevel interpolation is $N_s = 2P$, as it is the same as for the FMA translator. The total complexity for the 2-D FIPWA translator is $N_{\text{fipwa}} = N_s N_Q$.

N_I, N_{II}, N_{III} have not been specified, but because N_{II} is on the real axis, it represents the number of propagating waves, and is equal to P for $\alpha \in [-\pi/4, \pi/4]$. As for N_I and N_{III} , the exact relationship between the bandwidth of $e^{ik\rho_{JI}\cos(\alpha-\phi_{JI})}$ and the Gauss-Laguerre quadrature rule has not been established. In [21], $N_I = N_{III} = 15$ for the 3-D SDP (θ). However, to compare the 2-D translators fairly, N_I, N_{III} must be set high enough to ensure double machine precision accuracy. In this work, $N_I = N_{III} = 40$, but the threshold of equal cost is $N_{\text{fipwa}} = N_{\text{fma}}$. This occurs when

$$N_{\text{fipwa}} \leq N_{\text{fma}} \quad (3.13)$$

$$N_s N_Q = N_s N_Q N_P \quad (3.14)$$

$$(2P)(P + 2N_I) = (2P)(2P)(2P + 1) \quad (3.15)$$

$$N_I = 2P^2 + 0.5P \quad (3.16)$$

The last equation, (3.16) allows one to compare the cost to construct the 2-D FIPWA translator in terms of the multipole expansion number, P . As long as the number of quadrature points on paths I and III of the 2-D FIPWA translator can be kept low, the cost to construct the 2-D FIPWA translator is less than the cost to construct the 2-D FMA translator. A more detailed discussion of the total cost to construct the 3-D FIPWA and 3-D multipole-free translator is presented in Chapter 6.

3.5 Summary

In this chapter, the translator for FIPWA in layered media was modified to a purely plane-wave based translator. While simpler to derive, the added SDP complicates the translator because the 2-D translator must allow complex materials and broadband behavior. The SDP was derived and various paths were shown to illustrate differences between the lossless, lossy, and active paths. Finally, these differences revealed how the error control must adapt to the changing values of k_ρ . The next chapter discusses how to control the error.

CHAPTER 4

ERROR CONTROL

4.1 Introduction

To extend the plane-wave method to the layered-media problem, the 2-D FIPWA should be stable and error-controllable in complex media. The primary error sources are the truncation of the steepest descent path, the extrapolation error, and the numerical machine error. To establish a suitable control, this section follows the investigation in [28, 29], where the error for the 2-D FIPWA was studied for free-space cases. In [28] and [29], the excess bandwidth formula, as used in MLFMA [17, 30], was shown to be the dominant control mechanism for extrapolation. Also, the authors derived the interdependence between the number of sample points used for extrapolation and the truncation of the SDP. In cases where the accuracy could not be controlled, the authors optimized the controlling parameters: the number of sampling points and the length of the SDP.

A key element of FIPWA is extrapolation of the radiation patterns from real-valued sample points to the SDP. While the nature of the SDP in lossless media makes this quite suitable, the SDP in complex media does not exhibit the same behavior, resulting in less-controllable error. Therefore, additional effort must be made to control the associated errors in complex media before 2-D FIWPA can be applied to the 3-D layered-media problem.

In this chapter, complex media are categorized as lossy, lossless, or active. Furthermore, for each case, control regions are defined according to the loss, gain, and number of buffer

boxes, as used in FIPWA and MLFMA. It is shown that 2-D FIPWA is partially error-controllable for lossy media and highly error-controllable for active media. The results lead to the conclusion that 2-D FIPWA is stable, and error controllable, in the Sommerfeld integrals for layered-media problems.

4.2 SDP Truncation Error

The lengths of Paths I and III are set according to the desired accuracy. Fortunately, truncation error by itself is controllable to machine precision. Given the desired digits of precision, d_0 , the M-SDP truncation point, $\alpha = \alpha_{Rt} + i\alpha_{It}$, can be determined from

$$e^{\Im m\{(kD+kL)\cos(\alpha_{Rt}+i\alpha_{It})\}} = 10^{-d_0}, \quad (4.1)$$

which is slightly modified from [29] to account for complex k , and where D and L are the groups' diameters and separations, respectively. An alternative approach is to use the quadratic map $e^{-s^2\rho_{JI}} \leq 10^{-d_0}$. As long as the integrand decays exponentially along Paths I and III, $s = s_{max}$ is easily determined, and the error is highly controllable. Hence, the main effort is to enable highly accurate extrapolation.

4.3 Extrapolation Error

Hu et al. [22] studied interpolation and extrapolation with various functions: sinc, approximate prolate spheroidal (APS), and the truncated-APS. The truncated APS is a local interpolation method (with respect to the set of samples), and loses 1-2 digits of accuracy as a trade-off to gaining efficiency. However, Ohnuki and Chew [29] found it useful to study the error control with the sinc interpolation function because it is global and results in the least error. Hence, the sinc function is used in this section. Although the sinc function is used for both interpolation and extrapolation, the extrapolation contributes the largest error. With

the sinc function, the extrapolation error is partially controllable according to the refined excess bandwidth formula [17]

$$P = kD + 1.8d_0^{\frac{2}{3}}(kD)^{\frac{1}{3}}, \quad (4.2)$$

where P is the number of samples in $[-\pi, \pi]$, and d_0 is the number of digits of accuracy.

Ohnuki and Chew [29] defined intervals of P where the error is controlled with (4.2) for lossless media. Figures 4.1 and 4.2 show examples of the regions defined in [29], but for lossy and active media, respectively. Region I is uncontrollable and occurs when too few samples are used for extrapolation. Region II is controllable up to machine precision, and Region III is bounded by the computational noise floor. Region IV is considered optimizable because the error is still small, but it is not controlled. In comparing the two figures (both for small groups with large separation), extrapolation in lossy media becomes unstable (Region IV) for a smaller number of samples, P , than in the active media case. However, both can be controlled (Region II) to a degree of accuracy within machine precision. In addition, cases for large groups, or small buffers in complex media were also studied; the results appear in [31].

The highly lossy cases ($\tan \delta > 0.2$) in Fig. 4.1 have a large Region IV. This occurs for two reasons. First, the exact solution is very small, and round-off error corrupts the relative error measurement. By examining the absolute error, it can be seen that the absolute error decreases as predicted by (4.2). Although the relative error is large, the particle interaction becomes negligible and has little impact on the solution. Second, in cases where the groups have small separation, extrapolation in highly lossy cases actually loses accuracy. The reason is best explained by considering the complex α -plane and the SDPs for the lossy and active cases.

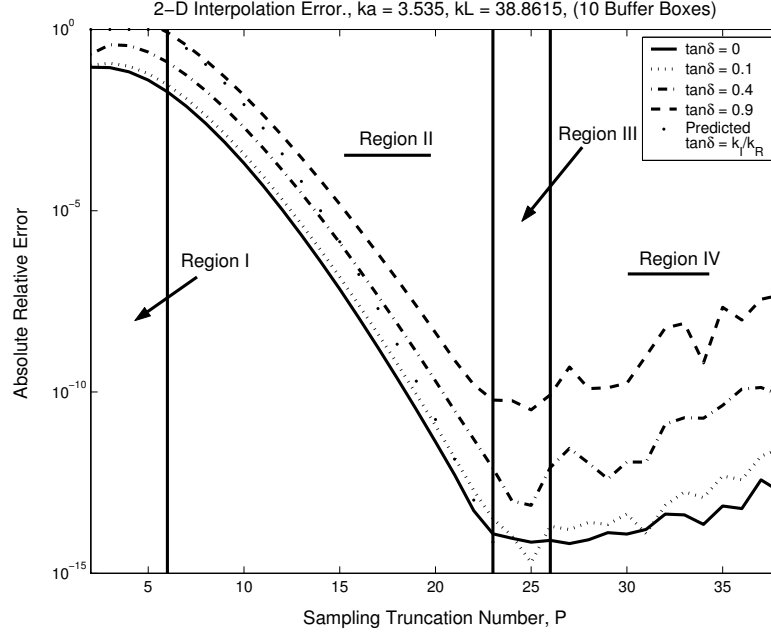


Figure 4.1: Interpolation error of small group in lossy medium with large separation. The group size is $kD = \sqrt{2}ka = 5$ and the separation is $kL = 39$, or 10 buffer boxes.

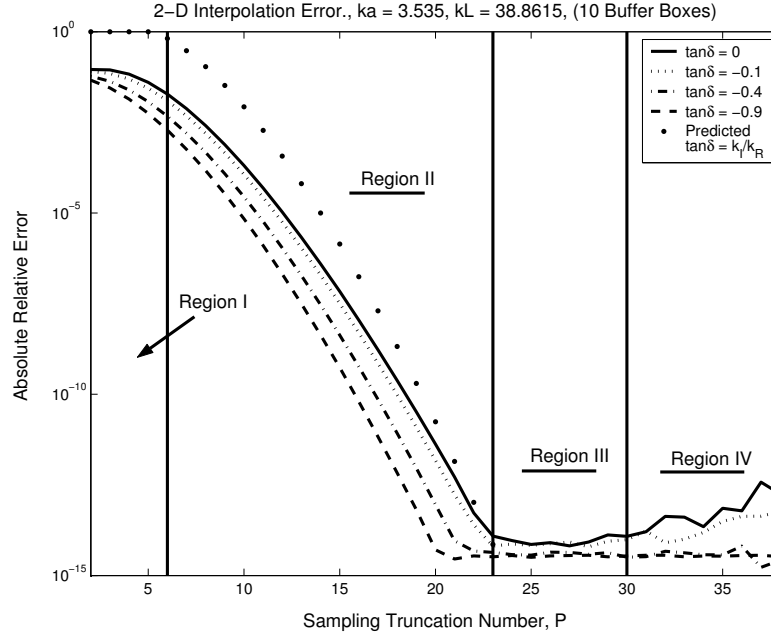


Figure 4.2: Interpolation error of small group in active medium with large separation. The group size is $kD = \sqrt{2}ka = 5$ and the separation is $kL = 39$, or 10 buffer boxes.

As seen in Figs. 4.1 and 4.2, the behavior of the extrapolation error for the lossy medium case differs from the active medium case. The difference is due to the topography of the α -plane for complex media. Referring to Figs. 3.6 and 3.7, page 25, in lossless media, Path II lies on the real axis and coincides with the constant magnitude path (CMP). Hence, interpolation of Path II is highly accurate. Additionally, only the phase of the radiation pattern changes when sampled from the CMP (real axis), so extrapolation to Paths I and III of the SDP is also performed with high accuracy.

However, the CMP for complex media is not on the real axis. Thus, the radiation patterns can change rapidly in both phase and magnitude when sampled from the real axis, and extrapolation to a given point on the SDP is not as controllable. Figures 4.3 and 4.4 illustrate the extrapolation process. The hatched solid lines represent the M-SDP for lossy and active cases, respectively. The dot-dashed line represents the constant CMP. The dotted lines represent the real-valued samples used for interpolation. The dashed lines illustrate the extrapolation from the real axis to the M-SDP. Note that all the values in the sample set are used to extrapolate to a given point on the M-SDP. In the lossy case, Fig. 4.3, extrapolation is across the CMP for Path I. It is conceivable that the resulting samples are orders of magnitude different from the magnitudes of the M-SDP values. Hence, the machine precision and roundoff error can limit the extrapolation process.

In the active case, Fig. 4.4, extrapolation does not cross the CMP for Path I, and the M-SDP is closer to the real axis. Thus, the extrapolation error is more controllable. Finally, the figures illustrate extrapolation from samples that are nearby the M-SDP point. These are the prominent samples. Although the samples from the positive real values are also used, the contribution is small. Hence, when considering the extrapolation to Path III, there are values from the negative real axis that must cross the CMP, but the contribution is minor compared to those from the positive real samples.

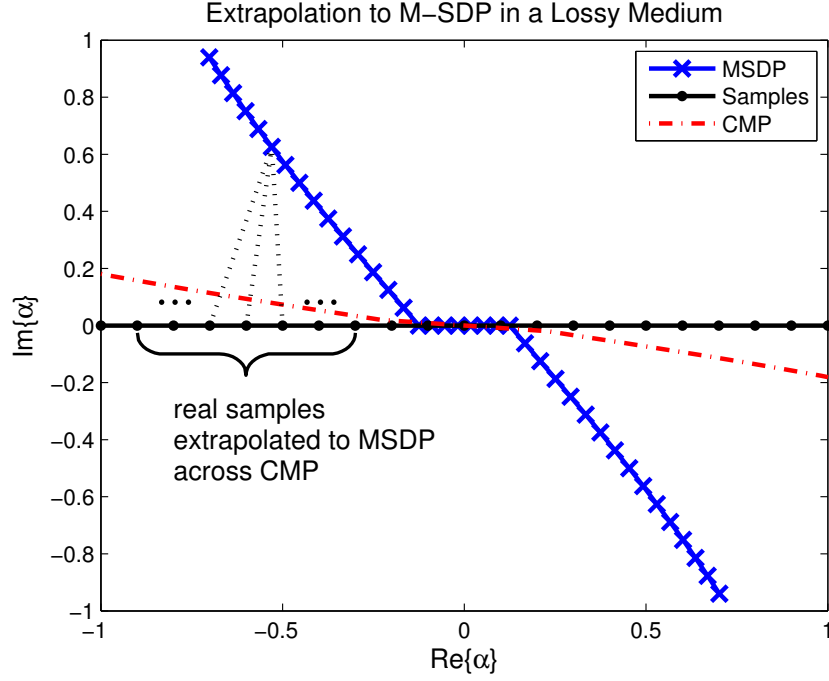


Figure 4.3: Extrapolation from real axis to MSDP in lossy media. Extrapolation must cross the constant magnitude path.

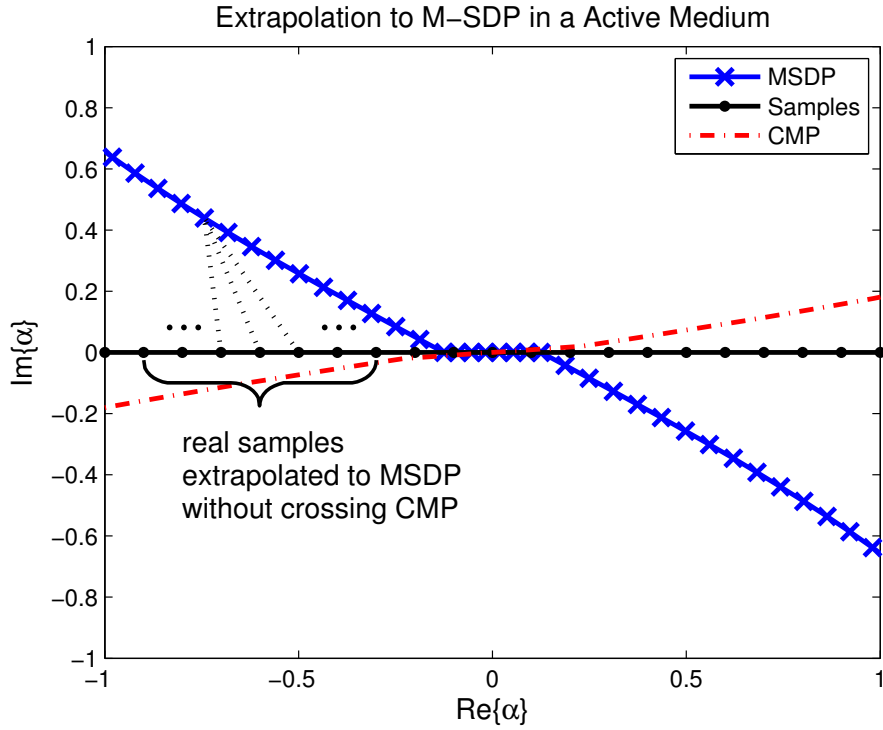


Figure 4.4: Extrapolation from real axis to MSDP in active media. Extrapolation does not cross the constant magnitude path.

4.4 Control Regions

To identify conditions when the error is controllable or optimizable, the behavior of the error is examined for various sets of the parameters: kD , kL , and $\tan \delta$. Figure 4.5 shows an example of the error versus the group separation distance, or translation distance, for a low-loss case ($\tan \delta = 0.1$). The absolute error is used instead of the relative error, because of round-off error, as discussed previously. P was set according to the refined excess bandwidth formula (4.2).

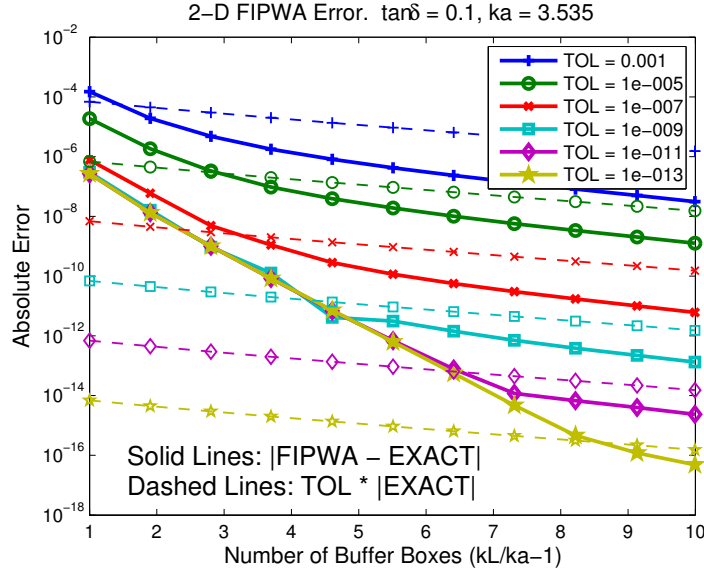


Figure 4.5: Error control of small groups by group separation in lossy media. Control occurs when the solid lines extend below the corresponding dashed lines. It is clear that the error is still low even when the desired accuracy is not achieved.

The solid lines represent the absolute error ($|FIPWA - EXACT|$) and the dashed lines show the exact solution after scaling by the error tolerance ($10^{-d_0} \times |EXACT|$). Control occurs for buffer sizes where $|FIPWA - EXACT| < 10^{-d_0} \times |EXACT|$. The numbers of boxes are not presented as ordinal numbers because, when the boxes have arbitrary orientation, they are not separated by an integer number of boxes. Hence, the number of boxes is a way to normalize the separation distance in terms of the box, or group, size. By locating where the control is lost for various parameters, a single graph was constructed to show

the minimum buffer region that is needed to achieve a desired accuracy in complex media. Figure 4.6 shows the control curves for accuracies of 3–13 digits. The abscissa denotes the range of $\tan \delta$ from gain (< 0) to loss (> 0), and the ordinate marks the minimum number of buffer boxes. Hence, the control region is above a tolerance curve, and the optimizable region lies below.

For example, the x -line denotes 7 digits of accuracy. In moderately lossy media ($\tan \delta = 0.2$), 5 buffer boxes are needed to ensure 7 digits. For fewer boxes, the accuracy loses digits of precision. In a high gain medium ($\tan \delta = -0.7$) for the same tolerance, less than a single buffer box is sufficient. It is important to remember that these charts do not show control versus no control. In highly lossy media, and a target of 9 digits of accuracy, the number of boxes needed is off the chart. However, the actual accuracy achieved is 6 digits, and may be improved, or optimized.

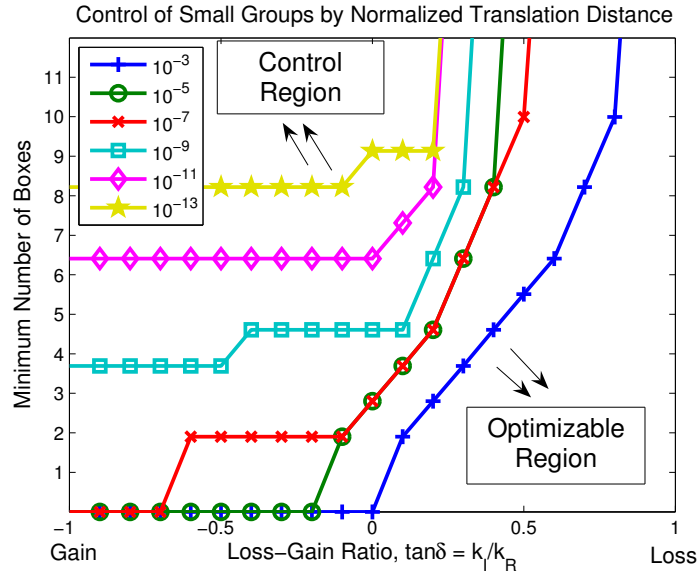


Figure 4.6: Control regions for small boxes. The relative error is used to determine controllability (as in Fig. 4.5). Translation distance is normalized by box size and the control region identifies when arbitrary accuracy can be achieved. The optimizable region shows where the error is acceptable, but arbitrary accuracy is not achievable.

Figure 4.7 presents the same type of control chart for large boxes. The error for large boxes is seen to be highly controllable in active media, but not so in lossy media. The

accuracy loses between 4 and 8 digits (not shown on graph) for target accuracies of single and double precision, respectively. Also, the curves nearly overlay each other, suggesting that $\tan \delta$ is the most influential factor in controlling the error for large groups.

The optimizable region can be used by simply requiring a higher accuracy than what is truly needed. Although some digits will be lost, single-digit precision can still be reached. Based on the control and optimizable regions, 2-D FIPWA is a viable solution for use in the Sommerfeld integrals in layered-media problems. Hu and Chew [14] used the 2-D multipole translator while noting that it can be replaced with the 2-D FIPWA translator if proper attention is given to the SDP in complex media. These results suggest that the 2-D FIPWA for complex media can be used in the 3-D FIPWA translator for layered-media.

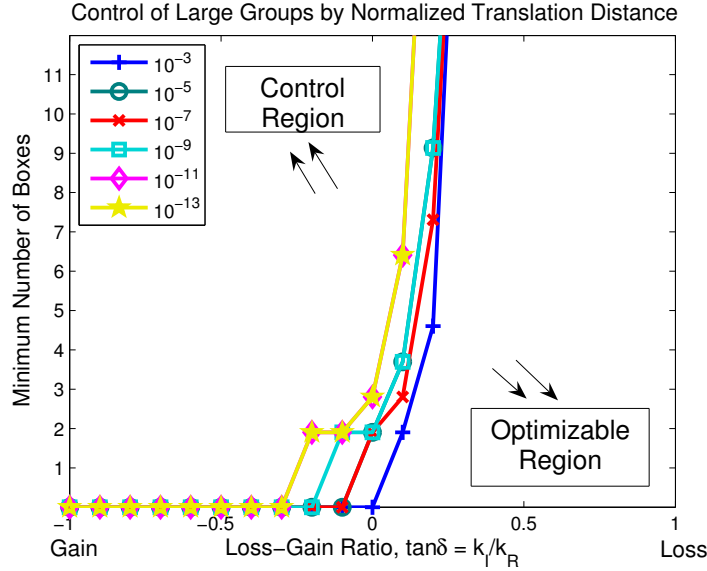


Figure 4.7: Control regions for large boxes. Relative error for large boxes is not controllable for highly lossy ($\tan \delta > 0.2$) media.

Effective broadband behavior of k_ρ

In the above results, k was assumed to have constant magnitude. In practice, k_ρ will take on varying magnitudes and effective material parameters (lossless, lossy, or active) because $k_\rho = k \sin \theta$ for complex θ . Effectively, this causes k_ρ to range from low to high frequency.

Extrapolation functions

It is important to note that these results were generated with the sinc extrapolation function. Alternative extrapolation may be more useful and should be considered for enabling high accuracy in the cases of small boxes.

4.5 Summary

In this chapter, the error sources of the 2-D plane-wave translator for complex media were shown to be controllable. Furthermore, regions of strict control and partial control (optimizable) were identified. The error for the 2-D FIPWA in complex media is controllable for low-loss problems, and while not completely controlled for high losses, it is still acceptable for practical applications. In contrast, the active media cases are highly controllable. The difference is due to the topography surrounding the steepest descent path. In lossy material, the constant magnitude path interferes with extrapolation from real samples to the M-SDP, but in active media, the constant magnitude path does not influence the extrapolation. Thus, the active case is more accurate than the lossy case.

Control regions were also presented for small and large group sizes in complex media. In a low loss medium ($0 < \tan \delta < 0.2$), the small boxes require several buffer boxes for arbitrary precisions. However, for practical applications, where 2–3 digits are required, a single buffer box is sufficient. For highly lossy cases ($\tan \delta > 0.2$), several buffer boxes are still needed. In contrast, large buffer boxes lose accuracy only for large losses. However, the error is still acceptable for most applications.

In consideration of large, layered-medium problems, the 3-D FIPWA is a natural formulation for the Sommerfeld integrals. The 2-D FIPWA translator provides a simpler and accurate option to the traditional multipole translator, when used as the 2-D Green's function in 3-D FIPWA.

CHAPTER 5

MULTIPOLE-FREE ALGORITHM

5.1 Introduction

Scattering by objects over or embedded in layered media is difficult and time consuming to solve with integral equation solvers because the Green's function is expressed in terms of complicated Sommerfeld integrals. In addition, scattering problems are often large, requiring upwards of millions of unknowns. Several methods have been developed for this class of problem, but most rely on the multilevel paradigm, as first proposed by [11], and implemented in the multilevel fast multipole algorithm (MLFMA) [5,6]. Recent work on fast algorithms for layered-medium problems include higher order hierarchical basis functions [32], the discrete complex image method (DCIM) with MLFMA [15,16], and steepest descent path integrals with the fast inhomogeneous plane wave algorithm (FIPWA) [10,14].

In [32], the goal is to reduce the number of unknowns, N , to achieve a faster solution. To reduce N , the target is meshed with curvilinear quadrilateral patches, and the basis functions are Legendre polynomials. This allows the higher order basis functions to be adapted to patches that span more than one medium. The computational complexity is higher than MLFMA with Rao, Wilton, Glisson (RWG) basis functions [18], but the number of unknowns can be made significantly smaller with the curvilinear patches. However, the method is only beneficial to geometries that can be represented with few curvilinear patches and the computational complexity is still at least $\mathcal{O}(N^2)$ with an iterative matrix solver.

In [15], MLFMA was extended from earlier work [33] with the fast multipole algorithm, in which complex images were computed with approximate reflection coefficients. Additionally, the multipole expansion needed a prohibitive number of terms for sources that reside in highly lossy media. The later work extended FMA to MLFMA, and achieved $\mathcal{O}(N \log N)$ complexity of MLFMA. However, the cost to compute the complex images was equal to the free space case, and increased memory requirement up to 50 %.

The most recent work by the same group introduced parallel algorithms to solve scattering by multiple scatterers [16]. In addition, a new MLFMA technique was proposed where one solves the scattering solutions on separate processors, and then iteratively sums contributions. These approaches, as well as the next, still rely on the multipole expansion of the Green's function. Yet, FIPWA allows a simpler formulation for removing the multipole expansion [10, 14, 22].

FIPWA was presented as an alternative to MLFMA and was shown to achieve $\mathcal{O}(N \log N)$ complexity in memory and processing time. In FIPWA, the spectral dyadic Green's function for layered media achieves high accuracy because it rigorously treats the reflection coefficients, and still has fast computation of the dyadic components by use of steepest descent path integrals. In the case of the layered media problem, FIPWA only costs up to 20 % more in memory than the free space case.

FIPWA still relies on the multipole expansion, yet the formulation is highly adaptable to a completely multipole-free form. The previous chapters laid the groundwork for the multipole-free fast algorithm, with the implication that one can simply substitute the 2-D FIPWA translator for the 2-D FMM translator.

Hence, the goal of this chapter is to demonstrate the multilevel multipole-free fast algorithm. In addition, the advantages and disadvantages are discussed for a simple black box substitution of the proposed translator in the Multipole-Free Fast Inhomogenous Plane-Wave Algorithm (ML-FIPWA), and the new multipole-free algorithm is validated by com-

parison to FIPWA with a benchmark problem. Finally, ML-FIPWA is demonstrated for a larger complex target and shown to retain $\mathcal{O}(N \log N)$ complexity in total memory cost and matrix-vector product processing time.

5.2 Formulation

5.2.1 Combined field integral equation

The multipole-free fast algorithm is implemented in the multipole-free fast inhomogeneous plane-wave algorithm (MF-FIPWA) as a method of moments (MOM) solution to the combined field integral equation (CFIE). Acceleration of the (MOM) matrix solution is achieved with the conjugate gradient iterative matrix solver and the multilevel paradigm of MLFMA and FIPWA. The CFIE is expressed as a combination of the electric and magnetic field integral equations (EFIE and MFIE, respectively) as $\text{CFIE} = \alpha \text{EFIE} + (1 - \alpha) \text{MFIE}$, and the coupling parameter α is typically set to 0.5. Each integral equation component is succinctly expressed in terms of integral operators as

$$\text{EFIE :} \quad -\hat{\mathbf{n}} \times \mathbf{E}^{inc}(\mathbf{r}) = \hat{\mathbf{n}} \times \mathcal{L}(\mathbf{J}(\mathbf{r})), \quad (5.1)$$

$$\text{MFIE :} \quad -\hat{\mathbf{n}} \times \mathbf{H}^{inc}(\mathbf{r}) = \hat{\mathbf{n}} \times \mathcal{K}(\mathbf{J}(\mathbf{r})), \quad (5.2)$$

where $\mathbf{J}(\mathbf{r})$ is the unknown electric current density and $\hat{\mathbf{n}}$ is the unit normal to the surface, S , of the scatterer. Finally, the integral equation operators $\mathcal{L}$ and $\mathcal{K}$ are expressed in terms of the symmetric dyadic Green's function $\overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}')$ as

$$\mathcal{L}[\mathbf{J}(\mathbf{r})] = \int_S \overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}'), \quad \forall \mathbf{r} \in S \quad (5.3)$$

$$\mathcal{K}[\mathbf{J}(\mathbf{r})] = \nabla \times \int_S \overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}') dS' - \frac{1}{2} \mathbf{J}(\mathbf{r}). \quad (5.4)$$

The components of $\overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}')$ for layered media were presented in [19] as the sum of direct, TE, and TM scalar Green's functions, and more recently rederived in a more elegant manner

in [34]. The components are repeated here for completeness.

$$\begin{aligned}\hat{\alpha} \cdot \overline{\mathbf{G}} \cdot \hat{\alpha}' &= (\hat{\alpha}_s \cdot \hat{\alpha}_s')(g^P - g^{\text{TE,R}}) + \alpha_z \alpha'_z (g^P + g^{\text{TM,R}}) \\ &+ \frac{1}{k_1^2} \hat{\alpha} \cdot \nabla \nabla \cdot \hat{\alpha}'' g^{\text{TM,R}} \\ &+ \hat{\alpha} \cdot \nabla_s \nabla_s \cdot \hat{\alpha}'' g^{\text{EM}},\end{aligned}\tag{5.5}$$

with

$$\hat{\alpha} = \hat{\alpha}_s + \hat{\mathbf{z}} \alpha_z \tag{5.6}$$

$$\hat{\alpha}' = \hat{\alpha}'_s + \hat{\mathbf{z}} \alpha'_z \tag{5.7}$$

$$\hat{\alpha}'' = -\hat{\alpha}'_s + \hat{\mathbf{z}} \alpha'_z \tag{5.8}$$

and

$$g^\beta = \begin{cases} \frac{i\omega\mu}{4\pi} \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} = \int_{\Gamma_\theta} d\theta W^P(\theta) e^{i\mathbf{k}_s \cdot (\mathbf{r}_s - \mathbf{r}_s')} e^{ik_z|z-z'|}, & \beta = P \\ \int_{\Gamma_\theta} d\theta W^{\text{TM}}(\theta) H_0^{(1)}(k_\rho \rho) e^{i\mathbf{k}_s \cdot (\mathbf{r}_s - \mathbf{r}_s')} e^{ik_z(z+z')}, & \beta = \text{TM} \\ \int_{\Gamma_\theta} d\theta W^{\text{TE}}(\theta) H_0^{(1)}(k_\rho \rho) e^{i\mathbf{k}_s \cdot (\mathbf{r}_s - \mathbf{r}_s')} e^{ik_z(z+z')}, & \beta = \text{TE} \end{cases} \tag{5.9}$$

for the primary, TM reflected, and TE reflected components, and $g^{\text{EM}} = (\frac{1}{k_\rho^2} g^{\text{TE}} - g^{\text{TM}})$. Γ_θ is the path of integration in the complex θ -plane. The weight functions are

$$W^P(\theta) = -\frac{\omega\mu_0}{8\pi^2} k \sin \theta \tag{5.10}$$

$$W^{\text{TM}}(\theta) = -\frac{\omega\mu_0}{8\pi^2} k \sin \theta \tilde{R}^{\text{TM}}(k_{1z}, \dots, k_{Nz}) \tag{5.11}$$

$$W^{\text{TE}}(\theta) = \frac{\omega\mu_0}{8\pi^2} k \sin \theta \tilde{R}^{\text{TE}}(k_{1z}, \dots, k_{Nz}), \tag{5.12}$$

where $k_z = k \cos \theta$, $k_{iz} = \sqrt{k_{iz}^2 - k_\rho^2}$ and $\tilde{R}$ is the generalized reflection coefficient for the N -layer medium, as defined in [1].

The unknown current is expanded with the well-known Rao, Wilton, Glisson (RWG) basis functions [18],

$$\mathbf{J}(\mathbf{r}) = \sum_{n=1}^N I_n \mathbf{\Lambda}_n(\mathbf{r}), \tag{5.13}$$

to form the matrix equation

$$\sum_{n=1}^N (Z_{mn}^E + Z_{mn}^M) I_n = V_m. \tag{5.14}$$

The matrix and vector elements are

$$Z_{mn}^E = \frac{ik\eta}{4\pi} \int_S dS \mathbf{\Lambda}_m(\mathbf{r}) \cdot \int_S \overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{\Lambda}_n(\mathbf{r}') dS' \quad (5.15)$$

$$Z_{mn}^M = - \int_S dS \frac{\mathbf{\Lambda}_m \cdot \mathbf{\Lambda}_n}{2} + \frac{1}{4\pi} \int_S dS \mathbf{\Lambda}_m(\mathbf{r}) \cdot \hat{\mathbf{n}} \times \nabla \times \int_S g(\mathbf{r}, \mathbf{r}') \mathbf{\Lambda}_n(\mathbf{r}') dS' \quad (5.16)$$

$$V_m = - \left(\alpha \int_S \mathbf{\Lambda}_m(\mathbf{r}) \cdot \mathbf{E}^{inc}(\mathbf{r}) dS + \eta(1 - \alpha) \int_S \mathbf{\Lambda}_m(\mathbf{r}) \cdot \hat{\mathbf{n}} \times \mathbf{H}^{inc}(\mathbf{r}) dS \right). \quad (5.17)$$

5.2.2 Multilevel multipole-free fast algorithm

To achieve a multilevel algorithm, the scalar components of $\overline{\mathbf{G}}$ are factored and diagonalized in a recursive fashion. In MF-FIPWA, the initial factorization is generated with the multipole-free 2-D translator in the same fashion as FIPWA uses the 2-D FMA translator [10, 14]. Then, successive factorization is aided with interpolation between levels as

$$\begin{aligned} g^\beta(\mathbf{r}_j, \mathbf{r}_i) &= \boldsymbol{\beta}_{jJ_{l_{max}}}^t \cdot \overline{\mathbf{I}}_1^t \cdot \overline{\boldsymbol{\beta}}_{J_{l_{max}}J_{l_{max}+1}} \cdot \overline{\mathbf{I}}_2^t \cdots \\ &\quad \overline{\boldsymbol{\beta}}_{J_{L+1}J_L} \cdot \mathcal{T}_{J_LJ_L} \cdot \overline{\boldsymbol{\beta}}_{J_LJ_{L+1}} \cdots \overline{\mathbf{I}}_2 \cdot \overline{\boldsymbol{\beta}}_{J_{l_{max}+1}J_{l_{max}}} \cdot \overline{\mathbf{I}}_1 \cdot \boldsymbol{\beta}_{J_{l_{max}}i}, \end{aligned}$$

where g^β represents the TM, TE, or TEM component, $\boldsymbol{\beta}$ is the shifted radiation and receiving patterns, $\mathcal{T}$ is the diagonalized translator between patterns, and $\overline{\mathbf{I}}$ and $\overline{\mathbf{I}}^t$ represent interpolation and antepolation matrices.

In addition, each component of the dyadic Green's function represents the direct and reflected contributions in the form of generalized reflection coefficients [1], and Sommerfeld integrals. Calculation of the Sommerfeld integrals is accelerated with integration along the steepest descent path (SDP) as shown in Fig. 2.6, page 14 [22, 23]. By rigorously treating the Sommerfeld integral, the algorithm maintains the integrity of the physics and captures surface wave and guided wave contributions, unlike approximations such as the discrete complex image method [15, 33].

In [10, 14], the 3-D translator was constructed with the 2-D fast multipole algorithm. It is repeated here for clarification.

$$\mathcal{T}_{JI,3D\text{fipwa}}(\Omega_s) = \int_{\Gamma_\theta} d\theta W^\beta(\theta) e^{ikz_{JI} \cos \theta} \mathcal{T}_{JI,\text{fma}}(\theta, \alpha_s) I(\theta - \theta_s), \quad (5.18)$$

$$\mathcal{T}_{JI,\text{fma}}(\theta, \alpha_s) = \int_0^{2\pi} d\alpha \frac{1}{2\pi} \sum_{p=-P}^P H_p^{(1)}(k \sin \theta \rho_{JI}) e^{-ip(\alpha - \phi_{JI} - \pi/2)} I(\alpha - \alpha_s), \quad (5.19)$$

where Γ_θ is the modified steepest descent path.

In the proposed multipole-free algorithm, the 3-D translator is constructed with the 2-D FIPWA translator, whereby no multipoles are used.

$$\mathcal{T}_{JI,3D}(\Omega_s) = \int_{\Gamma_\theta} d\theta \int_{\Gamma_\alpha} d\alpha W^\beta(\theta) e^{ikz_{JI} \cos \theta} e^{i\mathbf{k}_\rho(\alpha) \cdot \boldsymbol{\rho}_{JI}} I(\theta, \alpha, \theta_s, \alpha_s), \quad (5.20)$$

in which the 2-D FIPWA translator is

$$T_{2D\text{fipwa}}(\alpha_s) = \frac{1}{\pi} \int_{\Gamma_\alpha} d\alpha e^{i\mathbf{k}_\rho(\alpha) \cdot \boldsymbol{\rho}_{JI}} I(\alpha - \alpha_s), \quad (5.21)$$

As in MLFMA and 3-D FIPWA, the translation and interpolation matrices are stored for different levels in the tree, and the matrix-vector product is accelerated with computational complexity of $\mathcal{O}(N \log N)$.

5.2.3 Surface wave contributions and poles

When a scatterer is placed on the surface, or slightly above the layered medium, lateral surface waves [1] are excited. In most cases, the contribution is small and can be dismissed, as in [15, 33]. Yet, the contribution is easily managed by defining paths around the appropriate branch cut. By defining the weight function W^β for the correct Riemann sheet, the additional computational burden is negligible. This approach was demonstrated in FIPWA for layered media and buried object problems. Complete details are given in [10, 14].

Similarly, the poles due to $\tilde{R}^{\text{TE}, \text{TM}}$ are also easily included. When poles are present, residue theory is used to add the contribution. Of course, an appropriate pole finding algorithm [21] must first locate the poles, but storage and computational costs are minimal.

In terms of fast algorithms, these approaches are unique to FIPWA and the multipole free version, MF-FIPWA. Each is mentioned here to show how FIPWA and MF-FIPWA capture the physical phenomena that approximation methods cannot.

5.3 Error Control

As seen in various works on error control [20, 29, 30, 35], there are a finite set of error sources in the fast algorithm. Typically, one strives to control the error so it decreases exponentially. Hence, the expressions “controllable error” and “exponentially controllable error” both mean the computational error decreases exponentially. Common to each multilevel algorithm are controllable error sources of numerical integration and interpolation, and in MLFMA and FIPWA, also controllable error from truncation of the multipole expansion. FIPWA and MF-FIPWA have controllable error from interpolation and extrapolation of the radiation patterns. For each of the error sources, error analysis has been provided in the references, but to properly control the error, one must understand the physical significance of the source. Hence, each of MF-FIPWA’s error sources is presented with physical interpretation.

5.3.1 Integration along the SDPs

Referring to Eq. (5.20), the integrand is interpreted as a summation of inhomogeneous plane waves. Figure 2.6, page 14, represents the modified paths for Γ_θ and Γ_α , where points on path II are propagating waves and points on paths I and III represent evanescent waves. To accurately compute the double integral, each path must be of the appropriate length, and special attention must be given to Γ_α .

Selecting Γ_α

As discussed in Section 3.2, $\theta = \theta_R + i\theta_I$, so that $k_\rho = k \sin \theta$ implies that each point on Γ_θ has a corresponding path Γ_α . Figure 5.1 illustrates how each value of k_ρ has a unique SDP

that depends on the effective complex media. To retain the accuracy of the outer integral along Γ_θ , the inner integral along Γ_α must be computed with high accuracy.

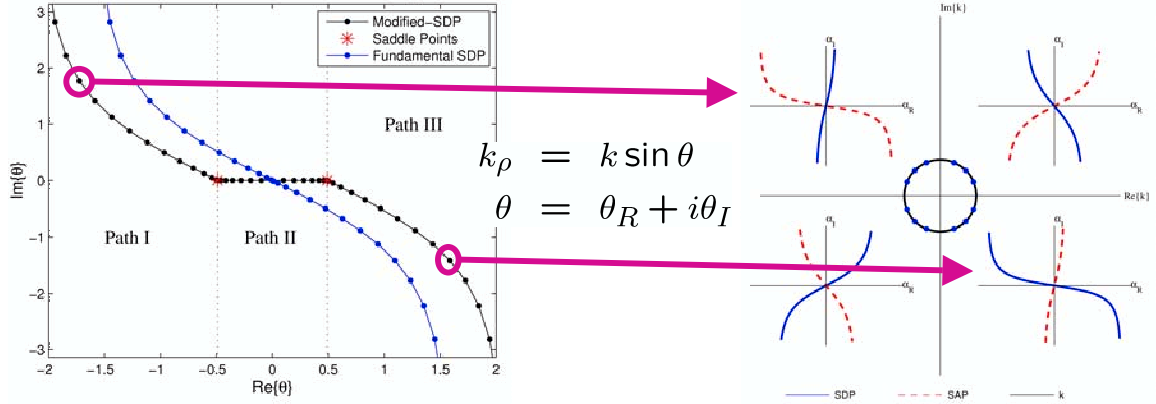


Figure 5.1: Illustration of how k_ρ depends on $k \sin \theta$.

Highly accurate integration on Γ_α

When the double SDP is computed aside from interpolation and extrapolation in MF-FIPWA, high accuracy is easily achieved with Gauss-Laguerre integration along paths I and III and trapezoidal or Gauss-Legendre rules on path II. However, when interpolation and extrapolation of the radiation patterns are included, the particular points on paths I and III influence the accuracy of interpolation and extrapolation. The discrete form of Eq. (5.21) is

$$T_{2D\text{fipwa}}(\phi_s) = \frac{1}{\pi} \sum_{q=1}^{N_\alpha} w_q e^{i\mathbf{k}_\rho(\alpha_q) \cdot \mathbf{p}_{JI}} I(\alpha_q - \phi_s), \quad (5.22)$$

where $\phi_s \in (0, 2\pi)$, N_α is the number of quadrature points, and w_q and α_q are the quadrature weights and nodes. In this form, it appears as though the 2D translator is formed by antinterpolation, or performing the inverse of interpolation, from the SDP to a set of propagating waves. In other words, the stored translator is created by filtering the mix of propagating

waves and evanescent waves to retain only representative propagating waves. In this light, one can think of paths I and III as having two regions: a shallow evanescent region and a deep evanescent region, illustrated in Fig. 5.2. The shallow waves are dominant, so more shallow than deep waves should be used to form the translator for long distance translations. Similarly, near translations need to retain more evanescent waves.

Under lossless conditions, the Gauss-Laguerre rule inherently selects more shallow points when parameterizing the SDP with the real part: $\alpha(s) = s + if(s)$; but when the background medium is complex, the path is defined by mapping from the real axis to the SDP with the quadratic map $-s^2 = i\mathbf{k}(\beta, \alpha) \cdot \mathbf{r}_{JI'}$. This causes the shallow points to become sparse and the deep points to become dense, and poses a challenge to retaining the accuracy of the translator. Either more points are needed in the integration rule to increase the density of the shallow region, or alternative quadrature rules with high accuracy are needed. In this work, N_α is increased in proportion to the length of paths I and III to achieve 15 digits accuracy of the 2D translator.

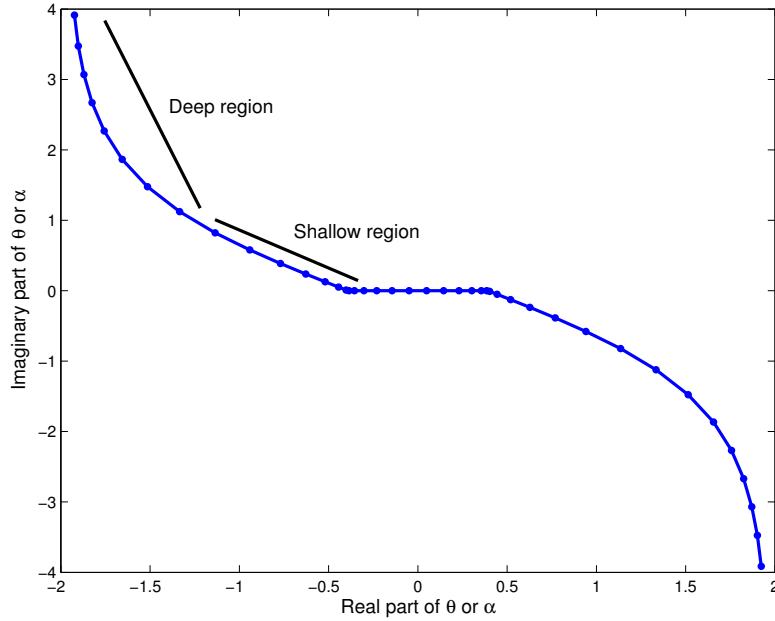


Figure 5.2: Illustration of the shallow and deep evanescent regions of paths I and III.

5.3.2 Interpolation and extrapolation of radiation patterns

Interpolation

The key features of FIPWA and MF-FIPWA are interpolation and extrapolation of radiation and receiving patterns from a stored set of samples. Along path II, the SDP is purely real in $(-\pi, \pi)$. Thus, sinc interpolation provides high accuracy when sampling at the Nyquist rate of twice the bandwidth. It has been shown that the number of samples is correctly set with the refined excess bandwidth formula $P = kD + 1.8d_0^{2/3}(kD)^{1/3}$ [29], where P represents the single sided bandwidth of the sinc function. As a result, interpolation on path II has exponentially controllable error.

Extrapolation

Under lossless conditions, the sinc interpolation function can be used to extrapolate the radiation patterns, but in MF-FIPWA, the 2-D wavenumber k_ρ is complex. This alters the form of the path as discussed in [20], and makes extrapolation with high accuracy difficult. When extrapolating the radiation patterns from the set of real samples to points in the deep evanescent region, the integrand of Eq. (5.20) is extremely small. These contributions can be on the order of the machine precision and add as numerical noise. Subsequently, the extrapolation is sensitive to oversampling. To avoid extrapolation to very small values, the path length must be shortened. Yet, shortening the path length decreases the accuracy of the translator. Empirical studies show that it is better to use Lagrange polynomials to extrapolate the radiation patterns to the deep evanescent waves, instead of the sinc function, at a higher computational cost.

5.3.3 Interpolation between levels

Typically, the sampling rate for interpolation between levels in the MLFMA tree structure is predetermined according to the bandwidth of the radiation patterns. The error has been

shown to be exponentially controllable when the refined excess bandwidth formula is used to set the number of samples [17]. As a matter of convenience, the radiation pattern, when thought of as a sphere, is sampled at the same rate at each latitude, which greatly oversamples the regions near the north and south poles. This does not impede the matrix-vector multiply because the interpolation is local with few samples, but it can create added overhead during the setup stage of the algorithm. It is mentioned here, because this type of storage is not optimum for the MF-FIPWA translator. This topic is discussed more in the next chapter.

5.4 Numerical Results

5.4.1 Validating the multilevel multipole-free algorithm

Set up for validation problem

MF-FIPWA is validated by comparison to FIPWA (previously validated by comparison to a full matrix solution [10, 14, 21]). The validation problem [21], is to compute the bistatic radar cross section of a right circular cylinder above a two-layer medium. Figure 5.3 shows the problem setup. The frequency is 600 MHz, and the number of unknowns is 9 708 with 5 levels and the smallest box set to $\lambda/10$. Excitation is incident from $\theta^{inc} = 60^\circ, \phi^{inc} = 0^\circ$, and the observation angles are $\theta^{obs} = 60^\circ, \phi \in [0, 2\pi)$. The cylinder is placed 0.2 m above the layered medium, and the medium parameters are $\epsilon_0 = 1.0$, $\epsilon_1 = 2.56$ with a thickness of 0.3 m, and $\epsilon_2 = (6.5, 0.6)$. Following the error control of Chapter 4, the accuracy of the 2-D translator is set for double machine precision, or 14 digits.

Validation of MF-FIPWA

Figure 5.4 shows the comparison between the two algorithms and a full matrix solution. MF-FIPWA has excellent agreement for both vertical and horizontal polarizations.

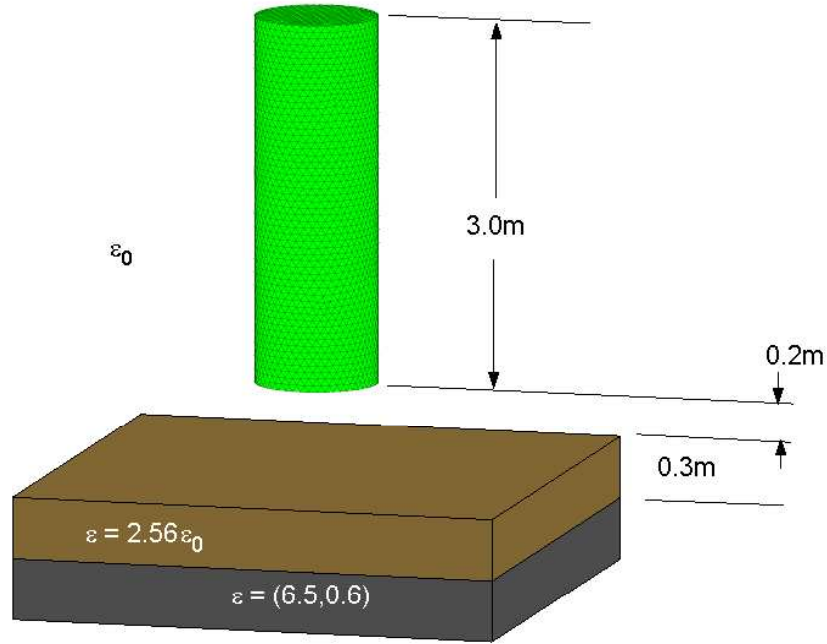


Figure 5.3: Validation scattering case. Bistatic scattering of a right circular cylinder over a 2-layer medium. The frequency is 600 MHz; the number of unknowns is 9 708, and 5 levels are used.

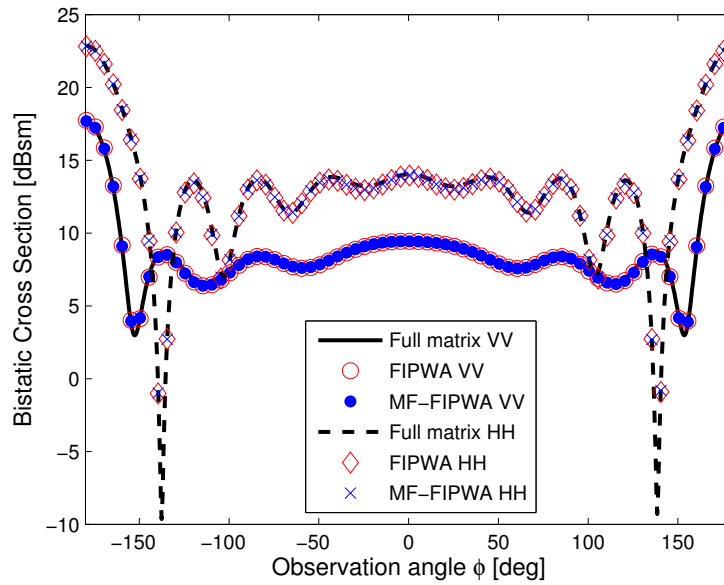


Figure 5.4: Comparison of MF-FIPWA with FIPWA. Shown are results of the validation problem of Fig. 5.3.

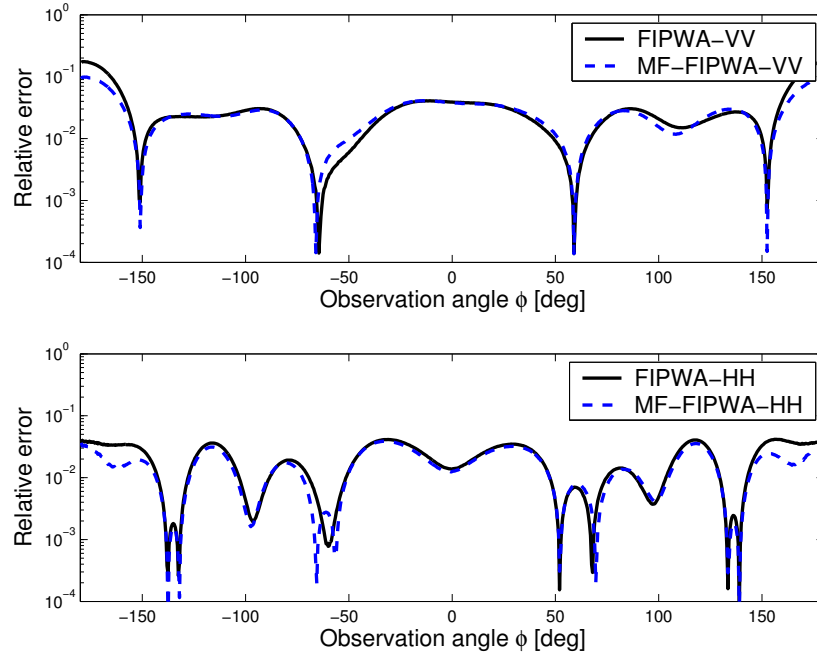


Figure 5.5: Error of FIPWA and MF-FIPWA relative to the full matrix solution. The validation case is shown with an average relative error of 3.0%.

A quantitative comparison is made by computing the relative error of the bistatic radar cross section. Figure 5.5 shows the relative error between FIPWA and the full matrix solver, and for MF-FIPWA and the full matrix solver across the bistatic angles. Again, the agreement is very good with an average relative error of 3.0%.

5.4.2 Scaling and efficiency

Before studying a complex target, the scaling of the multilevel multipole-free algorithm is shown to be equivalent to FIPWA for moderate to large size spheres. As FIPWA, MF-FIPWA achieves $\mathcal{O}(N \log N)$ complexity in memory and processing time per matrix-vector multiply for problem sizes of $N = 10\,002, 101\,568, 252\,300, 504\,300$, and $1\,002\,252$ unknowns. Figure 5.6 illustrates the scaling for moderately sized problems.

Because the multilevel paradigm is used, the scaling is not a surprise. Note that all of these results were run on a single Sun-Blade 1000 processor with 8 GB of RAM.

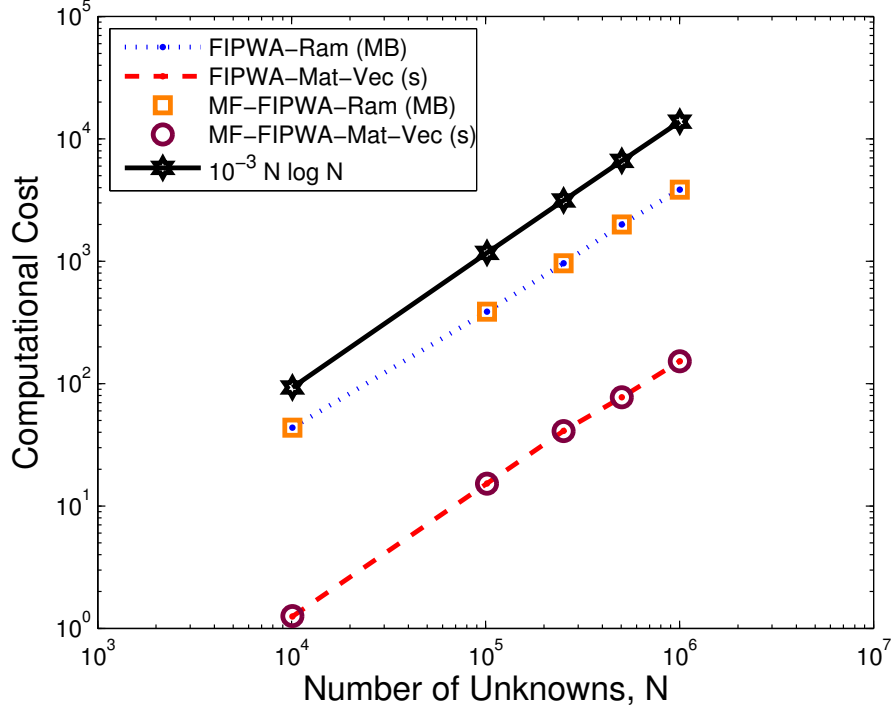


Figure 5.6: Scaling of memory and processing time for matrix-vector product. Memory is measured in MBytes and time is measured in cpu seconds.

5.4.3 Moderate sized complex target

To demonstrate the code for a complex target, the bistatic cross section is computed for the fictitious VFY-218 above a lossy half-space (Fig. 5.7). The frequency is 100 MHz, $N = 163,344$ at 7 levels, and the dielectric constant is $\epsilon = (6.5, 0.6)$. Excitation is at $\theta^{inc} = 60^\circ, \phi^{inc} = 0^\circ$. The agreement between FIPWA and MF-FIPWA is excellent, as shown in Fig. 5.8, and the total computational cost is 476 MB of RAM and less than 2.5 hours for both VV and HH polarization.

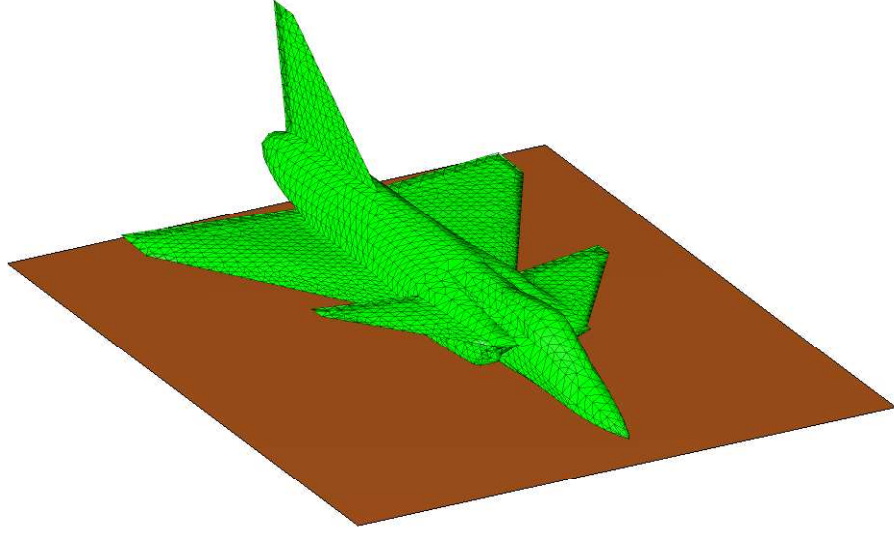


Figure 5.7: VFY-218 above lossy earth, $\epsilon = (6.5, 0.6)$. Note that the discretization shown in the figure is for illustration purposes only. The aircraft is situated 1 m above the ground.

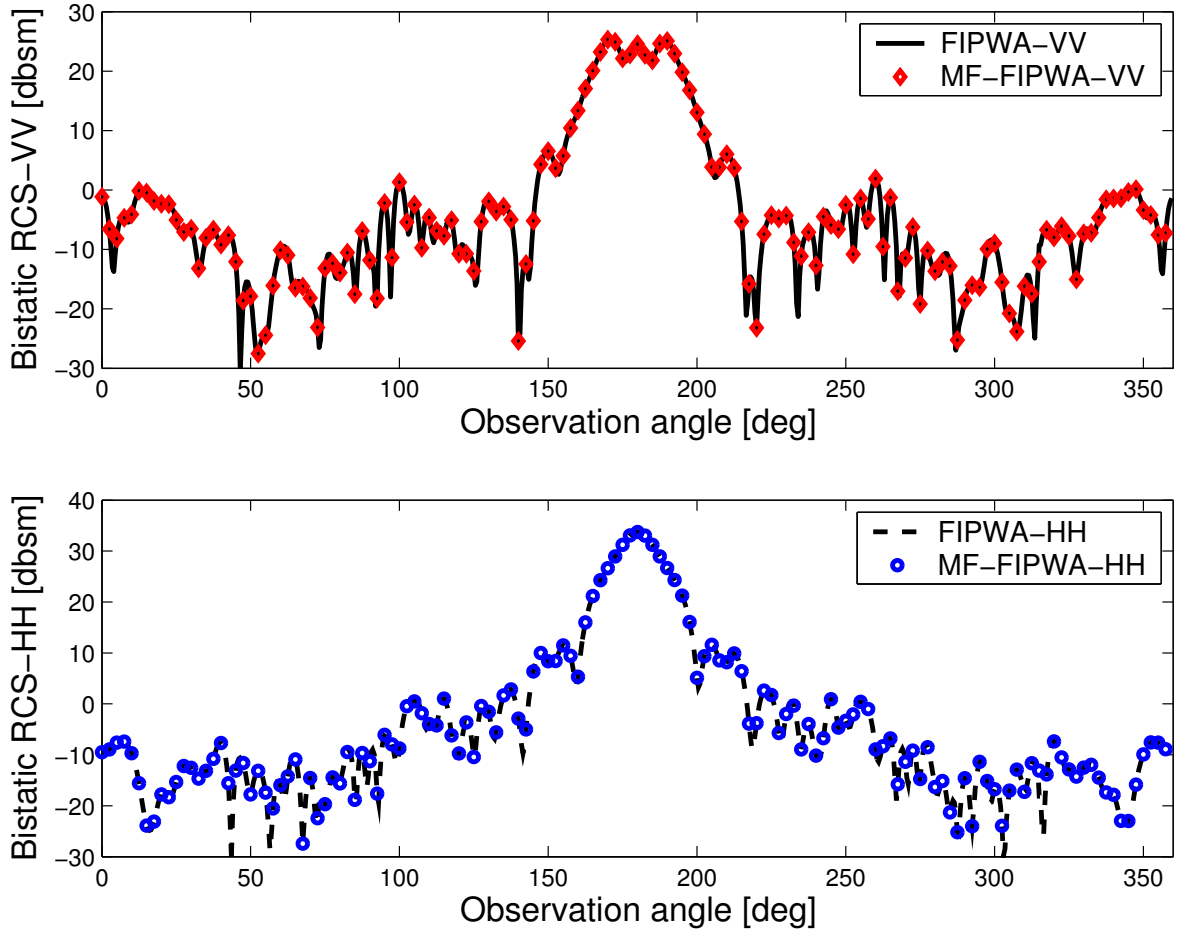


Figure 5.8: Bistatic radar cross section of VFY-218 above lossy earth.

5.5 Summary

The multilevel multipole-free fast algorithm is demonstrated for layered media problems with $\mathcal{O}(N \log N)$ complexity of memory and processing time. By carefully choosing the SDP for the 2-D translator, the accuracy is retained under the MLFMA paradigm, and equals FIPWA in complexity. The benefit of MF-FIPWA over FIPWA is that the translator is simpler, and with proper implementation, can achieve faster setup.

CHAPTER 6

COMPARISON OF FIPWA AND MF-FIPWA

6.1 Introduction

The fast multipole algorithm (FMA) [3,4] and multilevel fast multipole algorithm (MLFMA) [6,7] greatly advance computational electromagnetics by reducing the storage cost and computational complexity of boundary integral methods to $\mathcal{O}(N \log N)$. Applications range from free-space radiation and scattering problems to S-parameter extraction on multilevel chips [36]. One application of interest is electromagnetic scattering by an object in the vicinity of a layered, or stratified, medium. Recent work showed that MLFMA is applicable to the layered medium problem when combined with the discrete complex image method (DCIM) [16,33]. This approach maintains the efficiency of MLFMA and nicely approximates scattering contributions from the layered medium, but requires many complex image contributions. Alternatively, the fast inhomogeneous plane wave algorithm (FIPWA) [14] uses the MLFMA paradigm, but it rigorously computes the Sommerfeld integral that is inherent in layered medium problems. Using generalized reflection coefficients, FIPWA only requires a single image contribution from the multilayer medium, making the setup time in FIPWA faster than MLFMA with DCIM.

The latest work with FIPWA showed that the multipole expansion is replaceable with the 2-D form of FIPWA to create a completely multipole-free fast algorithm for the layered medium problem [37], called the multipole-free FIPWA (MF-FIPWA). The purpose of this chapter is to compare the computational cost and accuracy of FIPWA and MF-FIPWA. The chapter is organized as follows. First, the scope of the study is outlined with a brief description of the application problem and a review of the primary computation in the dyadic Green's function for layered media. Second, the layered medium translation matrix is presented for each algorithm followed by a general method to construct the matrix with $\mathcal{O}(N)$ efficiency. Finally, numerical results of accuracy and setup cost show the advantages of each algorithm and are applied in a mixed-form algorithm.

6.2 Scope of Comparison

The comparison focuses on the accuracy and efficiency of the 3-D translation matrix for layered medium. In the MLFMA paradigm there are three types of translation matrices [17]: outgoing-to-outgoing (O2O) shifting matrices that translate a child group to a parent group, incoming-to-incoming (I2I) shifting matrices that translate a parent group to a child group, and outgoing-to-incoming (O2I) matrices that translate between sibling groups. For layered medium applications there are also O2I matrices for the layered, or stratified, medium contribution that are denoted as O2I-SM translation matrices. The O2O and I2I matrices are not discussed because they are already multipole-free [21].

As the bistatic scattering problem and fast algorithms are thoroughly discussed in the references and Chapters 2–5, this chapter emphasizes how FIPWA and MF-FIPWA compute the O2I-SM translation matrix. In the fast algorithm, the dominant computation of the dyadic Green's for layered media (DGLM) is still the Sommerfeld integral, but FIPWA and MF-FIPWA accelerate the computation with simplified expressions of DGLM and steepest descent path integrals.

6.3 3-D translators for layered medium

Using pilot vectors, the scalar components of DGLM have been shown to depend on two forms of the Sommerfeld integral due to TM and TE sources [19, 34]. Denoted as g^{TM} and g^{TE} , they are

$$g^{\text{TM}}(kr) = \int_{\Gamma_\beta} d\beta W^{\text{TM}}(k \cos \beta, z) H_0^{(1)}(k \sin \beta, \rho), \quad (6.1)$$

$$g^{\text{TE}}(kr) = \int_{\Gamma_\beta} d\beta W^{\text{TE}}(k \cos \beta, z) H_0^{(1)}(k \sin \beta, \rho), \quad (6.2)$$

where Γ_β is the modified steepest descent path in the θ -plane. The functions $W^{\text{TM}}(k \cos \beta, z)$ and $W^{\text{TE}}(k \cos \beta, z)$ represent a general form for the exponential function, the generalized reflection coefficient [1] and the associated constants.

Upon applying the MLFMA paradigm (receiving pattern, O2I-SM translator, radiation pattern) to factor (6.1) and (6.2), the vector-matrix-vector factorization is

$$g^\gamma(kr) = \boldsymbol{\beta}_{jJ}(\theta_s, \phi_s) \cdot \bar{\boldsymbol{\tau}}_{JI'}^\gamma(\theta_s, \phi_s) \cdot \boldsymbol{\beta}_{I'i}(\theta_s, \phi_s), \quad (6.3)$$

where γ represents TM or TE, $\boldsymbol{\beta}_{jJ}$ is a vector that represents the receiving pattern of particle j in group J , $\boldsymbol{\beta}_{I'i}$ is a vector that represents the radiation pattern of particle i in group I' , and $\bar{\boldsymbol{\tau}}_{JI'}^\gamma$ represents the O2I-SM translation matrix from group I' to group J . Note that I' denotes the image source to follow notation in previous works on FIPWA.

6.3.1 Matrix representation of the translator

In MLFMA-based algorithms, the Green's function is expanded in terms of plane-waves, and specifically in MF-FIPWA, the Green's function is expanded in terms of inhomogeneous plane waves. Furthermore, the double integral of the O2I-SM translator represents summation of a continuum of inhomogeneous plane waves that have direction $\mathbf{k}(\beta, \alpha)$. Thus, the translator is a function that is defined on a sphere in momentum, or Fourier, space. Each (β, α) pair

represents a complex direction, resulting in a complex wave vector. Yet, it is inefficient to construct the O2I matrices for all possible directions.

For efficiency, only real-valued angles (θ, ϕ) are used to represent the translator by using interpolation and extrapolation. Figure 6.1 illustrates the set of plane-wave directions as points on the sphere. Each sample on the sphere represents a translation direction $\mathbf{k}(\theta_s, \phi_s)$. Hence, (θ_s, ϕ_s) represents a plane-wave direction emanating from group I' to group J .

The difference between FIPWA and MF-FIPWA becomes evident once the O2I-SM translator in (6.3) is viewed explicitly. In the next sections, the explicit forms of the FIPWA and MF-FIPWA O2I-SM translation matrices are shown.

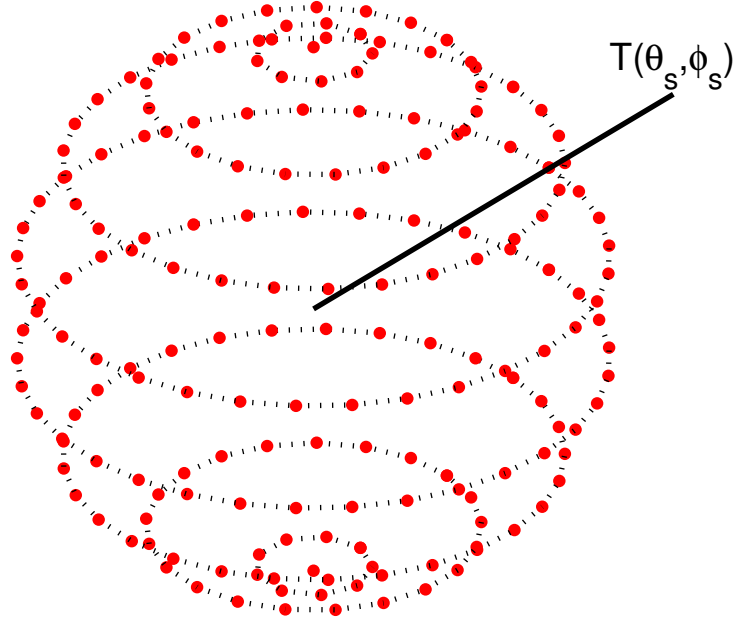


Figure 6.1: The 3-D translator is illustrated as a sampled function on the unit sphere. Each sample is a direction of translation during the matrix-vector multiply.

6.3.2 FIPWA 3-D O2I-SM translator

The 3-D O2I-SM translator for FIPWA uses 2-D FMA to expand the Hankel function in (6.1) and (6.2) as

$$\begin{aligned} \mathcal{T}_{3\text{Dfipwa}}(\theta_s, \phi_s) &= \int_{\Gamma_\beta} d\beta \int_0^{2\pi} d\alpha W(\beta) e^{i\mathbf{k}_z(\beta) \cdot \mathbf{z}_{JI'}} \\ &\quad \times \mathcal{T}_{2\text{Dfma}}(\beta, \alpha) I(\beta - \theta_s) I(\alpha - \phi_s) \end{aligned} \quad (6.4)$$

$$= \int_{\Gamma_\beta} d\beta e^{i\mathbf{k}_z(\beta) \cdot \mathbf{z}_{JI'}} \tilde{\mathcal{T}}_{2\text{Dfma}}(\beta, \phi_s), \quad (6.5)$$

where

$$\tilde{\mathcal{T}}_{2\text{Dfma}}(\beta, \phi_s) = \int_0^{2\pi} d\alpha \mathcal{T}_{2\text{Dfma}}(\beta, \alpha) I(\alpha - \phi_s), \quad (6.6)$$

$$\mathcal{T}_{2\text{Dfma}}(\beta, \alpha) = \sum_{p=-P}^P H_p^{(1)}(k \sin \beta \rho_{JI'}) e^{ip(\alpha - \phi_{JI'} + \pi/2)}, \quad (6.7)$$

$W(\beta)$ is a general expression composed of the generalized reflection coefficient and constants, $I(\beta - \theta_s)$ and $I(\alpha - \phi_s)$, are interpolation functions, and

$$\beta = \beta_R + i\beta_I,$$

$$\alpha \in (0, 2\pi),$$

$$\mathbf{r}_{JI'} = \hat{\mathbf{x}}(x_J - x_I) + \hat{\mathbf{y}}(y_J - y_I) + \hat{\mathbf{z}}(z_J + z_I),$$

$$\mathbf{k}(\beta, \alpha) = k \sin \beta (\hat{\mathbf{x}} \cos \alpha + \hat{\mathbf{y}} \sin \alpha) + k \cos \beta \hat{\mathbf{z}},$$

$$\mathbf{k}_z(\beta) = k \cos \beta \hat{\mathbf{z}}.$$

Upon discretizing the integrals, the FIPWA O2I-SM translator becomes

$$\begin{aligned} \mathcal{T}_{3\text{Dfipwa}}(\theta_s, \phi_s) &= \sum_{q_1=1}^{N_\beta} \sum_{q_2=1}^{N_{\alpha, q_1}^{\text{fipwa}}} w_{q_1} w_{q_2} W(\beta_{q_1}) e^{i\mathbf{k}_z(\beta_{q_1}) \cdot \mathbf{z}_{JI'}} \\ &\quad \times \mathcal{T}_{2\text{Dfma}}(\beta_{q_1}, \alpha_{q_2}) I(\beta_{q_1} - \theta_s) I(\alpha_{q_2} - \phi_s), \end{aligned} \quad (6.8)$$

$$\mathcal{T}_{2\text{Dfma}}(\beta_{q_1}, \alpha_{q_2}) = \sum_{p=-P_{q_1}}^{P_{q_1}} H_p^{(1)}(k \rho_{JI'}) e^{ip(\alpha_{q_2} - \phi_{JI'} - \pi/2)}, \quad (6.9)$$

where (w_{q_1}, w_{q_2}) are quadrature weights, N_β is the number of quadrature points on Γ_β , $N_{\alpha, q_1}^{\text{fipwa}}$ is the number of quadrature points in $(0, 2\pi)$, and P_{q_1} is the multipole truncation number.

6.3.3 MF-FIPWA 3-D O2I-SM translator

In MF-FIPWA, the 2-D translator is the 2-D FIPWA translator [22,23]. The 3-D MF-FIPWA O2I-SM translator is

$$\mathcal{T}_{\text{mfipwa}}(\theta_s, \phi_s) = \int_{\Gamma_\beta} d\beta W(\beta) \mathcal{T}_{\text{2dfipwa}}(\beta, \phi_s) I(\beta - \theta_s) \quad (6.10)$$

$$\mathcal{T}_{\text{2dfipwa}}(\beta, \phi_s) = \int_{\Gamma_\alpha} d\alpha e^{i\mathbf{k}(\beta, \alpha) \cdot \mathbf{r}_{JI'}} I(\alpha_{q_2} - \phi_s), \quad (6.11)$$

and the discrete form is

$$\mathcal{T}_{\text{3dmfipwa}}(\theta_s, \phi_s) = \sum_{q_1}^{N_\beta} w_{q_1} W(\beta_{q_1}) \mathcal{T}_{\text{2dfipwa}}(\beta_{q_1}, \phi_s) I(\beta_{q_1} - \theta_s) \quad (6.12)$$

$$\mathcal{T}_{\text{2dfipwa}}(\beta_{q_1}, \phi_s) = \sum_{q_2=1}^{N_{\alpha, q_1}^{\text{mfipwa}}} w_{q_2} e^{i\mathbf{k}(\beta_{q_1}, \alpha_{q_2}) \cdot \mathbf{r}_{JI'}} I(\alpha_{q_2} - \phi_s) \quad (6.13)$$

where N_β is the number of quadrature points on Γ_β , and $N_{\alpha, q_1}^{\text{mfipwa}}$ denotes the number of quadrature points on Γ_α .

6.3.4 Integration differences between FIPWA and MF-FIPWA

Outer integral $\int_{\Gamma_\beta} d\beta$

Figure 6.2 shows how the double integrals of FIPWA and MF-FIPWA differ. The outer SDP integral $\int_{\Gamma_\beta}$ is the same for both, and is shown in Figure 6.2(a). There are a fixed number of points on paths I and III. On path II the number of quadrature points scales with the size of the group, D , and the number of samples in $\{\theta : 0 < \theta_s < \pi\}$. The excess bandwidth formula $P = kD + 1.8d_0^{2/3}(kD)^{1/3}$ determines the number of samples $N_s = 2P$ needed to represent the bandwidth with d_0 digits of accuracy. Referring to (6.8), the total number of quadrature points can be expressed as $N_\beta = C_1 + C_2P$, where $C_2 = 2$ following Nyquist sampling theory, and $C_1 = 30$ is sufficient to integrate with 3–4 digits of accuracy.

FIPWA—inner integral $\int_0^{2\pi} d\alpha$

Figure 6.2(b) shows the path of integration for the inner integral of FIPWA. The path is always on the real axis in $(0, 2\pi)$, but the number of integration points is proportional to the multipole expansion number, $P_{q_1} \propto k_\rho D_{2D} = k_\rho \sqrt{\frac{2}{3}} D$, where $k_\rho = k \sin \beta_{q_1}$. To distinguish between different sets of integration points for $0 < \alpha < 2\pi$, $N_{\alpha, q_1}^{\text{fipwa}}$ denotes the number of quadrature points for each q_1 . Additionally, P_{q_1} is the 2-D bandwidth of the group, so integration on path II with the trapezoidal rule or Gauss-Legendre rule is exact when $N_{\alpha, q_1}^{\text{fipwa}} = C_{q_1} P_{q_1} = 2P_{q_1}$. Note that $2P_{q_1}$ is different from the number of samples for ϕ_s . The number of quadrature points in the double integral of FIPWA is $\sum_{q_1=1}^{C_1+2P} 2P_{q_1}$ and the integrand contains a multipole expansion of size $C_{q_1} P_{q_1} = 2P_{q_1}$.

MF-FIPWA—inner integral $\int_{\Gamma_\alpha} d\alpha$

Referring to (6.12), the integrand of MF-FIPWA is simpler than the multipole expansion of FIPWA, but the integration path in the α -plane is a steepest descent path, shown in Fig. 6.2(c). Like the SDP in the β -plane, the inner SDP integral is separated into three parts, of which two have fixed numbers of quadrature points (paths I and III), and path II has a number of points that is proportional to the 2-D bandwidth P_{q_1} . The number of quadrature points on Γ_α is $N_{\alpha, q_1}^{\text{mfipwa}} = C_{1, q_1} + C_{2, q_1} P_{q_1}$, where empirical results show that $30 < C_{1, q_1} < 80$ achieves high accuracy on paths I and III and $C_{2, q_1} \propto 2 \frac{\pi}{\alpha_0} \leq 2 \frac{\pi}{\sin^{-1}(\frac{D_{2D}}{\rho_{JJ'}})}$.

The last coefficient, C_{2, q_1} , is set according to Nyquist sampling theory so that the interval is sampled at twice the bandwidth of the integrand. This is similar to the inner integration of FIPWA, but the interval of path II is $-\pi/4 < \alpha < \pi/4$ because the interval corresponds to the field of view from one group to another with $\rho_{JJ'} \geq \sqrt{2} D_{2D}$. Thus, the spacing of the quadrature points is the same as in FIPWA, but at most, only one-fourth of the integration

points are needed. This reduces the cost to compute the integration of path II of MF-FIPWA as compared to the inner integral of FIPWA.

In addition, $C_{1,q1}$, the number of integration points along paths I and III, varies because the path must be truncated according to the size of the group and the value of k_ρ . To achieve at least single precision accuracy in the integration of paths I and III, $C_{1,q1}$ varies from 30–80, but numerical results showed that the accuracy of all 2-D translators does not need high accuracy, so in some cases $C_{1,q1} < 30$. Details of the numerical integration method are discussed in Chapter 7.

Finally, it is worthwhile to note the similarity between the integrand of 2-D FIPWA and the integrand of 3-D FIPWA for free space. In the latter, the translator is [21]

$$\mathcal{T}_{3D, \text{fipwa}}^{\text{free space}}(\theta_s, \phi_s) = \int_{\Gamma_\beta} d\beta \int_0^{2\pi} d\alpha f(\beta) I(\beta - \theta_s) I(\alpha - \phi_s) \quad (6.14)$$

$$= \int_{\Gamma_\beta} d\beta f(\beta) I(\beta - \theta_s) \int_0^{2\pi} d\alpha I(\alpha - \phi_s) \quad (6.15)$$

$$= \int_{\Gamma_\beta} d\beta f(\beta) I(\beta - \theta_s) \tau_{2D}(\phi_s), \quad (6.16)$$

where $f(\beta) = \frac{ik}{2\pi} \sin \beta$, and $\tau_{2D}(\phi_s) = \int_0^{2\pi} d\alpha I(\alpha - \phi_s)$. When $I(\alpha - \phi_s)$ is the sinc or Dirichlet interpolation function, $\tau_{2D}(\phi_s) = \frac{2\pi}{N_\phi}$, where N_ϕ is the number of sample points in the ϕ coordinate. The analytic solution of the inner integral reduces the double integral to a single integral. For free-space problems, the 3-D FIPWA O2I matrix is much faster to set up than the 3-D MLFMA O2I matrix.

However, in the layered-medium translator, the 2-D FIPWA translator in Eq. (6.11) does not have an analytic solution, requiring double numerical integration to construct the 3-D MF-FIPWA translator. Depending on the desired accuracy of the integration and optimization methods, in some cases MF-FIPWA is faster than FIPWA in setting up the O2I-SM translation matrix, and in other cases, FIPWA is faster.

6.3.5 Summary of integration differences

The discrete forms of FIPWA and MF-FIPWA revealed that the differences in the double integration are contained within the inner integration. In FIPWA, the integrand is a product of a multipole sum and an interpolation function that is integrated along the real axis in a straightforward fashion. In MF-FIPWA, the integrand is the product of a complex exponential function and either an interpolation or an extrapolation function, and the integration path is along the SDP.

In other words, the 2-D translation matrix in FIPWA has costly multipoles, simpler integration points than MF-FIPWA. Hence, in some cases the 2-D translation matrix is more expensive to compute in FIPWA than in MF-FIPWA and vice-versa. Also, FIPWA and MF-FIPWA have similar forms for the translation matrix that is constructed in the same fashion for both algorithms. The next section examines the cost to construct the O2I-SM matrix for FIPWA and MF-FIPWA.

6.4 Construction Method for O2I-SM Translator

6.4.1 Scaling with problem size

It is often desirable to consider the cost to construct the matrix in terms of the number of unknowns, N , of the surface scattering problem. This can be done for surface scattering problems because the discretization of the surface is proportional to the spatial bandwidth of the scatterer. Hence, the number of samples needed to store the translation matrix scales according to N .

For surface scattering problems, if P^2 represents the number of samples needed to construct the translation matrix, and $P^2 \propto (kD)^2$, by the excess bandwidth formula, then $N \propto P^2$. In fact, it is well known that N is related to the spatial bandwidth $(kD)^2$ [11, 38], meaning that the total construction cost scales according to N . Furthermore, in the previous

section N_β , $N_{\alpha,q_1}^{\text{fipwa}}$, $N_{\alpha,q_1}^{\text{mfipwa}}$, and P_{q_1} were shown to be proportional to P , so that the total setup cost of FIPWA and MF-FIPWA are related to N .

6.4.2 Efficient construction method

The total cost to construct the O2I-SM matrix is proportional to the number of samples stored in the translation matrix and the cost to construct a single entry. For each angular direction, the excess bandwidth formula sets the angular spacing and the number of samples in each angular coordinate.

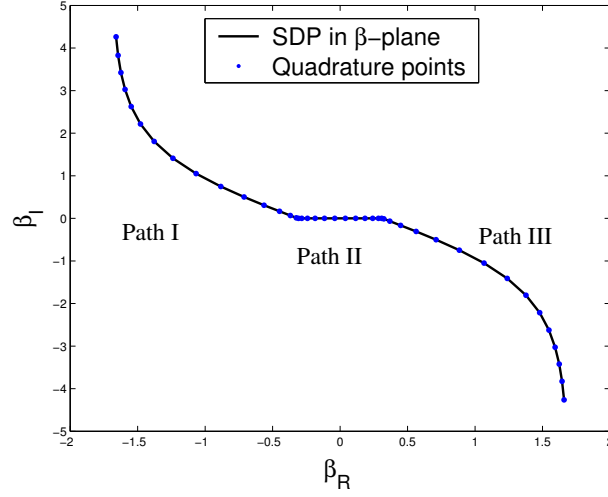
Letting N_θ equal the number of samples in the θ coordinate, and $N_{\phi,j}$ equal the number of samples in the ϕ coordinate at latitude θ_j , the construction of the FIPWA and MF-FIPWA O2I-SM translation matrix are shown to scale as $\mathcal{O}(N^{1.5})$, as $N \rightarrow \infty$.

Starting with a straightforward method to construct the matrix:

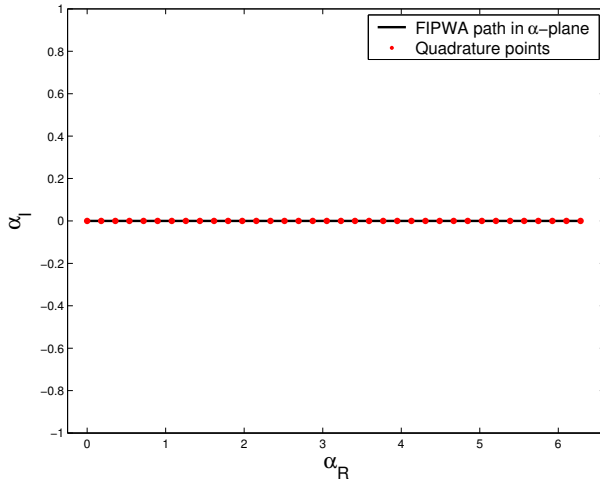
$$\sum_j^{N_\theta} \sum_k^{N_{\phi,j}} \mathcal{T}(\theta_j, \phi_{k,j}), \quad (6.17)$$

where $\mathcal{T}(\theta_j, \phi_{k,j})$ is defined by either Eq. (6.4) or (6.10), the cost is greatly reduced by oversampling the radiation patterns, and in turn, storing more samples in the translation matrix. Referring to the illustration of the translator in Fig. 6.1, fewer samples occur near the poles of the sphere. Yet, in FIPWA and MF-FIPWA, the number of samples at each latitude equals the number of samples at the equator. By fixing the number of samples, with the exception of the poles on the sphere, the translation matrix is rectangular of size $(N_\theta - 2) \times N_\phi$.

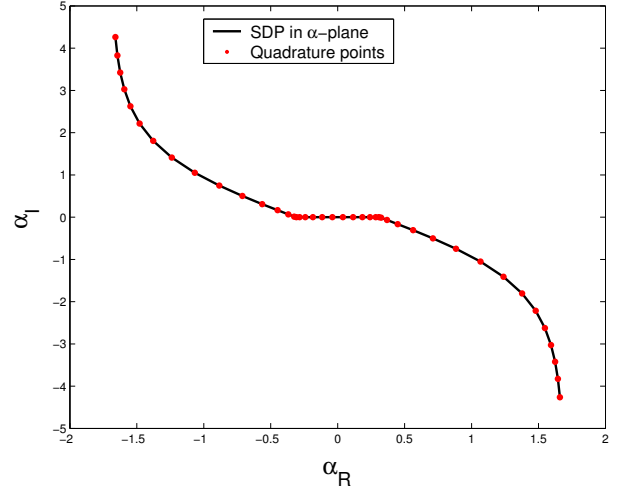
With N_ϕ constant for each latitude, and consequently for each value of β , it is efficient to compute the summations in Eqs. (6.17), (6.8) and (6.12) in two stages. This method has previously been implemented in FIPWA [21], but has not been reported. Here, the derivation follows that of [39].



(a)



(b)



(c)

Figure 6.2: Integration paths used to construct O2I-SM translator. (a) Integration path of outer integral in (6.4) and (6.10). (b) Integration path of inner integral for (6.4) (FIPWA). (c) Integration path for inner integral of (6.10) MF-FIPWA).

Given the translator in Eq. (6.4) or (6.10), it can be shown that the Green's function is equal to the sum of the stored samples in the special case of

$$\mathbf{r}_{ji} = \mathbf{r}_{jJ} + \mathbf{r}_{JI'} + \mathbf{r}_{I'i}, \quad |\mathbf{r}_{jJ}|, |\mathbf{r}_{I'i}| \rightarrow 0. \quad (6.18)$$

In other words, the plane-wave expansion of the translation matrix is equivalent to the scalar Green's function when evaluated with $r_{JI'}$.

Upon substituting Eq. (6.8) or (6.12) into (6.17), and simplifying the notation by using only the indices as arguments of the function, the expansion of the Green's function is

$$g(\mathbf{r}_{JI'}) = \sum_{j=1}^{N_\theta} \sum_{k=1}^{N_\phi} \sum_{m=1}^{N_\beta} \sum_{n=1}^{N_{\alpha,m}} W(n, m) I(n, k) I(m, j), \quad (6.19)$$

$$= \sum_{m=1}^{N_\beta} \sum_{n=1}^{N_{\alpha,m}} W(n, m) \sum_{k=1}^{N_\phi} I(n, k) \sum_{j=1}^{N_\theta} I(m, j) \quad (6.20)$$

$$= \sum_{m=1}^{N_\beta} \sum_{n=1}^{N_{\alpha,m}} W(n, m) \tilde{I}(n) \tilde{I}(m), \quad (6.21)$$

where $\tilde{I}(n) = \sum_k^{N_\phi} I(n, k)$ and $\tilde{I}(m) = \sum_j^{N_\theta} I(m, j)$.

Per the discussion in Section 6.3.4, it appears that N_β scales as $\mathcal{O}(\sqrt{N})$ because the integration of path II scales with the bandwidth. However, interpolation can be computed with only a few nearby values, i.e., local interpolation, and results in high accuracy [21]. In practice, N_β is a constant and does not scale with N .

Therefore, the translation matrix can be constructed at a cost less than

$$\text{Cost:} \approx \mathcal{O}(\max(N_\theta, N_\phi, N_p) \cdot \max(N_{\alpha,m})) \quad (6.22)$$

$$\approx \mathcal{O}(\sqrt{N} \cdot \sqrt{N}) = \mathcal{O}(N), \quad (6.23)$$

where N_θ, N_ϕ, N_p and $N_{\alpha,m}$ are all proportional to $\sqrt{N}$.

Following the above method, the general procedure to construct the O2I-SM translation matrix in FIPWA or MF-FIPWA is

1. Construct Γ_β ,
2. Construct $I(\beta - \theta_s)$ for $s = 1, \dots, N_\theta$,
3. Construct $\tau_{2D}(\beta, \phi_s)$ for $s = 1, \dots, N_\phi$,
 Construct Γ_α ,
 Construct $I(\alpha - \phi_s)$ for $s = 1, \dots, N_\phi$,
 Assemble $\tau_{2D}(\beta, \phi_s)$.
4. Assemble $\mathcal{T}_{3D}(\theta_s, \phi_s)$.

Depending on the number of integration points on Γ_α , MF-FIPWA is constructed faster than FIPWA and vice-versa for a given accuracy setting. Typically, large boxes require such a large number of multipoles and quadrature points that MF-FIPWA is computed faster at the highest levels in the MLFMA tree. On the other hand, when boxes are very small, few multipoles and integration points are needed so that FIPWA is faster than MF-FIPWA. Results of the accuracy and setup cost are shown in the next section.

6.5 Numerical Results

In this section, the accuracy and setup time of the O2I-SM translation matrices are compared. To generate comparison data, several scattering problems are set up to represent a random set of translation matrices of various sizes. The bistatic radar cross section is computed for moderate to large sized spheres that are placed above a lossy half-space.

Table 6.1 lists the problem sizes, frequencies, and number of levels for the various cases. In each case, the half-space has $\epsilon = (6.5, 0.6)$, the sphere is located 0.2 m above the half-space, and the smallest box in the tree is 0.10λ .

Table 6.1: List of scattering examples and set up parameters.

	Number of Unknowns					
	10 092	101 568	252 300	504 300	1 002 252	2 007 372
Frequency (GHz)	0.297	0.943	1.482	2.099	2.959	4.184
Maximum levels	5	6	7	8	8	9

6.5.1 Accuracy of spectral Green's function expansion

FIPWA and MF-FIPWA use the dyadic Green's function for layered media (DGLM), but the core computation is the Sommerfeld integral for TM and TE sources, denoted as g^{TM} and g^{TE} . Each scalar component of DGLM contains g^{TM} , or g^{TE} , or both. Also, the TM and TE O2I-SM translation matrices are easily shown to represent g^{TM} and g^{TE} when the radiation patterns are omnidirectional. Therefore, let g_{SIP}^{γ} represent the Sommerfeld integration path (SIP), and let $g_{\text{FIPWA}}^{\gamma}$, and $g_{\text{MFFIPWA}}^{\gamma}$ represent the FIPWA and MF-FIPWA approximations to g_{SIP}^{γ} , respectively, where γ is TM, or TE or d for direct.

To compute the error in FIPWA and MF-FIPWA, the scattering cases in Table 6.1 are used to simulate a variety of translators of different sizes and orientations. In each case, the box size is normalized by the wavelength so that translators of different problems can be compared. The error estimate of FIPWA and MF-FIPWA is based on the special case when the generalized reflection coefficients are set to unity. In this case, $g_{\text{SIP}}^{\text{d}} = g_{\text{FIPWA}}^{\text{d}} = g_{\text{MFFIPWA}}^{\text{d}} = g_0 = e^{ikr}/r$. This method is similar to calibrating a measurement system where the system error is accounted for and understood to provide a limitation to the accuracy. Using the root mean square error (RMSE), the error is determined for each box size that corresponds to a different level in the MLMFA tree. The error is

$$\text{RMSE}_a = \sqrt{\frac{1}{N_a} \sum_{n=1}^{N_a} |g_{\gamma} - g_0|^2}, \quad (6.24)$$

where N_a is the number of boxes of size a , and γ is either SIP, FIPWA, or MF-FIPWA.

Table 6.2 summarizes the error estimate for box sizes of 0.10–12.8 λ . For each box size, the translators from all cases in Table 6.1 are grouped together regardless of separation or angular orientation. Each group has N_a boxes. The Sommerfeld integral is independent of the box size and results in a nearly constant RMS error. The RMS error represents the overall integration error in computing $g_{\text{SIP}}^{\text{d}}$. FIPWA ($g_{\text{FIPWA}}^{\text{d}}$) depends on the box size and shows better accuracy than SIP for large boxes, but it loses accuracy as the box size decreases, indicating low frequency breakdown. These results establish the usable range of box sizes for FIPWA. The smallest box size should be larger than or equal to 0.10 λ .

Table 6.2: RMS error of O2I-SM translator by box size. SIP, FIPWA and MF-FIPWA are compared to $g_0 = e^{ikr}/r$. The box size a is normalized to wavelengths and N_a is the number of translators per box size.

a (λ)	N_a	g_0		
		$g_{\text{SIP}}^{\text{d}}$	$g_{\text{FIPWA}}^{\text{d}}$	$g_{\text{MFFIPWA}}^{\text{d}}$
12.80	105	7.82e-04	2.75e-06	2.37e-04
6.40	727	3.32e-05	2.23e-06	1.10e-03
3.20	1 872	3.75e-05	2.66e-05	7.93e-03
1.60	4 334	8.37e-05	1.90e-04	1.90e-04
0.80	8 873	1.03e-04	9.12e-04	9.12e-04
0.40	18 033	1.27e-04	3.23e-03	3.22e-03
0.20	36 266	1.34e-04	7.12e-03	7.06e-03
0.10	72 370	1.35e-04	4.32e-01	1.13e-02

MF-FIPWA behaves only slightly different from FIPWA. For large boxes, the accuracy is less than MF-FIPWA, and for small boxes, the accuracy is slightly better than FIPWA. The reason is because the accuracy of the 2-D translation matrix in MF-FIPWA depends directly on the number of samples used to store the translation matrix, while it does not in FIPWA. The details are deferred to the next chapter because they are related to optimization of the interpolation and extrapolation. In addition, the better accuracy for the smallest box size suggests that low frequency breakdown of MF-FIPWA occurs at smaller boxes than FIPWA.

This last point indicates that there is potential to use MF-FIPWA for even smaller boxes before low frequency breakdown occurs.

6.5.2 Accuracy of solution

The accuracy of the scattering solution is compared in two ways. First, direct comparison of the scattering solutions of FIPWA and MF-FIPWA are presented for various spheres above a lossy halfspace. Second, the number of iterations needed to achieve a residual error of 0.001 is shown to be nearly equal for all cases, implying that the accuracy of the MF-FIPWA translator is equivalent to the FIPWA translator.

Scattering results when the smallest box is $\lambda/10$.

Here, the first comparison is of the bistatic scattering solutions. Figure 6.3 compares the FIPWA and MF-FIPWA solutions of the scattering of a 1-m sphere with $N = 101\,568$ unknowns. In each algorithm, the smallest box size is $\lambda/10$ at 1.887 GHz. Excellent agreement is observed between the two solutions. As mentioned previously, when the smallest box in the MLFMA-tree is close to $\lambda/5$, the solution is more accurate. By comparing the worst case settings, the agreement is expected to be even better for larger sizes of the smallest box. Typically, the box size is set between $\lambda/5$ and $\lambda/10$.

Scattering results when the smallest box is $\lambda/5$.

To gauge the difference more quantitatively, Figure 6.4 shows the same results when the smallest box size is $\lambda/5$. The results have strong agreement for each polarization. However, to obtain a better perspective of the bistatic solution, the relative error between FIPWA and MF-FIPWA is shown in Figure 6.5, page 74. The mean error is less than 1.0% for all observation angles and demonstrates the error controllability of the multipole-free algorithm.

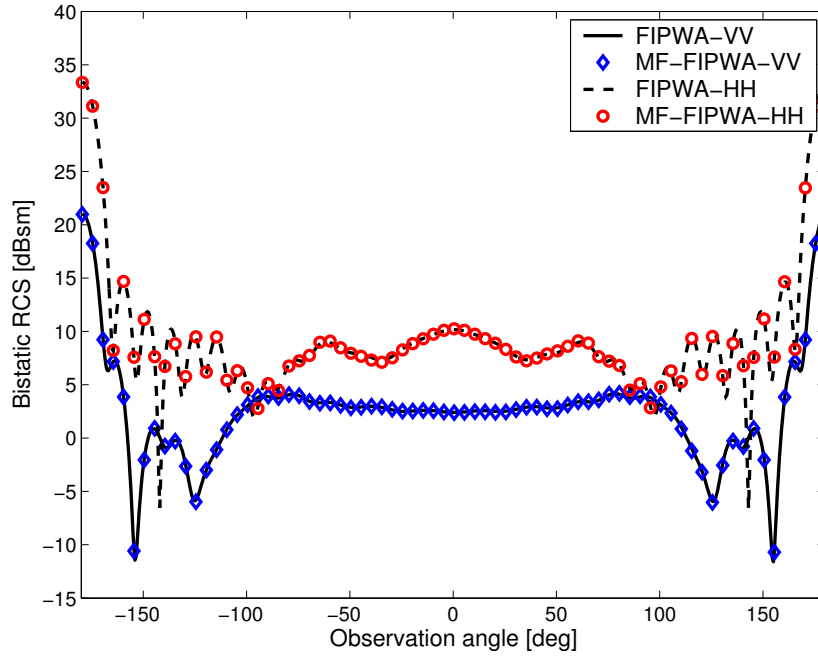


Figure 6.3: Comparison of bistatic solutions when smallest box size is $\lambda/10$. Frequency is 0.943 GHz, $N = 101\ 568$ with 6 levels, and $\epsilon = (6.5, 0.6)$.

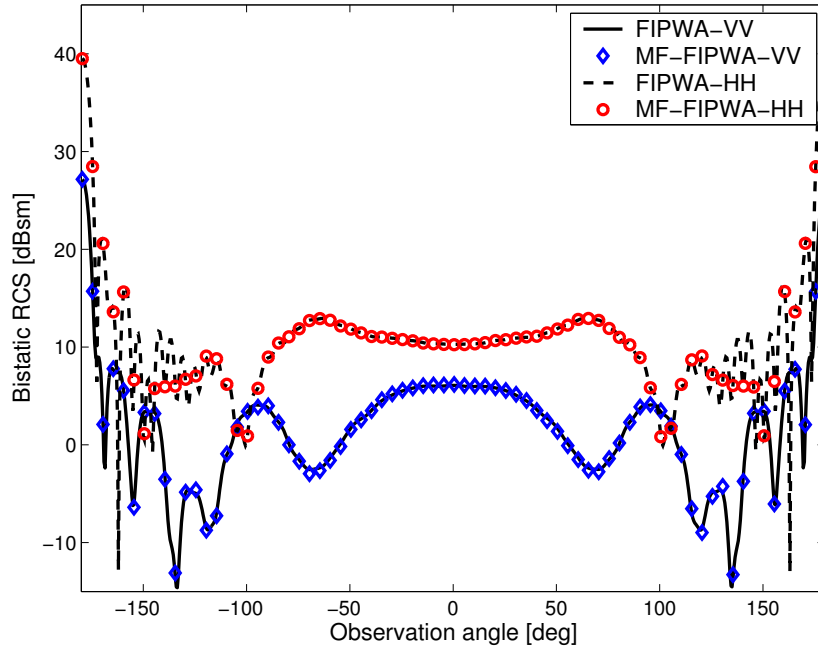


Figure 6.4: Comparison of bistatic solutions when smallest box size is $\lambda/5$. Frequency is 1.887 GHz, $N = 101\ 568$ with 6 levels, and $\epsilon = (6.5, 0.6)$.

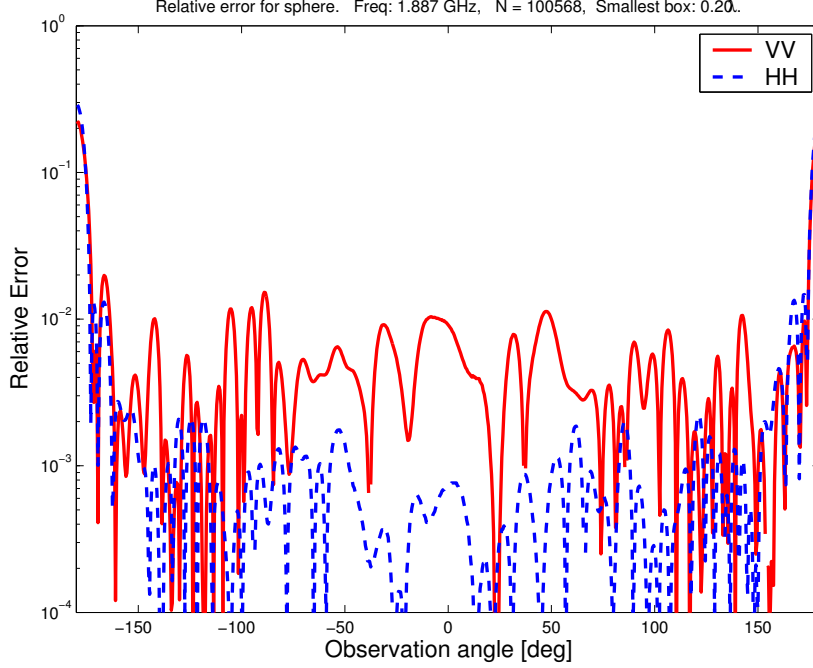


Figure 6.5: Relative error between scattering solutions of MF-FIPWA and FIPWA. The bistatic scattering results of 6.4 are used.

Scalability of error control

Ideally, the relative error between the two algorithms should not increase as the problem size increases. The full set of data is presented in Appendix B, but here, the global relative error is plotted versus the size of each problem in Figure 6.6. The global relative error is defined as

$$\text{global relative error} = \frac{\sum_{n=1}^{N_\phi} \frac{|\sigma_{\text{MF-FIPWA}}(\phi_n) - \sigma_{\text{FIPWA}}(\phi_n)|}{\sum_{n=1}^{N_\phi} \sigma_{\text{FIPWA}}(\phi_n)}}{N_\phi}, \quad (6.25)$$

where N_ϕ are the number of bistatic angles, and $\sigma_{\text{MF-FIPWA}}(\phi_n)$ is the scattering solution at the n^{th} sample of ϕ .

The global relative error remains less than 10% for all sizes, and this comparison is for the smallest box size of $\lambda/10$. In other words, the difference in the scattering solutions for FIPWA and MF-FIPWA is less than 10% for low accuracy settings. When using a larger size for the smallest box, the global relative error is less than 2%.

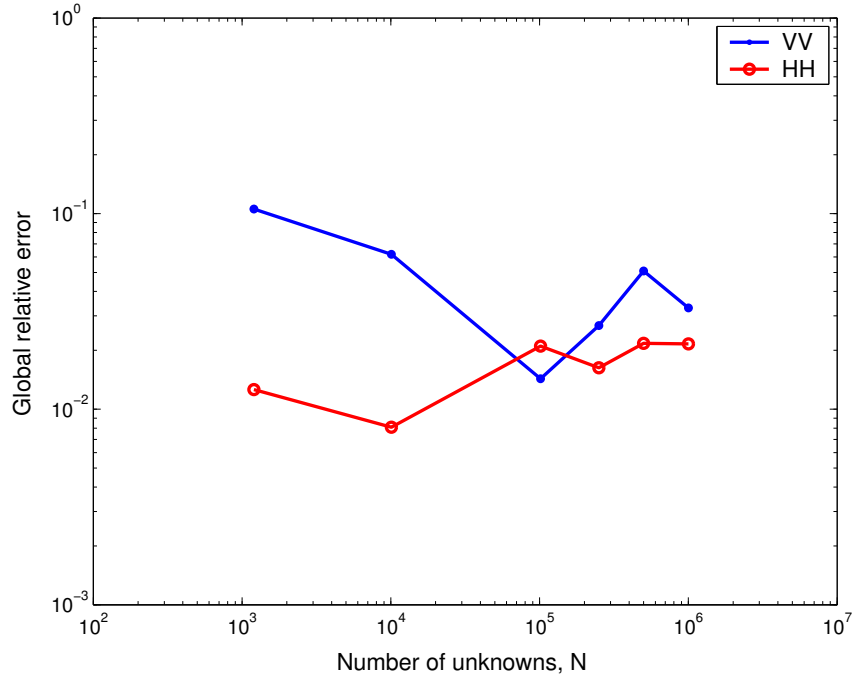


Figure 6.6: Scaling of global relative error. Scattering by spheres over a lossy half space for VV and HH polarizations.

6.5.3 Efficiency in setup and matrix solver

As shown in Section 6.4, the time to construct the complete O2I-SM translation matrix scales as $\mathcal{O}(N)$ for FIPWA and MF-FIPWA, and the memory cost scales as $\mathcal{O}(N)$.

For various cases of the spheres used in Chapter 5, Tables 6.3 and 6.4 present the time and memory usage. Table 6.3 lists the memory costs and setup times for MF-FIPWA and Table 6.4 shows the costs for FIPWA. These results have not been previously reported, so results are shown for each algorithm for several problem sizes.

In comparing results from the two tables, the total cost of the outgoing-to-incoming translators (O2I) for free space has the same memory cost, because this part of the algorithm is identical for FIPWA and MF-FIPWA. Also, the storage cost of the O2I-SM translation matrix is the same for each algorithm because the data structures are the same, resulting in equivalent costs to compute the matrix-vector (mat-vec) multiply.

Table 6.3: MF-FIPWA construction cost of O2I-SM matrices when $a = \lambda/10$. Results for various 1-m spheres.

	Number of unknowns					
	1 200	10 092	101 568	250 300	500 300	1 002 252
Frequency (GHz)	0.103	0.297	0.943	1.481	2.099	2.959
Max level	3	5	6	7	8	8
O2I (MBytes)	0.407	2.312	5.357	15.380	50.653	50.653
O2I-SM (MBytes)	1.879	11.989	89.307	212.113	421.629	782.526
O2I (CPU sec)	0.9	3.0	10.7	83	917	917
O2I-SM (CPU sec)	33	131	475	1 057	1 572	2 420
Mat-vec (CPU sec)	0.10	1.3	15	41	77	152
Iteration (CPU sec)	0.21	2.5	30	81	149	295
N_{iter} VV	33	35	39	44	52	55
N_{iter} HH	33	35	39	44	52	55
Total memory (MB)	5.8	43.7	387	962	2 002	3 847
Total run time (CPU sec)	263	1 405	1 039	1 671	122 250	247 033

Table 6.4: FIPWA construction cost of O2I-SM matrices when $a = \lambda/10$. Results for various 1-m spheres.

	Number of unknowns					
	1 200	10 092	101 568	250 300	500 300	1 002 252
Frequency (GHz)	0.103	0.297	0.943	1.481	2.099	2.959
Max level	3	5	6	7	8	8
O2I (MB)	0.423	2.312	5.357	15.380	50.653	50.653
O2I-SM (MB)	1.879	11.989	89.307	212.113	421.629	782.526
O2I (CPU sec)	0.7	3.0	10.7	84	918	900
O2I-SM (CPU sec)	5.0	21.6	90.3	220	672	1 060
Mat-vec (CPU sec)	0.11	1.3	16	41	77	152
Iteration (CPU sec)	0.21	2.5	30	81	149	295
N_{iter} VV	33	35	39	44	52	55
N_{iter} HH	33	35	39	44	51	55
Total memory (MB)	5.8	43.7	387	962	2 002	3 847
Total run time (CPU sec)	234	1 290	619	949	121 253	244 049

The differences occur in the setup and total run times for the O2I-SM translator because the 2-D translation matrix has higher integration cost in MF-FIPWA than FIPWA. Optimization can be performed, but that is deferred to the next chapter.

Tables 6.5 and 6.6 list similar results when the smallest box size is $a = 0.2\lambda$. From MLFMA, it is known that larger sizes of the smallest box result in better accuracy in the solution. However, the larger size requires more memory at the leafy level, the dominant cost of memory and computation during the setup stage. Hence, only values of N up to 250 000 unknowns were used in Table 6.5 to illustrate the cost.

Table 6.5: MF-FIPWA construction cost of O2I-SM matrices when $a = \lambda/5$. Results for various 1-m spheres.

	Number of unknowns			
	1 200	10 092	101 568	250 300
Frequency (GHz)	0.206	0.594	1.887	2.963
Max level	3	5	6	7
O2I (MB)	0.977	5.251	15.275	50.539
O2I-SM (MB)	3.397	19.252	130.57	320.274
O2I (CPU sec)	1.1	11	84	910
O2I-SM (CPU sec)	40	156	722	2 094
Mat-vec (CPU sec)	0.22	2.5	59	85
Iteration (CPU sec)	0.43	4.9	31	162
N_{iter} VV	17	25	39	43
N_{iter} HH	17	25	38	42
Total memory (MB)	11.7	86.88	770	1 949
Total run time (CPU sec)	266	2 096	22 476	59 674

Again, the cost to store the translators, and the time to perform the matrix-vector multiply are the same. Only the setup time is compared. For $N = 1\,200$, the O2I-SM translator for MF-FIPWA takes 6.6 times longer to set up than the FIPWA translator. For $N = 1\,002\,252$, the set-up time for MF-FIPWA takes 2.3 times longer than FIPWA.

Table 6.6: FIPWA construction cost of O2I-SM matrices when $a = \lambda/5$. Results for various 1-m spheres.

	Number of unknowns			
	1 200	10 092	101 568	250 300
Frequency (GHz)	0.206	0.594	1.887	2.963
Max level	3	5	6	7
O2I (MB)	0.977	5.251	15.275	50.539
O2I-SM (MB)	3.397	19.252	130.57	320.274
O2I (CPU sec)	1.1	11	84	913
O2I-SM (CPU sec)	10	50	273	1 010
Mat-vec (CPU sec)	0.23	2.6	60	84
Iteration (CPU sec)	0.43	4.9	30	163
N_{iter} VV	17	25	38	43
N_{iter} HH	17	25	38	42
Total memory (MB)	11.7	86.88	770	1 949
Total run time (CPU sec)	236	1 997	22 041	58 814

Scaling of setup time by box size

To illustrate how the algorithms compare in setting up the O2I-SM translation matrix, Figure 6.7 shows the mean time in CPU seconds needed to construct the O2I-SM translation matrix per box size. Box sizes range from $0.1\text{--}12.8\lambda$, where the largest box size is based on a two-million unknown sphere with 9 levels. MF-FIPWA (solid line with diamond) has lower setup time than FIPWA (solid line with circles) when boxes are larger than 8.0λ . Also, note that the curves for MF-FIPWA and FIPWA have shallow slopes when the boxes are small. In these cases, the SDP integration on paths I and III of Γ_β uses a fixed number of points. As the box size increases, the integration along path II dominates the computation and the slopes of the curves increase in proportion to the box size.

The hashed line represents the setup time for MF-FIPWA with single precision accuracy in the 2-D integration. MF-FIPWA still has longer setup time for the smallest boxes, but it is faster when the box size is greater than 2.0λ . These results show that one could use

MF-FIPWA at the top levels of the tree and FIPWA at the lower levels of the tree to slightly improve the O2I-SM matrix setup time. Finally, note that the setup times scales with N . The dotted line is proportional to the box edge size a which is proportional to N .

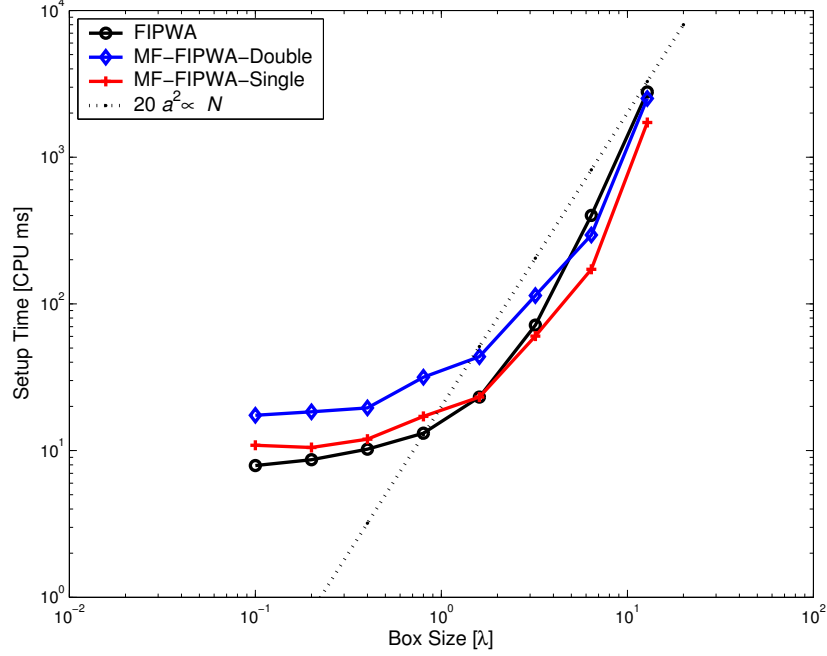


Figure 6.7: Setup time for O2I-SM translation matrix by box size. Boxes are set according to problem sizes from $N = 10\,092$ to $N = 2\,007\,372$ and range from 0.1λ to 12.8λ in size.

Memory storage cost

Besides the setup time, the storage cost is also evaluated. Figure 6.8 illustrates the equivalent storage cost of FIPWA and MF-FIPWA. As expected, the storage scales as $\mathcal{O}(N)$.

6.5.4 Mixed-form algorithm

As in mixed-form FMA, both FIPWA and MF-FIPWA can be used to reduce construction time of the O2I-SM translation matrices. Using the results of Figure 6.7, the 2-million unknown problem was rerun using a mixed-form method where MF-FIPWA computed the O2I-SM matrix when the box edge was larger than 2.0λ . Otherwise, FIPWA created the matrix. The mixed-form showed a speedup of 17% over the original FIPWA.

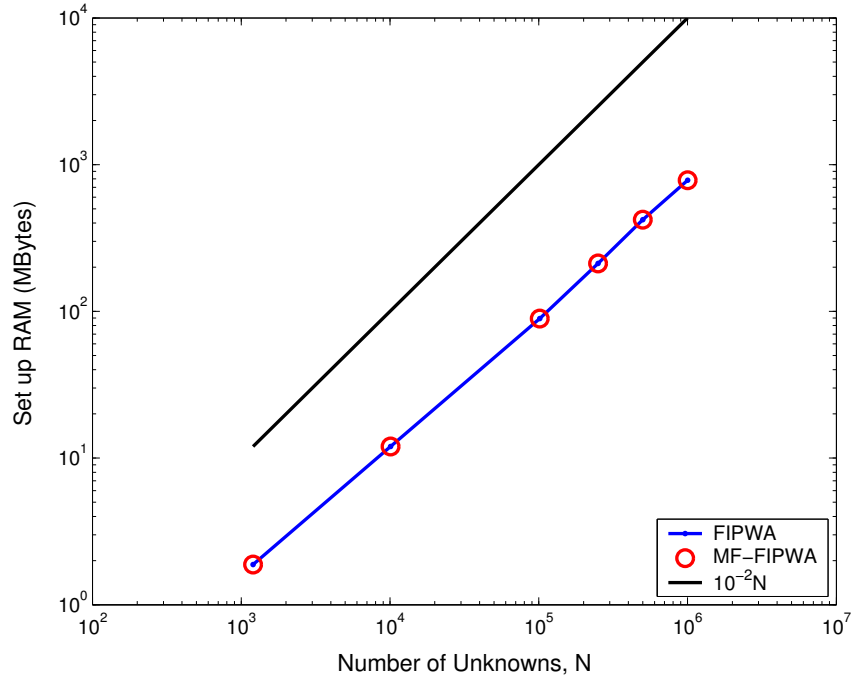


Figure 6.8: Comparison of memory cost when the smallest box size is $\lambda/10$.

6.6 Conclusions on MF-FIPWA Versus FIPWA

The computational cost of MF-FIPWA equals FIPWA and scale as $\mathcal{O}(N \log N)$. In addition, the accuracy of the plane-wave approximations to the spectral Green's function compares well to the Sommerfeld integral, but has a limited range of use. The smallest box in FIPWA and MF-FIPWA should be at larger than or equal to 0.10λ , and under these conditions the overall accuracy of MF-FIPWA for the bistatic scattering solution is controllable and agreeable to FIPWA.

Without using optimization, the time to set up the O2I-SM translation matrices showed that FIPWA is faster for small boxes (low levels in the tree), and MF-FIPWA is faster for the top levels of the tree. To capitalize on both algorithms, a mixed form solution was shown to provide 17% savings in construction time.

CHAPTER 7

IMPLEMENTATION AND OPTIMIZATION

7.1 Introduction

With any code development, it is important to pass on methods and motivating factors, so that others may repair, modify, improve, or simply understand how the code works. There is a short user's manual for MF-FIPWA [40], but this chapter provides implementation details on the data structures, numerical integration, and interpolation. These details are applicable to other fast algorithms, especially those that use the MLFMA paradigm.

Besides clarifying these aspects of MF-FIPWA, which is mainly a research code, areas of optimization are provided so that MF-FIPWA may be transformed into an application code. To facilitate such an effort, debugging subroutines have been modularized and the advanced user is given more debugging options that are transparent to the casual user. To speed up MF-FIPWA, local error control is described with optimization.

This chapter is organized as follows. First, the architecture of the fast algorithm is presented to highlight where the new multipole-free translator is incorporated into FIPWA. Second, the data structures of the translators are discussed, and third, specific details of numerical integration and interpolation are presented. Optimization is discussed in the pertinent sections. Finally, special attention is given to applied error control.

7.2 Architecture of the Fast Algorithm

The original idea of a purely plane-wave-based algorithm came from Professor Chew's group during the development of FASDPA [24], and again, during development of FIPWA for layered media [21]. Hence, the approach to MF-FIPWA was to develop error control for the 2-D FIPWA translator in complex media, followed by a simple substitution of the 2-D FIPWA translator for the 2-D FMA translator. As seen in Chapter 6, the benefit of creating a black-box translator is the ability to retain the memory usage and solve time of $\mathcal{O}(N \log N)$. However, to make such a substitution, and provide means for error control, one must understand all aspects of the architecture of the fast algorithm.

A general diagram for implementing the fast algorithm is shown in Fig. 7.1. There are three stages to the fast algorithm:

1. Setup stage – create tree, store translation and interpolation matrices, store radiation patterns and direct interaction matrices.
2. Solve stage – apply matrix preconditioner, run iterative matrix solver.
3. Postprocessing stage - compute the fields, radar cross section, radiation pattern, input impedance, or current distribution, and write output files.

The objective of MF-FIPWA was to change the 3-D stratified medium translation, or O2I-SM, matrix in the set up stage. Figure 7.2, page 84, shows the primary components used to set up the 3-D translation matrix for FIPWA and MF-FIPWA. Also shown is the special case where the translation direction is along $\hat{\mathbf{z}}$, and is multipole-free as discussed in [21]. Whenever the translation direction is along the z axis, the special case is used in lieu of FIPWA or MF-FIPWA. Once the type of translator is selected, there are two steps: initialize the interpolation and quadrature parameters, and generate the 3-D translation matrix.

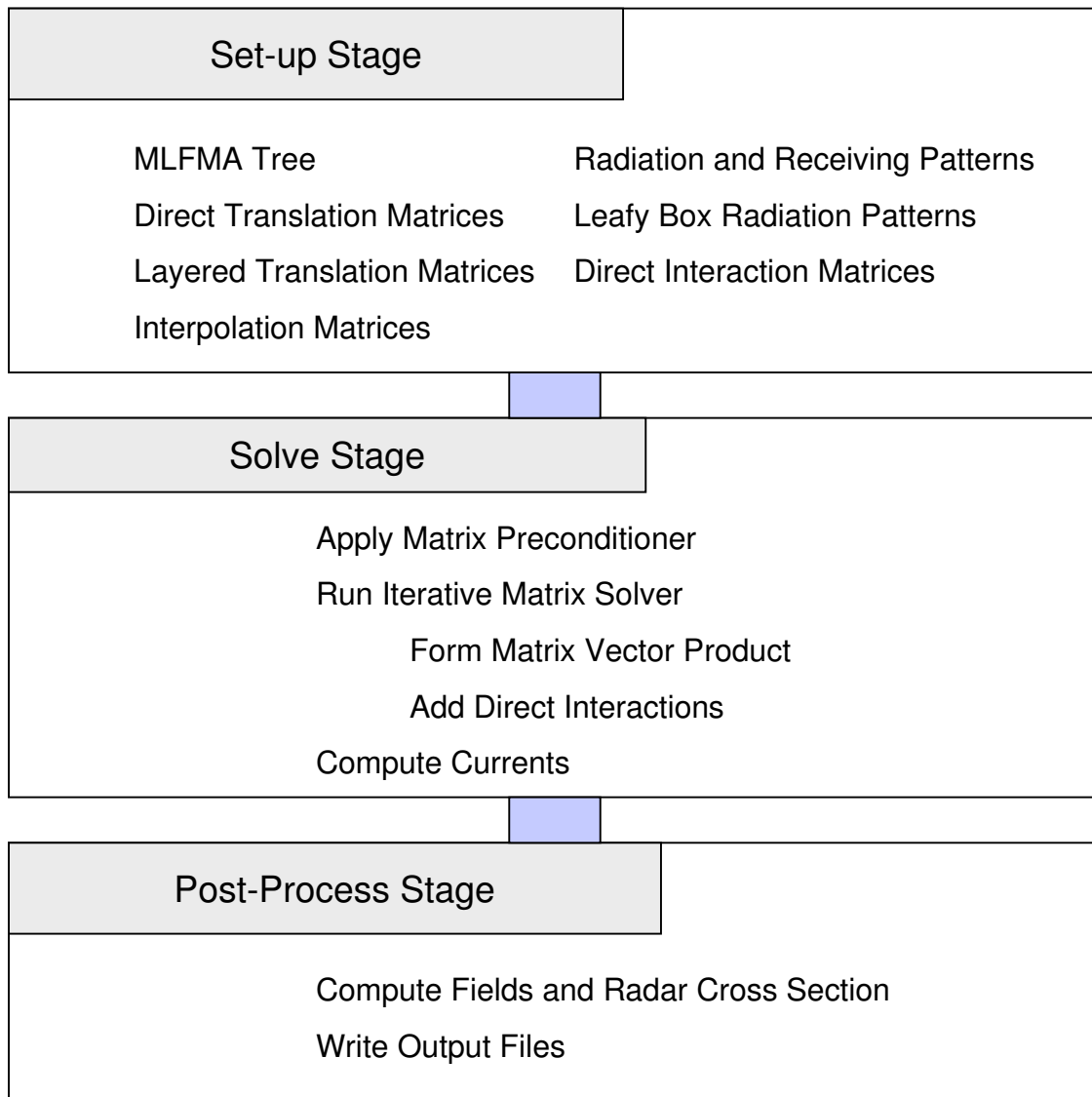


Figure 7.1: Top level diagram of MF-FIPWA under the MLFMA paradigm.

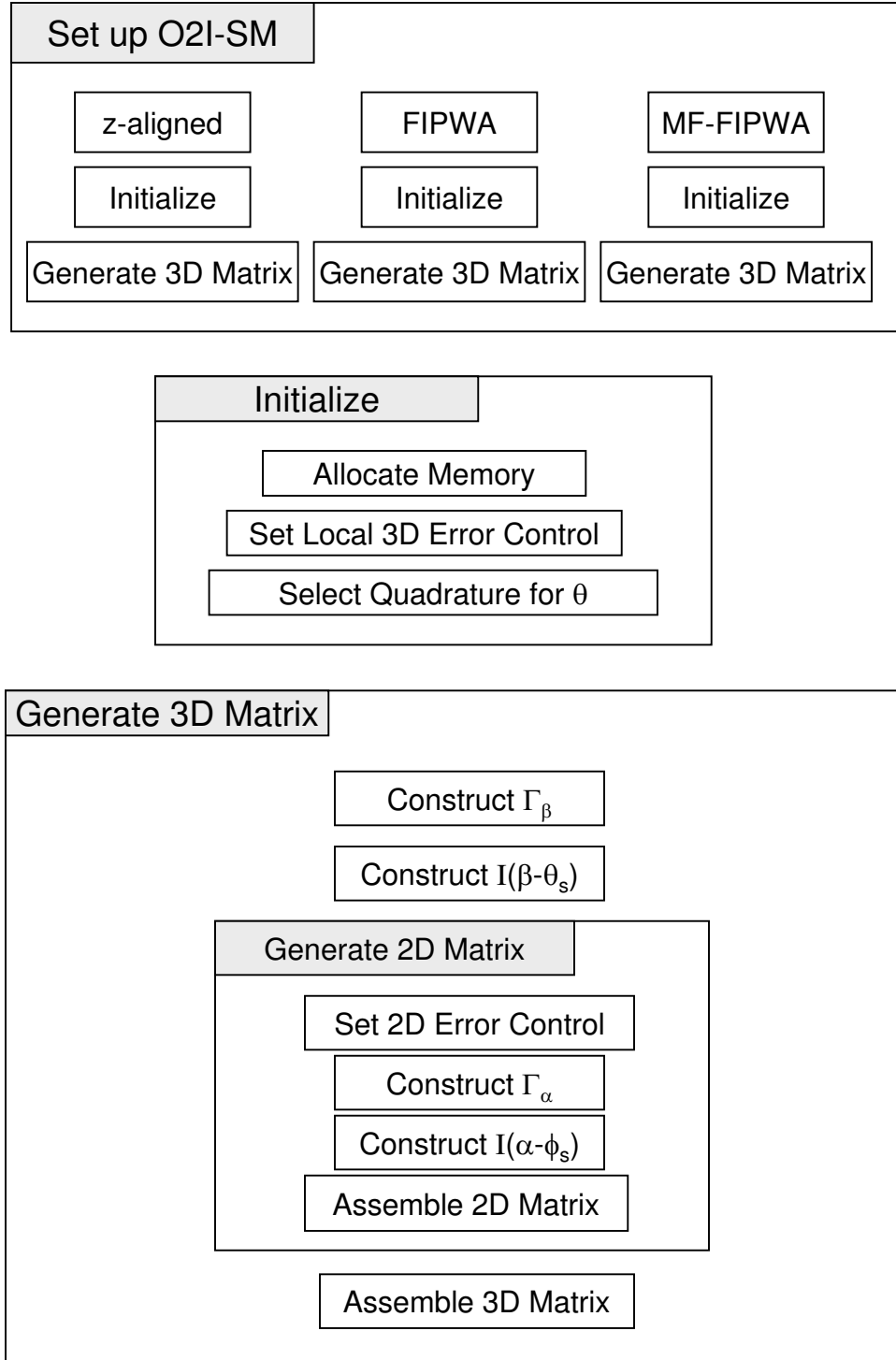


Figure 7.2: Setting up the O2I-layer translation matrix for FIPWA and MF-FIPWA.

Regardless of the type of translator, the same steps are used:

1. **Initialize** – set the 3-D error control for interpolation and extrapolation on Γ_β for the specific function and compute the Gauss-Laguerre quadrature points.
2. **Generate 3D Matrix** – compute the SDP Γ_β , the interpolation and extrapolation coefficients, and the 2-D translation matrix, followed by assembly of the 3-D translation matrix.

Initial efforts were spent creating the parallel branch for the multipole-free translator (shown in Fig. 7.2), so that the code could be compiled as FIPWA or MF-FIPWA. Later, the need to recompile the code was removed, and instead, the input file was adapted to allow the user to select the FIPWA, MF-FIPWA, or mixed-form translator at run time. Hence, MF-FIPWA now implies either the original 3-D FIPWA for layered media or the new multipole-free algorithm. For clarification in this chapter, FIPWA refers to the 3-D FIPWA translation matrix, and MF-FIPWA refers to the 3-D MF-FIPWA translation matrix and the simulation program.

The subroutines **Generate 3D Matrix** and **Generate 2D Matrix** of Fig. 7.2, are similar in design. First, construct the SDP, followed by the interpolation and extrapolation coefficients. Second, the matrix is assembled. The main difference is in the error control. For the outer integration, the error control is set one time, but the inner integral changes with every quadrature point of the outer integral. Hence, the error control for the 2-D matrix must be set with every call to **Generate 2D Matrix**. This error control is modularized in MF-FIPWA, so that it can be improved as new techniques develop. It also serves as a model for designing translator classes in object oriented programming.

In consideration of future work, it is also important to note that the data structure used to store the translation matrix is arbitrary, but if it is not designed for use in the subroutine, or module, that computes the matrix-vector product, then added computational cost is incurred via interface routines. The specific data structures used in MF-FIPWA are discussed in the next section.

7.3 Data Structures

Data structures in themselves are not of particular interest, but when using mixed programming languages and large scale codes, they need to be considered early in the design. Yet, research codes are often developed by several people over several years, and there is a tendency to write interface routines that federalize old and new codes. Hence, it is important to design any new code so that different modules are interoperable and easily modified. There is a large effort in Professor Chew's research group to incorporate new developments into a programming library. The library follows the C++ model of object oriented programming, but it is helpful to understand the data structures of existing programs to merge, or rewrite them into the library. This section discusses some aspects of the core data structures in MF-FIPWA so that future researchers may add or adapt components of MF-FIPWA to the FMA library.

7.3.1 Translation matrices

Of all the data structures used in MLFMA, and consequently, MF-FIPWA, the structures of the leafy box radiation pattern and 3-D translation matrix are the most important. These two components constitute more than 60% of the total cost of memory and computation. Therefore, it is important to reduce copying or reordering of these matrices with interface routines.

In MF-FIPWA, the translation matrices are stored according to the radiation patterns of the leafy boxes. In cases where the interpolation matrices and matrix-vector (mat-vec) multiply routines are stored, or computed with different formats, the translation matrices are reordered to comply with the other routines.

The desired storage for the mat-vec routine is

$$\mathcal{T}_{JI}^{\text{mat-vec}} = \begin{bmatrix} \mathcal{T}_{JI}(\theta_1 = \pi, 0) \\ \mathcal{T}_{JI}(\theta_2 = \pi - d\theta, \phi_1) \\ \mathcal{T}_{JI}(\theta_2, \phi_2) \\ \vdots \\ \mathcal{T}_{JI}(\theta_2, \phi_{N_\phi}) \\ \mathcal{T}_{JI}(\theta_3, \phi_1) \\ \vdots \\ \mathcal{T}_{JI}(\theta_{N_\theta-1} = d\theta, \phi_1) \\ \vdots \\ \mathcal{T}_{JI}(\theta_{N_\theta-1}, \phi_{N_\phi}) \\ \mathcal{T}_{JI}(\theta_{N_\theta} = 0, 0) \end{bmatrix}. \quad (7.1)$$

Yet, the translation matrices are created with the format

$$\mathcal{T}_{JI}^{\text{creation}} = \begin{bmatrix} \mathcal{T}_{JI}(\theta_1 = 0, 0) \\ \mathcal{T}_{JI}(\theta_2 = d\theta, \phi_1) \\ \mathcal{T}_{JI}(\theta_2, \phi_2) \\ \vdots \\ \mathcal{T}_{JI}(\theta_2, \phi_{N_\phi}) \\ \mathcal{T}_{JI}(\theta_3, \phi_1) \\ \vdots \\ \mathcal{T}_{JI}(\theta_{N_\theta-1} = \pi - d\theta, \phi_1) \\ \vdots \\ \mathcal{T}_{JI}(\theta_{N_\theta-1}, \phi_{N_\phi}) \\ \mathcal{T}_{JI}(\theta_{N_\theta} = \pi, 0) \end{bmatrix}. \quad (7.2)$$

This difference is a legacy of the original FIPWA code that requires use of an interface routine to convert from the creation format to the mat-vec format. As seen in (7.1) and (7.2), only the θ values are reversed, so it is straightforward to reorder the matrices. While it will not make the code tremendously faster, storing the translator in the mat-vec sense simplifies the code and facilitates easier optimization of various subroutines. Some of the subroutines are for debugging, reporting memory and CPU usage, or new features of the algorithm, such as revised error control methods.

7.3.2 Error control modules

The error control for MF-FIPWA is different from MLFMA in the sense that one must use a localized form of error control for MF-FIPWA. The error control modules are shown in blocks `Initialize` and `Generate 2D Matrix` of Fig. 7.2. For each translation matrix, one must consider the interlevel interpolation that occurs in the matrix-vector product. The error control for the interlevel interpolation sets the number of samples according to the bandwidth of the group size, and is determined before even allocating memory for the translation matrix.

In MF-FIOWA, there is also interpolation and extrapolation to the SDPs with real-valued samples, and the number of samples is determined according to the bandwidth of the 3-D or 2-D box. Recall that the inner integral depends on the outer integral. Hence, each call to `Generate 2-D Matrix` requires localized error control based on $k_\rho D$.

Yet, the interpolation and extrapolation functions may also require additional control measures. For example, the sinc interpolation function used in FIPWA can be computed with N_s samples, but extrapolation may need to consider the excess bandwidth, as discussed in [21]. For this additional reason, a local error control is used.

These modules for error control should be incorporated so that they are easily updated or replaced as the technology develops. In terms of a C++ class structure, an error control module would constitute a class method or member function. It should have access to those parameters that are unique to the specific translator.

7.3.3 Proposed translator class for the FMA library

Based on the development of Chapters 3 and 4, and following an object-oriented paradigm, the translator should be programmed with a constructor that creates an instance of the translation matrix, a member function to allocate memory for the number of samples, a member function for error control of the interpolation and extrapolation routines, a member

```

class Translator {
public:
    Translator();           // constructor
    ~Translator();         // destructor
    void Allocate();        // allocate memory
    void ErrorControl();    // set local error control
    void InterpolateFunc(); // interpolation function
    void ExtrapolationFunc(); // extrapolation function
    void Quadrature();      // compute quadrature
    void ContourMap();      // mapping function for SDP
    void TranslateFunc();   // core function in translator
    void Assemble();        // assemble translator
}

```

Figure 7.3: Proposed translator class for FMA library. The above is based on current research of error control and efforts to modify existing translators.

function to compute the quadrature points, a member function to map quadrature points to the SDP, a member function that constitutes the core function in the translator ($e^{i\mathbf{k}(\beta,\alpha)\cdot\mathbf{r}_{JJ'}}$ in the case of MF-FIPWA), and a member function to assemble the translation-matrix elements.

This last function would allow one to invent new implementations for constructing the translation matrix without having to rewrite an entire subroutine or class. In the C++ language, the translator could be written for MLFMA, FIPWA, and MF-FIPWA translators as shown in Figure 7.3.

In the fast multipole algorithm (FMA) library of Prof. Chew's group, there is a class for constructing assorted translators. The translator class sets up various 2-D and 3-D data structures for storing the matrices according to the matrix-vector product routines. Upon selecting the desired translator, such as FMA, FIPWA, or others, the appropriate subroutines are called to generate the translator. The class is written in such a way that a new translator can be added to the library by writing an interface routine to existing code. However, the class is not written for using inheritance. Inheritance is where a generic class is constructed,

and specific instances can be tailored by changing or adding member functions. Without using inheritance, new translators must be completely rewritten, or interface routines must be used to call the existing code. This makes it difficult to add translators without having to rewrite an entire class, and to update methods based on new research. It would be a worthwhile investment to upgrade these classes in the FMA library.

7.4 Numerical Methods

To construct the translator to arbitrary accuracy requires specialized integration routines and interpolation functions. In this section, the specific details are described.

7.4.1 Numerical integration

Computing the numerical integration of the SDP in the 2-D FIPWA translator in complex media was demonstrated in [20], but when the 2-D FIPWA translator was used to construct the 3-D MF-FIPWA translator, it was not straightforward to achieve 15 digits of accuracy. In this section, Gauss-Laguerre integration is used to achieve the desired accuracy. By controlling the accuracy of the inner integral, MF-FIPWA can be made more efficient than FIPWA.

Quadrature on path II of the SDP

On path II of Γ_β and Γ_α , Gauss-Legendre quadrature achieves high accuracy in the integration. The path is in the interval $(-\pi/4, \pi/4)$ and the points on the path represent propagating waves. Therefore, the number of points should be proportional to the bandwidth. Yet, the path does not extend the full range of $(0, 2\pi)$, so the number of points is set in proportion to the ratio of the length of path II to the full range. For example, if the bandwidth requires 24 points in the interval $(0, 2\pi)$, and the length of path II is $\pi/2$, the number of quadrature points is proportionally set to 6.

Windowed Gauss-Laguerre integration

Given the integration of the function $F(x)$, where $F(x)$ has the form $f(x)e^{-x}$ on the interval $(0, \infty)$, Gauss-Laguerre quadrature is preferred when $f(x)$ can be approximated with Laguerre polynomials. In the case of MF-FIPWA, the paths of integration on paths I and III are semi-infinite and the integrand has exponential decay. The integrand $F(\alpha) = e^{ik_\rho \rho \cos \alpha}$ is not in the exact form, but is transformed by mapping to α with the variable s as $F(s) = [e^{-s^2 + ik_\rho \rho} J(s) e^s] e^{-s}$, $s \in (0, \infty)$, where $J(s) = \frac{\partial \alpha}{\partial s}$.

As discussed in [29], the integrand of the SDP integral decays exponentially fast along the integration path, but extrapolation in FIPWA becomes unstable when the path extends deep into the complex plane. Thus, it is best to truncate the path so that $s \in (0, s_{\max})$.

The integration can be written as

$$I = \int_0^\infty ds F(s), \quad (7.3)$$

$$\approx \int_0^{s_{\max}} ds F(s) W(s), \quad (7.4)$$

$$\approx \sum_{q=1}^{N_q} w_q F(s_q) W(s_q), \quad (7.5)$$

where N_q is the number of integration points, and

$$W(s) = \begin{cases} 1, & s \leq s_{\max} \\ 0, & s > s_{\max} \end{cases}. \quad (7.6)$$

If the integrand decays sufficiently before $s = s_{\max}$, then the truncation has negligible effect.

Scaled and windowed Gauss-Laguerre quadrature

Although the integrand is windowed, α may still extend too far into the complex plane after mapping with s because the quadrature points have a large range of values. For example, consider the 15-point rule shown in Table 7.1.

Typically in the SDP integration, $s_{\max} < 10$, so s_{15} is much larger than necessary, and there are too few points in the range of interest. One approach has been to select a very

high number of points such that at least 15 points occur in $(0, s_{\max})$, and then the integrand is windowed as in (7.4). This method achieves only a few digits of precision. It can be used in the outer integral of the 3-D translator for efficiency, but not for the inner integral where high accuracy is needed.

A similar approach uses N points for $s \in (0, \infty)$, and then the points are scaled according to $s_{\max}$. Here, s_N scales according to the maximum path length, i.e., $s_N = s_{\max}$. The weights are also scaled so that the integral becomes

$$I \approx \int_0^{s_{\max}} ds \frac{1}{\kappa} F(s/\kappa) W(s/\kappa), \quad (7.7)$$

$$\approx \sum_{q=1}^N \frac{1}{\kappa} w_q F(s_q/\kappa) W(s_q/\kappa), \quad (7.8)$$

where $\kappa = s_N/s_{\max}$.

Table 7.1: Gauss-Laguerre quadrature with 15 points.

q	Nodes	Weights
1	9.331e-002	2.182e-001
2	4.927e-001	3.422e-001
3	1.216e+000	2.630e-001
4	2.270e+000	1.264e-001
5	3.668e+000	4.021e-002
6	5.425e+000	8.564e-003
7	7.566e+000	1.212e-003
8	1.012e+001	1.117e-004
9	1.313e+001	6.460e-006
10	1.665e+001	2.226e-007
11	2.078e+001	4.227e-009
12	2.562e+001	3.922e-011
13	3.141e+001	1.457e-013
14	3.853e+001	1.483e-016
15	4.803e+001	1.601e-020

This method of scaling results in high accuracy and stability because the path is sufficiently sampled by the quadrature rule in the shallow region of the SDP. Comparison of various translators reveal the minimum number of points needed to achieve single and double precision accuracy. In each case, the translation distance is two box widths, also referred to as one buffer box separation between groups. Note that 2-D FMA achieves double precision accuracy for most cases, so its accuracy is not shown.

Accuracy for large groups

A group is considered large when it spans one wavelength. Large groups have a large bandwidth, requiring more multipoles and integration points in 2-D FMA. It has been shown that the number of points N_q is proportional to $|k_\rho D_{2D}|$. Similarly, path II in 2-D FIPWA should have $\frac{1}{4}N_q$ points as it spans only one fourth the interval. Path II is integrated with the Gauss-Legendre rule to high accuracy. Yet, paths I and III are integrated with the scaled and windowed Gauss-Laguerre rule.

To determine the minimum number of points, results are shown in Tables 7.2–7.4 for lossless, and complex k_ρ . Table 7.2 lists results for various sizes of k_ρ and loss tangent $\tan \delta = \frac{\mathcal{I}m\{k_\rho\}}{\mathcal{R}e\{k_\rho\}} = 0$. It is clear that 15 quadrature points ensure single precision accuracy. Note that as $|k_\rho D_{2D}|$ decreases, the excess bandwidth must be used to increase the number of points on path II. However, various simulations showed that MF-FIPWA can be computed with only single precision accuracy in the 2-D FIPWA translator and still achieves excellent agreement with 3-D FIPWA.

Tables 7.3 and 7.4 list the accuracy for cases when $\tan \delta = \pm \frac{\pi}{3}$, respectively. In these cases, single precision accuracy is achieved with only 15 points. Therefore, large groups can be integrated to 7 digits of accuracy with up to 15 points. By optimizing for k_ρ and D_{2D} , fewer points can be used to expedite setup at higher levels in the MLFMA tree.

Table 7.2: Integration error of large boxes in 2-D FIPWA for $\tan \delta = 0$. Gauss-Laguerre integration is used on paths I and III, and Gauss-Legendre integration is used on path II.

$ k_\rho D_{2D} $	N_q						
	5	9	15	20	30	40	50
5	2.8e-003	2.4e-005	1.3e-007	2.8e-008	3.0e-008	3.0e-008	3.0e-008
10	3.6e-003	1.4e-006	5.4e-008	5.1e-008	5.1e-008	5.1e-008	5.1e-008
20	6.7e-003	1.4e-006	2.7e-011	1.3e-012	1.0e-012	1.0e-012	1.0e-012
30	1.0e-002	3.1e-006	5.3e-012	4.6e-015	1.1e-015	1.8e-015	2.3e-015
50	1.7e-002	9.5e-006	1.3e-011	3.5e-015	2.8e-015	3.1e-015	3.4e-015

Table 7.3: Integration error of large boxes in 2-D FIPWA for $\tan \delta = \frac{\pi}{6}$. Gauss-Laguerre integration is used on paths I and III, and Gauss-Legendre integration is used on path II.

$ k_\rho D_{2D} $	N_q						
	5	9	15	20	30	40	50
5	2.2e-002	1.6e-004	5.4e-007	3.1e-008	1.9e-008	1.9e-008	1.9e-008
10	5.0e-002	1.2e-004	2.2e-007	5.2e-008	4.8e-008	4.8e-008	4.8e-008
20	2.8e-001	3.7e-004	1.4e-007	2.4e-009	2.6e-011	3.9e-011	5.5e-011
30	1.3e+000	1.5e-003	3.8e-007	6.7e-009	2.1e-010	3.1e-010	4.8e-010
50	1.3e+001	1.6e-002	2.7e-005	3.7e-006	1.6e-006	1.6e-006	1.8e-006

Table 7.4: Integration error of large boxes in 2-D FIPWA for $\tan \delta = -\frac{\pi}{6}$. Gauss-Laguerre integration is used on paths I and III, and Gauss-Legendre integration is used on path II.

$ k_\rho D_{2D} $	N_q						
	5	9	15	20	30	40	50
5	5.4e-005	6.4e-006	6.2e-008	3.6e-008	3.6e-008	3.6e-008	3.6e-008
10	2.6e-004	4.7e-008	4.5e-008	4.4e-008	4.4e-008	4.4e-008	4.4e-008
20	1.7e-004	8.3e-009	6.7e-013	6.7e-013	6.6e-013	7.0e-013	6.6e-013
30	8.0e-005	7.9e-009	2.2e-015	2.0e-015	1.4e-015	1.1e-014	4.5e-015
50	1.1e-005	3.5e-009	1.7e-015	1.4e-015	1.7e-015	1.4e-015	1.7e-015

Accuracy for small groups

Smaller sized translation matrices have been shown to be problematic for the lossless cases [21], and the following studies show that the same is true when k_ρ is complex. Tables 7.5–7.7 record results of integration error for small boxes, i.e., $k_\rho D_{2D} < 5$.

For the lossless and lossy cases, 15 points are sufficient to achieve single precision accuracy, but in the active case, 20 points are needed. The principal reason is due to the quadratic mapping function. When the magnitude of k_ρ becomes small, the quadrature points become scaled by the mapping function and are badly distributed along the path.

Depending on the length of the path, the nodes can become sparsely distributed in the shallow region where the waves are more important, and densely distributed in the deep evanescent region. Figure 7.4, page 97, illustrates the distribution of points when using the quadratic path. The points are sparsely distributed in the shallow evanescent region, and densely distributed in the deep evanescent region. When this happens, the subsequent interpolation and extrapolation loses accuracy. To overcome this, the number of quadrature points must be increased to obtain more points in the shallow region. Other maps and quadrature routines were studied, but none could provide at least single precision accuracy for all conditions.

Based on the higher number of quadrature points that are needed on paths I and III of 2-D FIPWA, it is apparent that 2-D FIPWA must call the interpolation routine more often than 2-D FMA. For example, given the small box size $|k_\rho D_{2D}| = 0.1$, 2-D FMA requires only 8 quadrature points and 8 calls to the interpolation routine. In contrast, 2-D FIPWA requires at least 2 quadrature points on path II and 30 quadrature points on paths I and III combined, making at least 32 calls to the interpolation routine. For small boxes, 2-D FMA is much faster, in spite of the cost to compute special functions in the multipole expansion.

Table 7.5: Integration error of small boxes in 2-D FIPWA for $\tan \delta = 0$. Gauss-Laguerre integration is used on paths I and III, and Gauss-Legendre integration is used on path II.

$ k_\rho D_{2D} $	N_q						
	5	9	15	20	30	40	50
0.1	2.8e-002	6.7e-004	6.2e-006	1.1e-007	4.1e-011	2.0e-012	3.6e-013
0.2	2.0e-003	5.5e-004	1.8e-006	2.2e-008	6.5e-011	2.4e-012	4.2e-013
0.5	2.3e-002	2.5e-004	1.0e-006	2.1e-008	7.4e-011	2.7e-012	4.6e-013
1.0	2.2e-002	2.6e-004	9.7e-007	2.0e-008	7.1e-011	3.1e-012	6.8e-013
3	5.5e-003	1.0e-004	3.8e-007	7.9e-009	6.8e-010	7.0e-010	7.0e-010

Table 7.6: Integration error of small boxes in 2-D FIPWA for $\tan \delta = \frac{\pi}{6}$. Gauss-Laguerre integration is used on paths I and III, and Gauss-Legendre integration is used on path II.

$ k_\rho D_{2D} $	N_q						
	5	9	15	20	30	40	50
0.1	2.0e-002	1.6e-004	9.9e-007	1.7e-008	5.9e-011	2.2e-012	4.0e-013
0.2	1.2e-002	2.7e-004	9.6e-007	2.0e-008	7.2e-011	2.7e-012	5.2e-013
0.5	2.1e-002	3.1e-004	1.2e-006	2.5e-008	9.1e-011	3.5e-012	7.2e-013
1.0	2.4e-002	3.4e-004	1.3e-006	2.7e-008	1.0e-010	4.0e-012	9.7e-013
3	2.0e-002	2.5e-004	8.9e-007	2.0e-008	4.1e-010	3.4e-010	3.3e-010

Table 7.7: Integration error of small boxes in 2-D FIPWA for $\tan \delta = -\frac{\pi}{6}$. Gauss-Laguerre integration is used on paths I and III, and Gauss-Legendre integration is used on path II.

$ k_\rho D_{2D} $	N_q						
	5	9	15	20	30	40	50
0.1	6.4e-002	6.3e-003	2.0e-004	1.2e-005	4.6e-008	1.8e-010	1.1e-012
0.2	3.9e-002	3.3e-003	5.7e-005	2.1e-006	3.2e-009	5.5e-012	3.7e-013
0.5	4.3e-002	1.1e-003	6.8e-006	1.5e-007	9.0e-011	2.3e-012	4.0e-013
1.0	2.8e-002	1.6e-004	9.5e-007	2.5e-008	5.6e-011	9.3e-013	1.0e-012
3	1.7e-003	9.0e-005	2.9e-007	5.5e-009	1.1e-009	1.0e-009	1.0e-009

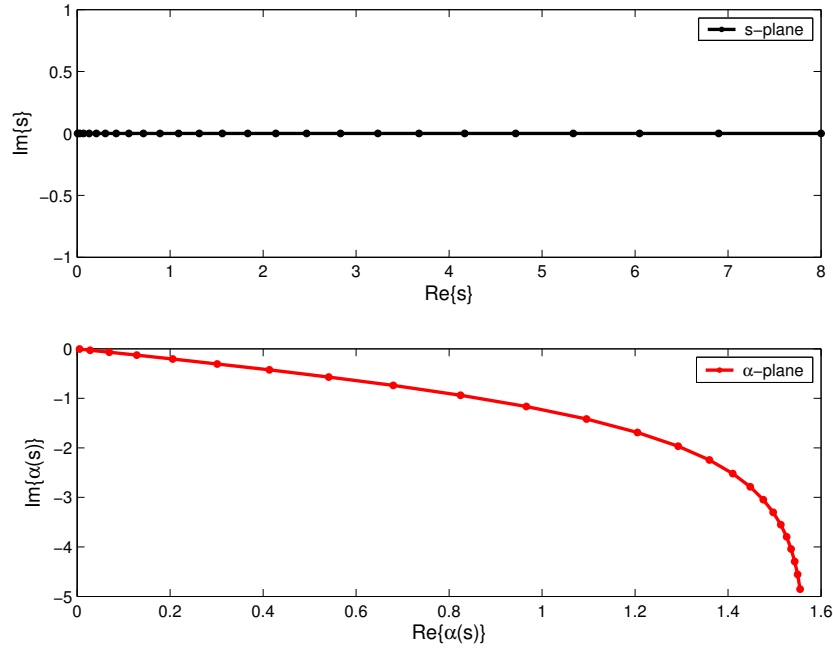


Figure 7.4: Distribution of quadrature points from quadratic map. Top: Distribution of Gauss-Laguerre quadrature points on the real axis. Bottom: Distribution of $\alpha(s)$ when using a quadratic map. The points are sparsely distributed in the shallow evanescent region and densely distributed in the deep evanescent region.

Summary of numerical integration

The accuracy of the numerical integration of 2-D FIPWA was compared to the exact solution. Using scaled and windowed Gauss-Laguerre quadrature, it was shown that 15 points are needed to achieve single precision accuracy. Based on the number of points needed for the case of large translation matrices, 2-D FIPWA constructs the translator faster than 2-D FMA. Yet, in the case of small boxes, 2-D FMA creates the matrix faster than 2-D FIPWA.

7.4.2 Interpolation and extrapolation

Various interpolation functions were studied in [21] for FIPWA, so the details of the sinc and Dirichlet interpolation function are not presented. In addition, the Lagrange interpolation function and error control are presented in [17], so they are not presented in detail either.

In constructing the 3-D translator from the 2-D FIPWA translator in complex media, one must remember that $k_\rho = k \sin \beta$, where β is complex. The hyperbolic behavior of $\sin \beta$ causes k_ρ to range from zero to infinity, and the integrand to have an effective bandwidth of $(0, \infty)$. In addition, the effective background medium for k_ρ varies from pure loss to pure gain, for the same reason. This requires an interpolation function that can extrapolate with high accuracy for all possible values of k_ρ . It was soon discovered that the traditional interpolation functions, sinc and Dirichlet, could not be used to arbitrary accuracy for all possible values of k_ρ .

Results with sinc and Dirichlet functions

For the cases when $|k_\rho| < |k|$, the bandwidth is effectively smaller, so the interpolation function must be computed for a smaller bandwidth. With the data structure designed to store a higher sampling rate than that called for by $|k_\rho|$, the sinc and Dirichlet functions lose 2-6 digits of accuracy for extrapolation. As a means to overcome the limitation of the sinc and Dirichlet interpolation function, an interim translator is constructed with fewer samples, and then the interim translator is interpolated up to the size of the desired translator. This would cost more in setup time, but by using an optimized method, the cost is actually reduced. The optimized method is discussed at the end of this section.

Results with Lagrange polynomials

Alternatively, the Lagrange interpolation polynomials were used for extrapolation and could be controlled to double machine precision accuracy. However, the computational expense of computing the Lagrange coefficients was 10-100 times more expensive than the sinc function. Local extrapolation did not produce accurate results.

Alternate continuum interpolation function

One option is to design a new extrapolation function that is based on digital sinc interpolation [41–43]. The interpolation function is derived, in Appendix A, for the class of functions $g(x)$ that have a Fourier transform $\tilde{g}(k)$, are band-limited to $-P < k < P$, and are periodic with period L . The functions are sampled at the rate L/N and the interpolation function is

$$I_L(x) = \frac{N}{2\pi L} \frac{\sin[(2M+1)\frac{N}{2L}x]}{\sin(\frac{N}{2L}x)}, \quad (7.9)$$

where N is the number of desired samples for the translation matrix, L is the period, $M = \lfloor \frac{L}{N}P \rfloor$, P is the effective bandwidth of the specific translator function, and $M = \lfloor \cdot \rfloor$ means to round down to the nearest integer.

Using this interpolation approach for extrapolation, one can eliminate the need to interpolate at a low sampling rate, and then interpolate to a higher sampling rate. The result is that the extrapolation function does not degrade the accuracy set by the numerical integration. However, this was not necessary because the optimized method, discussed next, retained the accuracy and reduced setup time.

Optimization of interpolation and extrapolation

Interpolation and extrapolation with 3-D FIPWA in free space and 2-D FIPWA in complex media have been discussed in the literature and a previous chapter. Hence, the details are not discussed here. Instead, a method to improve extrapolation in MF-FIPWA with better efficiency is presented. It also applies to FIPWA.

As discussed in Chapter 4, extrapolation with the sinc or Dirichlet function loses accuracy when P is large and $\mathcal{I}m\{\alpha\}$ is large, so oversampling degrades the accuracy when extrapolating to α . To retain high accuracy for MF-FIPWA, the number of samples used to extrapolate to α in the complex plane is reduced when $k_\rho < k$. This corresponds to a smaller bandwidth that needs fewer samples for exact interpolation with the sinc interpolation function. Correspondingly, the extrapolation error is reduced.

For example, in constructing the 2-D translation matrix, there are N_q integration points on Γ_α and each point must be interpolated or extrapolated from a set of samples on the real axis. The real bandwidth kD requires N_s samples, so the diagonal translation matrix is computed as a vector with N_s elements. When $k_\rho D_{2D}$ requires only N'_s samples, where $N'_s < N_s$, an interim diagonal translation matrix is constructed with $N'_s = N_s/L$, and L is an integer factor of N_s . In digital signal processing theory, this process is known as decimation and is used to change the sampling rate by a rational number. Using a reduced number of samples improves extrapolation accuracy and efficiency, but the final translation matrix must be constructed with N_s samples. Using vector notation, the interim translator is interpolated to N_s samples as

$$[\mathbf{T}]_{N_s \times 1} = [\bar{\mathbf{I}}]_{N_s \times N'_s} \cdot [\bar{\mathbf{I}}']_{N'_s \times N_q} \cdot [\boldsymbol{\tau}]_{N_q \times 1}, \quad (7.10)$$

$$= [\bar{\mathbf{I}}]_{N_s \times N'_s} \cdot [\mathbf{T}']_{N'_s \times 1}, \quad (7.11)$$

where $\mathbf{T}'$ is the interim translation matrix, $[\boldsymbol{\tau}]_q = e^{ik_\rho \rho \cos(\alpha_q)}$, $[\bar{\mathbf{I}}']_{n'q} = \text{sinc}(\alpha_q - \phi_{n'})$, and $[\bar{\mathbf{I}}]_{nn'} = \text{sinc}(\phi_n - \phi_{n'})$.

When the sinc function is used to interpolate from $\mathbf{T}'$ to $\mathbf{T}$ it is exact. Computational savings occur because $N_s = LN'_s$ and all of the values in $\mathbf{T}'$ belong in $\mathbf{T}$. Before interpolation, $\mathbf{T}$ has N'_s nonzero values interlaced with L zeros. Only the zero elements of $\mathbf{T}$ need to be computed. Results showed that MF-FIPWA has a savings of up to 20% in constructing the 2-D translation matrix in this fashion. When this same method was applied to interpolation in the 2-D FMA component of 3-D FIPWA, computation time was reduced 10%. The reason for the difference is that MF-FIPWA uses more quadrature points to achieve the same accuracy.

7.5 Error Control Methodology

Approaches to error control typically try to find a single parameter to which all others are dependent. In MLFMA, it is simple to develop global error control, where all of the error

sources are dependent on P , the spectral bandwidth. The excess bandwidth formula also determines the spectral content of the multipole expansion for FIPWA, and the interpolation and extrapolation functions in MF-FIPWA. Yet, there are differences in how the interdependency of numerical quadrature, interpolation, extrapolation, and multipole expansion should be controlled between MLFMA, FIPWA and MF-FIPWA. Hence, design of algorithms such as FIPWA and MF-FIPWA require a different approach to error control.

7.5.1 Global error control

Global error control is defined, in this work, as controlling the error with a single set of parameters that are related to all the error sources. For example, in MLFMA, there are three sources of error: numerical integration, truncation of the multipole translator, and interpolation between levels. Each source can be controlled so that the error decreases exponentially with proper choice of the multipole truncation number, P [17,35]. The multipole truncation is related to the bandwidth of the radiation patterns, $e^{i\mathbf{k}\cdot\mathbf{D}}$; hence, P sets the sampling rate for interlevel sampling in the multilevel paradigm. The refined excess bandwidth formula,

$$P = kD + 1.8d_0^{2/3}(kD)^{1/3}.$$

determines the number of multipoles according to the bandwidth of the translator function. Given the desired number of digits of accuracy, d_0 , of the 2-D multipole translator, P represents the single-sided bandwidth of the expansion. This, in turn, establishes the sampling density needed for the interlevel interpolation. Note that the samples are on the unit sphere, so by letting N_θ and N_ϕ equal the number of samples in θ and ϕ , respectively, $N_{samples} = N_s = (N_\theta - 2)(2N_\phi) + 2$.

Finally, the integration path in MLFMA is $\alpha \in [0, 2\pi]$, and the integrand is periodic. Hence, the trapezoidal rule is accurate with $2P$ sample (quadrature) points. Each source of error has been shown to be exponentially controllable [17] and dependent on the single parameter P .

7.5.2 Local error control

On the other hand, local error control is defined as error control with more than one set of parameters, where the different sets are related to different error sources.

For 3-D FIPWA in layered media, the error sources have not been shown to be completely interdependent. In [29], the dependency on P was derived for the extrapolation function, and in [20, 44], the dependency was shown to be valid for complex media. Thus, the extrapolation error and interlevel interpolation error can be jointly controlled with P . However, the numerical integration on the SDP depends on the type of quadrature rule, and is not necessarily dependent on P . MF-FIPWA uses Gauss-Laguerre integration on paths I and III, and Gauss-Legendre integration on path II. While integration accuracy on path II is related to the bandwidth, integration accuracy on paths I and III is not. MF-FIPWA is considered to have local error control, and this extends to the 2-D FIPWA translator that is used in MF-FIPWA.

In both FIPWA and MF-FIPWA for layered media, error sources are numerical integration on the SDP, interlevel interpolation, interpolation and extrapolation in the θ -plane, interpolation and extrapolation in the α -plane, and multipole truncation. While not rigorously explored, local error control proved to be manageable and effective for FIPWA in layered media [14, 22, 23]. MF-FIPWA also uses a local error control approach for controlling the number of points on paths I and III of the SDP.

In MF-FIPWA, there are two elements to the localized error control. First is the control for the complex media, and second is the control for the magnitude of $k_\rho D$. When the 2-D FIPWA translator is used to construct the 3-D MF-FIPWA translator, k_ρ potentially takes on all values in the complex k_ρ plane. This makes the error control independent of the real bandwidth of the translator. The length of the SDP must be set to control the bandwidth, but the quadrature rule must compute the integral with high accuracy for many cases. In some cases, the accuracy can be relaxed. Hence, the local control is based on the exact value

of k_ρ , the group diameter D , and the quadrature rule. In the next section, these aspects are discussed for MF-FIPWA.

7.6 Applied Error Control in MF-FIPWA

There are three methods to local error control in MF-FIPWA. First, the error is controlled by proper choice of the interpolation and extrapolation function. Second, the inner integration is controlled according to the specific SDP for $k_\rho = k \sin \beta$, and third, the inner integration is also controlled according to the complex value β . Each of these can be considered a form of optimization because they incur the most efficiency for the desired accuracy.

7.6.1 Error control according to bandwidth

As discussed in previous sections, the bandwidth of the radiation patterns determines the number of samples that are used to store the translation matrix. To control the error, the interpolation function must consider the cases when $k_\rho < k$ and when $k_\rho \gg k$. The former case has been addressed in the previous section on optimization for interpolation and extrapolation. The latter occurs when $\mathcal{Im}\{\beta\} > 1$, where β is on Γ_β . When $k_\rho \gg k$, there should be very little propagation up from the layered medium.

If the interaction is between an image group and field group that are both near the interface, then extrapolation should be computed with an increased sampling rate to ensure that the accuracy is met. If a larger sampling rate is used, then downsampling will be required and the construction time will increase. However, when one or both groups are far from the interface, there are two options. First, the interaction is negligible and can be dismissed. Second, the interaction can be computed with reduced accuracy. For convenience in MF-FIPWA, the interactions are computed with reduced accuracy in the extrapolation process.

7.6.2 Error control according to the inner SDP

The SDP control depends on $\tan \delta = \frac{\mathcal{Im}\{k_\rho\}}{\mathcal{Re}\{k_\rho\}}$, and the effective bandwidth $k_\rho D_{2D}$, where $k_\rho = k \sin(\beta_R + i\beta_I)$, $D_{2D} = D \frac{\sqrt{2}}{\sqrt{3}}$, and D is the diameter of the sphere upon which the translator is defined. These error sources have been described in Chapter 4 and [20], but path is easily truncated as

$$s_{\max} \geq \sqrt{d_0 \log 10 - \mathcal{Im}\{k_\rho\} \rho_{JI}}, \quad (7.12)$$

where d_0 is the desired digits of accuracy in the integration and ρ_{JI} is the 2-D translation distance. It is seen from (7.12) that as k_ρ becomes lossy ($\mathcal{Im}\{k_\rho\} > 0$), the path decreases, and when $\mathcal{Im}\{k_\rho\} < 0$, the path lengthens. This agrees with the behavior of the SDP in complex media [20]. For 15 digits of accuracy and lossless media, $s_{\max} \leq 6.0$. Yet, not all 2-D translators need to have 15 digits of accuracy in the integration.

Based on the error results in Tables 7.2–7.7, it is sufficient to bound the path length in $3.0 < s < 10.0$ and adjust the number of quadrature points accordingly. In MF-FIPWA, the number of quadrature points on paths I and III of Γ_α is

$$N_\alpha = \begin{cases} 5, & s = 3.0, \\ 9, & 3.0 < s < 10.0, \\ 15, & s = 10.0 \end{cases} \quad (7.13)$$

While the accuracy of the 2-D translator is not as high as the multipole expansion, scattering results of MF-FIPWA agree with those of FIPWA to within 2% for small to moderate sized problems and within 1% for large problems. This optimization also enables MF-FIPWA to construct the 3-D translation matrix faster than FIPWA for box sizes greater than 2.0λ .

7.6.3 Error control according to the outer SDP Γ_β

An additional measure of control is to determine how far β is from the real axis. When β is on paths I or III of Γ_β , it is complex and the integrand decays exponentially fast. For

large values of β_I , the inner integral on Γ_α does not need to be computed with high accuracy because it makes a very small contribution to the double integral. This allows the path for Γ_α to be shortened and computed more quickly. This is especially true when the boxes are large, or several box lengths from the interface of the layered medium. In these cases, $s_{\max} \leq 1.0$ and $N_\alpha = 3$.

Clearly, there are various ways to optimize MF-FIPWA, but each method requires logic statements in the error control module. To enable compiler optimization, this logic should be minimized, or performed with elementary math operations. Otherwise, the added logic statements could prevent the code from gaining the benefits of certain compiler and machine combinations.

7.7 Summary

In this chapter, several aspects of FIPWA and MF-FIPWA were presented to show how MF-FIPWA is implemented. The fast algorithm architecture is shown with emphasis on how to construct the 3-D translation matrix. Additionally, specific details of the scaled and windowed Gauss-Laguerre integration showed that only 15 points are needed to achieve single precision accuracy in the 2-D translation matrix. Results of several interpolation functions were briefly presented to illustrate the differences, and a new method was shown to speed up the interpolation by 20%. Finally, a new error control methodology was presented for localized error control and optimization in MF-FIPWA.

CHAPTER 8

TESTING AND DEBUGGING

8.1 Introduction

In any code development, there are times when leaps and bounds occur during code production, and inevitably a bug is introduced that forces the developer to backtrack. Modifying FIPWA into MF-FIPWA was no different. There were two bugs in the code that affected the accuracy and stability for several weeks. This chapter highlights the general debugging approach and specific debugging approaches to fixing these bugs.

8.2 Technical Aspects

8.2.1 Code compatibility

When starting to modify FIPWA, Microsoft Visual Studio 6.0 was the tool of choice to trace through the mixed C and Fortran code. As a matter of personal preference, operating under the Windows operating system was more productive than using Sun's Workshop on the UNIX operating system. However, the code had to be modified to compile on IBM platforms, and even then, it did not run to completion.

Fortunately, the code could run through the setup stage, and the MF-FIPWA translator could be implemented and tested. The testing took a period of months while the code was

learnt and components for debugging and testing were added. Part of this time was spent implementing the error control criteria of Chapter 4.

During this development, Microsoft Visual .NET became available and provided support for newer Intel compilers that were faster and improved for mixed language programming. The code was moved to the .NET environment with the latest Intel C and Fortran compilers. During the porting effort, a few libraries under .NET were found to be different from the 6.0 environment, and it took some time to make the modifications. Ultimately, the code compiled, built, and executed, while retaining compatibility with UNIX. This was very helpful, because the code could be run under Windows and compared to the original UNIX solution.

When it came time to run moderate to large scale problems, a 64-bit processor was needed. The code was compiled on the cluster of Sun Blade 1000 computers in the Center for Computational Electromagnetics and Electromagnetics Laboratory (CCEML). Initially the code would compile, but it would not execute, even though there were no serious compiler or linker warnings. Professor Jose Shutt-Aine, of CCEML, made the suggestion to use the Linux compiler because it is known to have a stricter compiler. Surprisingly, it took less than one week to trace through the Linux compiler errors and warnings. Once modified, the 64-bit code compiled and executed to completion on Linux and UNIX. This effort enabled the one-million unknown scaling problems.

One nuisance in the code was the declaration of `long int` for storing memory usage. Declaring the integers as signed values limited the range of values for reporting bytes used by the code. Instead, `unsigned long int` was substituted for the data type to improve how the code reported memory usage. This upgrade also made it possible to compile and run a 32-bit version of the code for problems over 2 GBytes. With the range limitation, memory could not be allocated beyond 2 GBytes. Unfortunately, that would not happen until the last stage of setup that occurred hours into the simulation. Lastly, this change

made it possible to economize memory on the shared Sun cluster. While a simple upgrade, the code is now more versatile, and several moderate sized problems (typically fewer than 500,000 unknowns) can be run on a single processor.

8.2.2 Code limitations

Besides the technical updates and improvements, several limitations had been programmed into the code. For example, material data for the layers had to be compiled for any change, global parameters were defined that limited the maximum size of the translation matrices, and edge data had to be created with every simulation at a cost of $\mathcal{O}(N^2)$. To clean up the code, and spend time more productively, a material class was created and the input file was modified to allow the user to include the data directly in the input file, or specify a material file. Second, edge data was saved to a file in the case where it did not exist, avoiding the need to recompute the data. Several global parameters were stored in a single include file, but they had to be modified to run simulations with more than one million unknowns. Some of the key parameters were removed, and other parameters were set large enough so that they did not impede the large jobs with recompilation.

These modifications were necessary to make testing and debugging more efficient, but also made the code more user friendly. Additional modifications to the input file allow the user to select the FIPWA or MF-FIPWA translator and to select the number of buffer boxes at run time. The code was also modified to allow the user to scale the geometry file in an effort to keep the input file format functionally equivalent to the input file format of the Fast Illinois Solver Code [17].

8.3 Debugging Methods

A nontrivial part of this work was to replace the 2-D translator as a black box with MF-FIPWA, but various debugging routines had to be added to confirm the accuracy. Typically, these routines would compute the relative error and print out the associated parameters and settings. Instead of using preprocessing commands and recompiling the code, debugging flags were placed in the text file `debug.flags` that could be read at run time. A subroutine was added to check for the file `debug.flags`, and if it was not found, all debugging flags were switched off. Otherwise, the specified debugging routines were set according to `debug.flags`. For example, when setting up the error control, it was often helpful to run FIPWA side by side MF-FIPWA for comparison. By simply selecting the FIPWA translator and associated debugging flags, the translators could be compared side-by-side. Additionally, part of the cleanup effort included making the routines to compute the 2-D FMA and 2-D FIPWA translators modular.

8.3.1 Comparison by Green's function

Recall from Chapter 2, the 2-D Green's function, $g(k_\rho \rho_{ji}) = \frac{i}{4} H_0^{(1)}(k_\rho \rho_{ji})$ can be expanded by factoring $\rho_{ji} = \rho_{Jj} + \rho_{JI} + \rho_{Ii}$ and introducing interpolation to form a vector-matrix-vector product of a receiving pattern, a translator, and a radiation pattern. However, if $\rho_{Jj}, \rho_{Ii} \rightarrow 0$, then the radiation and receiving patterns become a constant equal to one, and the Green's function is equivalent to the sum of the samples stored in the 2-D translator when scaled by the constant $C = \frac{i}{4}$. Hence, comparison of the 2-D translator with the Hankel function is a quick way to confirm the error control of 2-D FIPWA for complex media. This approach helped find the reason why the accuracy was limited in spite of error control measures. Note that this approach also applies to checking the 3-D translator because the construction of the 3-D translator is similar in design to the 2-D translator.

Construction of the 2-D translator

To compare the Hankel function to the translator independently of the interpolation function, the discrete form of the translator function is computed in two stages. First, recall the discrete form of the 2-D FIPWA translator:

$$\mathcal{T}(\phi) = \int_{\Gamma_\alpha} d\alpha f(\alpha) I(\alpha - \phi), \quad (8.1)$$

$$= \sum_{q=1}^N w_q f(\alpha_q) I(\alpha_q - \phi), \quad (8.2)$$

$$= \sum_{q=1}^N h(\alpha_q) I(\alpha_q - \phi), \quad (8.3)$$

$$= \mathbf{h} \cdot \mathbf{I}(\phi), \quad (8.4)$$

where $f(\alpha) = \frac{1}{\pi} e^{ik_\rho \rho \cos \alpha}$, and $[\mathbf{h}]_q = w_q f(\alpha_q)$. The vector $\mathbf{h}$ is really just the discrete form of the integral expression for the Hankel function, so that

$$H_0^{(1)}(k_\rho \rho) = \sum_{q=1}^N [\mathbf{h}]_q. \quad (8.5)$$

The second stage is to compute the vector $\mathbf{I}(\phi)$ for each value of ϕ , compute the dot product $\mathbf{h} \cdot \mathbf{I}(\phi)$, and store the result $\mathcal{T}(\phi)$ in the vector $\mathcal{T}$. Traditionally, the translator is considered a matrix, but it is a diagonal matrix. Here, it is treated as a vector for simplification.

Confirming the accuracy

By computing each stage with vectors, debugging routines can be modularized to work on each vector at the end of the respective stage. The relative error is easily checked for the first stage as

$$\text{Relative error} = \frac{\left(\left| \sum_{q=1}^N [\mathbf{h}]_q - H_0^{(1)}(k_\rho \rho) \right| \right)}{\left| H_0^{(1)}(k_\rho \rho) \right|}. \quad (8.6)$$

Following the second stage of interpolation and extrapolation, the Hankel function can be compared to the sum of the elements of $\mathcal{T}$:

$$H_0^{(1)}(k_\rho \rho) = \sum_{k=1}^{N_\phi} [\mathcal{T}]_k. \quad (8.7)$$

This two-tier method to debugging the 2-D translator enables one to first verify the accuracy of the SDP integration, and then check the accuracy of the following interpolation and extrapolation. It also enables comparison of various interpolation and extrapolation routines.

Finding the accuracy bug

One aspect of this research was to compute the SDP integral with arbitrary accuracy (up to machine limits of 15 digits), but it was realized that only 12-13 digits could be achieved consistently. After using the above approach to isolate the stage where the error was introduced, the problem was solved quickly.

The error was introduced inadvertently into the form of the Jacobian of the mapping function that was used to find the SDP. For the fundamental SDP, i.e., when there is no modification as discussed in previous chapters, the SDP can be found for the integrand $h(\alpha) = e^{ik_\rho \rho \cos(\alpha)}$ by following the steepest descent method in [1]. The fundamental saddle point occurs at $\alpha_{sp} = 0$, and $h_0 = h(\alpha_{sp}) = ik_\rho \rho$, allowing the quadratic mapping function to be defined for points near the saddle point as

$$-s^2 = h(\alpha) - h(\alpha_{sp}), \quad (8.8)$$

$$= ik_\rho \rho \cos(\alpha) - ik_\rho \rho, \quad (8.9)$$

$$= ik_\rho \rho (\cos(\alpha) - h_0), \quad (8.10)$$

where $s, \rho \in [0, \infty]$, $k_\rho = k_{\rho,r} + ik_{\rho,i}$, and $h_0 = 1$. Note that the last parameter, h_0 , is determined by the location of the saddle point. Hence, it can be used as a control parameter for exploring alternate SDPs. For example, the fundamental SDPs that are used to form

the modified SDP have saddle points at $\alpha_{sp} = \pm\phi_G$, where $\phi_G = \sin^{-1}(D/\rho)$, and D is the diameter of the group. In these cases, $h_0 = h(\alpha_{sp}) = ik_\rho\rho \cos(\phi_G)$.

In this work, the modified SDP is formed by shifting the fundamental SDP by ϕ_G , so that the fundamental saddle point results in $h_0 = 1$, and the fundamental SDP is defined by

$$\alpha(s) = \cos^{-1} \left(h_0 - \frac{s^2}{ik_\rho\rho} \right). \quad (8.11)$$

There is an associated Jacobian function for the mapping function, $\alpha(s)$, that is defined as $J(s) = \frac{\partial}{\partial s}\alpha(s)$. The Jacobian function for (8.11) is

$$J(s) = \frac{\frac{2s}{ik_\rho\rho}}{\sqrt{h_0 - (1 - \frac{s^2}{ik_\rho\rho})^2}}. \quad (8.12)$$

However, in this form $J(s)$ has three divide operations that limit the accuracy in computing $J(s)$, and subsequently the SDP integral,

$$\begin{aligned} I = I_I + I_{II} + I_{III} &= \frac{1}{\pi} \int_{-\infty}^0 ds \, e^{ik_\rho\rho \cos(\alpha(-s))} J(-s) \\ &\quad \frac{1}{\pi} + \int_{-\phi_G}^{\phi_G} d\alpha \, e^{ik_\rho\rho \cos \alpha} + \frac{1}{\pi} \int_0^\infty ds \, e^{ik_\rho\rho \cos(\alpha(s))} J(s), \end{aligned}$$

to 12-13 digits of precision.

When $h_0 = 1$, $J(s)$ can be simplified to

$$J(s) = \frac{2}{\sqrt{2ik_\rho\rho - s^2}}. \quad (8.13)$$

Although it is possible for $k_\rho = is^2/(2\rho)$, it has not been observed. As a precaution, the original form of $J(s)$ is used should it occur. Also, $J(s)$ is an even function, which makes it possible to reduce the three integration paths to just two paths.

Ultimately, the bug was resolved and arbitrary accuracy was achieved.

Linear mapping function for finding the SDP

As mentioned above, the quadratic mapping function exhibits problems with the distribution of the quadrature points. A better method is to use the quadratic function only for points

near the saddle point, and to use a linear mapping function for the remaining parts of the path.

The linear mapping function is derived in the same fashion as Eq. (8.11).

$$-s = ik_\rho \rho \cos \alpha - ik_\rho \rho, \quad (8.14)$$

$$= ik_\rho \rho (\cos \alpha - 1), \quad (8.15)$$

$$\alpha(s) = \cos^{-1} \left(1 - \frac{s}{ik_\rho \rho} \right), \quad (8.16)$$

where the saddle point is located at $\alpha_{sp} = 0$. The Jacobian function is

$$J(s) = \frac{1}{\sqrt{2ik_\rho \rho - s^2}}. \quad (8.17)$$

8.3.2 Comparison by plane wave direction

Besides the inaccuracy of the Green's function expansion, the code was initially unstable, i.e., for small problems the results of MF-FIPWA agreed with FIPWA, and as the problem size increased, the two sets of results did not agree. After simplifying the Jacobian function, the error could be controlled to arbitrary accuracy, but the instability remained. While the sum of the matrix elements was verified to be arbitrarily accurate in Eq. (8.7), the iterative matrix solution would not converge, or when it did, the solution did not compare well with FIPWA. To find the bug in the code, the 2-D FIPWA translator was examined sample-by-sample and compared to the 2-D FMA translator.

Each translator is defined on a sphere of a certain size and for a specific direction. In cylindrical coordinates, the 2-D translator is defined for the vector $\boldsymbol{\rho}_{JI}$ with the angular direction ϕ_{JI} . The plane wave representation of the Green's function expansion is captured in the translator as angular samples, so that if one examines the stored samples of the translator, i.e., the plane wave representation, then the dominant values should occur in the angular direction for which the translator is defined (ϕ_{JI}).

Upon comparing the elements of the 2-D FMA and 2-D FIPWA translators side-by-side, it was immediately clear that the matrix elements of the 2-D FIPWA translator were improperly indexed during its construction. Hence, the plane waves were correct in value, but were placed in the wrong location.

Figure 8.1 illustrates how the plane wave directions were incorrectly indexed in the 2-D FIPWA translator, and how they appeared after the indexing was corrected. The solid stem lines with circles represent the location of the samples before the fix and the dashed stem lines show the location after the fix. The dotted line is the direction of the translation, ϕ_{JI} . With the indexing corrected, the dominant plane waves are symmetrically placed about ϕ_{JI} . With this bug repaired, the code scaled well for large problems, as seen in Chapter 5.

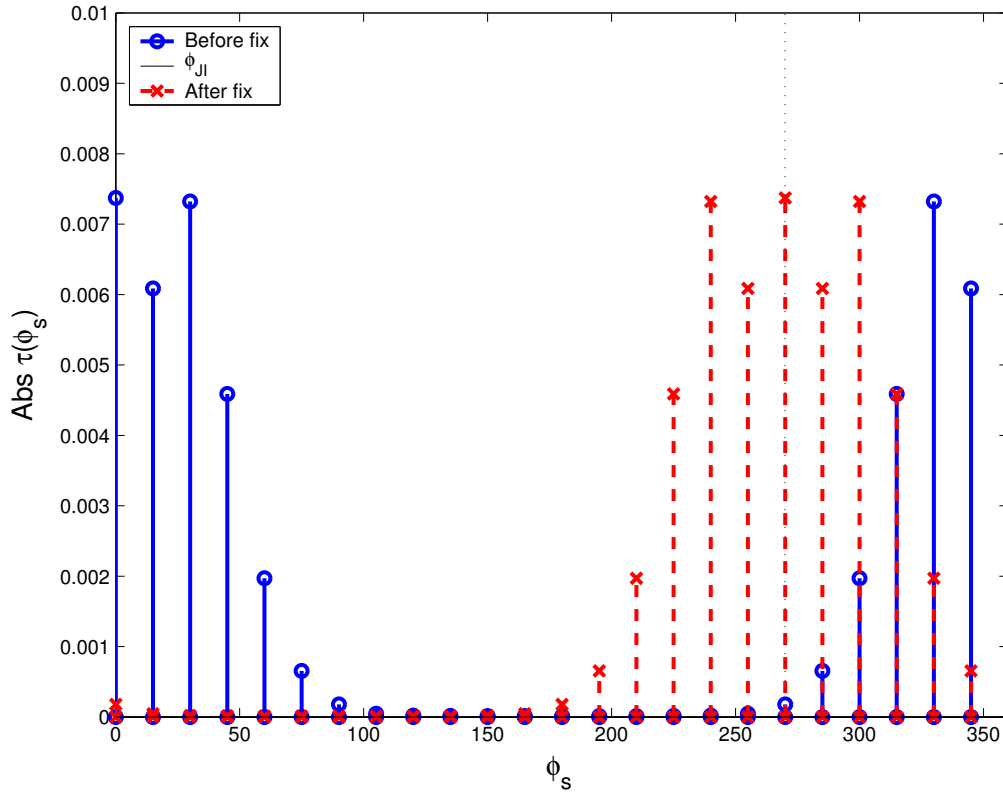


Figure 8.1: Debugging the stability problem. The instability was caused by misaligning plane waves during construction of the translation matrix. After fixing the code, the plane waves were properly aligned to the direction of the translation, ϕ_{JI} , shown by the dotted line.

8.4 Summary

To develop MF-FIPWA, the original FIPWA code had to be modified. To aid in the effort, various improvements were made, and the code was modified to run on alternate operating systems. When problems of accuracy and stability arose, debugging routines and techniques were added to find the bugs. Ultimately, the code was improved and debugged so that MF-FIPWA runs on Windows, UNIX, and Linux, and can solve large scale problems with arbitrary accuracy.

CHAPTER 9

NONUNIFORM SAMPLING OF θ - ϕ PLANE

9.1 Introduction

The fast algorithm provides a tremendous savings in memory and cpu resources with a cost of $CN \log N$. While there is not much more that can be done to improve the scaling, the scaling constant can be improved to achieve further savings. In this chapter, one such approach is taken to attack the largest cost of memory and setup time, i.e., the radiation patterns. As a result, the time to compute the matrix-vector product is reduced.

The idea of optimal sampling was presented by Bucci [38], who developed local interpolation of a sphere to reduce the data needed to store and represent radiation patterns of antennas. The idea has been considered for the fast algorithm previously [35], but it has not been implemented because an added level of sophistication must be programmed into the code. The traditional trade off has been to take the easier, less sophisticated code over the potential savings.

Here, the optimal sampling, or nonuniform sampling, of the radiation patterns is studied for FIPWA with the note that it automatically applies to MF-FIPWA. Previous estimates of reduction in memory were 20%. Using moderate sized spheres, the set-up time and cost of the radiation pattern is shown to be reduced by up to 30%.

The chapter is organized as follows. First, the distribution of typical memory costs in FIPWA is shown, followed by the new format to store the translation matrices. Lastly, results of the potential savings are presented.

9.2 Memory and CPU Costs of the Fast Algorithm

The distribution of memory and setup costs in the fast algorithm have been studied for MLFMA in various sources [17, 35]. FIPWA and MF-FIPWA for layered media follow the same cost trends to include the addition of the translator for layered medium. In the layered media problem, the outgoing-to-incoming (O2I) translator has two forms. One is the free space translator for the direct interaction, and the other is the O2I translator for the layered, or stratified, medium. The latter is called the O2I-SM translator to distinguish it from the free-space translator. Other components of FIPWA, such as interpolation and direct interaction matrices, have essentially the same cost of memory and time as in MLFMA. Hence, the reader is referred to the references for more details on interpolation, outgoing-to-outgoing (O2O), incoming-to-incoming (I2I) matrices, and direct interaction matrices.

In this section, the radiation pattern matrices are associated with the leafy boxes of the MLFMA tree, i.e., the radiation pattern of the basis elements. Recall that in MLFMA, FIPWA, and MF-FIPWA, the radiation patterns at the leafy level are aggregated to the smallest box in the tree to form a higher level radiation pattern. However, this occurs during the matrix-vector product. First, the leafy boxes of the tree are computed and stored during the set-up stage.

As a prelude to the nonuniform sampling approach, the O2I and O2I-SM translators have each been constructed with different methods than earlier versions of FIPWA. Hence, the test code is called FIPWA with **Nonuniform Sampling** (FIPWANOS).

For a representative example without using nonuniform sampling, the memory costs and times are shown in Tables 9.1 and 9.2 for the scattering by a 1-m sphere discretized with 500,300 unknowns.

Table 9.1: Distribution of memory for FIPWA. Results are for a 1-m sphere with 500,300 unknowns above a half space, and two typical sizes for the smallest box. In each case, 8 levels are used. The frequencies are 2.959 GHz and 4.197 GHz for smallest box sizes of $\lambda/10$ and $\lambda/5$, respectively.

	Memory Cost (MB)		Percentage of Total	
	$\lambda/10$	$\lambda/5$	$\lambda/10$	$\lambda/5$
O2O and I2I	2.6	8.7	0.13	0.22
O2I	50.7	172.8	2.53	4.30
O2I-SM	421.6	631.8	21.06	15.74
Pole Contribution	9.7	115.6	0.48	2.88
Interpolation	<1.0	<1.0	0.05	0.02
Block Preconditioning	33.9	33.9	1.69	0.84
Direct Interaction Matrix	349.9	350.1	17.48	8.72
Radiation Patterns	710.1	1 872	35.48	46.64
Other	422.5	828.1	21.10	20.63
Total memory	2 002	4 014	100.00	100.00

Table 9.1 lists the dominant costs of memory in MBytes and percentage of the total for FIPWA when the number of unknowns is 500,300. Also, consideration is given to the smallest box size, so results are shown for two sizes of the smallest box of the MLFMA tree. Recall that when the box size is smaller than $\lambda/4$ the accuracy of the solution is reduced. However, the cost in memory and time to compute the matrix-vector product is much improved over larger box sizes. In each case, the direct interaction and radiation patterns use the most memory. For the smallest box size of $\lambda/10$ in Table 9.1, the direct interaction matrices and

the radiation patterns constitute 53% of the total cost in memory. When the box size is doubled to $\lambda/5$, the cost for the radiation patterns increases 11%, while the cost of the direct interaction matrices is halved. For the surface integral equation, and a spherical scatterer, the number of direct interaction matrices is reduced as the smallest box size increases.

Table 9.2 shows the largest costs in CPU time during the setup stage, and the matrix-vector product. Notice that the mat-vec time increases by more than a factor of two when the smallest box size is doubled.

It should be noted that the reason the O2I setup time is longer than the O2I-SM setup time in FIPWANOS is due to different methods of construction for the nonuniform sampling method. At the time of this report, the particular construction method for the O2I translation matrix had a higher cost, but it can be kept the same as the setup time for FIPWA. As the setup time is not the focus of this study, it is disregarded in the following results.

Table 9.2: Distribution of computation time for FIPWA. Results are for a 1-m sphere with 500,300 unknowns above a half space, and two typical sizes for the smallest box. In each case, 8 levels are used. The frequencies are 2.959 GHz and 4.197 GHz for smallest box sizes of $\lambda/10$ and $\lambda/5$, respectively.

Setup Costs	CPU Cost (CPU sec)	
	$\lambda/10$	$\lambda/5$
O2I	917.6	10 190
O2I-SM	672.1	3 272
Pole Contribution	112.1	188.4
Direct Interaction Matrix and Radiation Patterns	80 030	138 316
Mat-vec	77.3	191.2

So far, only the matrices for the radiation patterns of the leafy boxes have been considered. Yet, there is potential to save in memory and time during the set up of the O2I and O2I-SM

matrices. The leafy box matrix and O2I matrices utilize the same data structure, and the leafy box matrices are the same size as the smallest O2I and O2I-SM matrices. Thus, if the number of samples is reduced, as in the nonuniform sampling method, then the translation matrices have lowered costs of memory and setup time. In the next section, the data structure is briefly examined, and results of moderate sized problems are presented.

9.3 Uniform and Nonuniform Sampling

9.3.1 Uniform sampling

The 3-D translator has $N_s = (N_\theta - 2)(2N_\phi) + 2$ samples in $\Omega_s \in \{\theta_s \in [0, \pi], \phi_s \in [0, 2\pi]\}$, where $N_\theta = P$ and $N_\phi = 2P$ to sufficiently sample the radiation patterns. In constructing the 3-D translator, every 2-D translator uses N_ϕ samples. This is uniform sampling. It is simple to implement and convenient for allocating memory for the interpolation and translation matrices.

The method used to construct the translator with uniform sampling is based on the number of samples, N_s , needed to store the radiation pattern. The Green's function has the form

$$g(r, r') = \sum_{n=1}^{N_s} \beta_{jJ}(\Omega_s) \cdot \mathcal{T}_{JI}(\Omega_s) \cdot \beta_{Ii}(\Omega_s), \quad (9.1)$$

where the translation matrix is diagonal, and stored as an array.

This structure is efficient in both memory allocation and in the construction of the translator. Clearly, with N_ϕ fixed, the memory is allocated according to N_s . The construction is straightforward.

9.3.2 Nonuniform sampling

In nonuniform sampling, each value of θ corresponds to a different set of samples ϕ . This requires a two-dimensional array to store the sample location on the sphere, because the ϕ

values are different at each latitude. The new translation matrix is

$$\mathcal{T}_{JI} = \begin{bmatrix} \mathcal{T}_{JI}(\theta_1 = 0, 0) \\ \mathcal{T}_{JI}(\theta_2, \phi_{2,1}) \\ \mathcal{T}_{JI}(\theta_2, \phi_{2,2}) \\ \vdots \\ \mathcal{T}_{JI}(\theta_2, \phi_{2,N_\phi^2}) \\ \mathcal{T}_{JI}(\theta_3, \phi_{3,1}) \\ \vdots \\ \mathcal{T}_{JI}(\theta_3, \phi_{3,N_\phi^3}) \\ \mathcal{T}_{JI}(\theta_{N_\theta-1}, \phi_{3,1}) \\ \vdots \\ \mathcal{T}_{JI}(\theta_{N_\theta-1}, \phi_{N_{\theta-1}, N_\phi^{N_{\theta-1}}}) \\ \mathcal{T}_{JI}(\theta_{N_\theta} = \pi, 0) \end{bmatrix}. \quad (9.2)$$

There are different ways to construct this translation matrix, but if one is not careful, the construction cost can exceed $\mathcal{O}(N \log N)$.

9.4 FIPWANOS Versus FIPWA

To compare the two methods, FIPWA and FIPWANOS are used to compute the setup stage and matrix-vector product for several small to moderate-sized spheres over a lossy half space. Additionally, cases are compared for two sizes of the smallest box. This is helpful for determining tradeoffs between accuracy and efficiency.

Table 9.3: Cost of uniform sampling for smallest box size of $\lambda/10$. Results for various 1-m spheres over a half space.

	Number of unknowns			
	1 200	10 092	101 568	250 300
Frequency (GHz)	0.103	0.297	0.943	1.481
Max level	3	5	6	7
O2I (MB)	0.4	2.3	5.4	15.4
O2I-SM (MB)	2.0	12.6	90.9	215.2
Radiation Patterns (MB)	1.7	14.2	143.0	355.2
Total memory (MB)	6.1	45.6	398.4	988.1
Mat-vec (CPU sec)	0.11	1.18	14.0	37.2

9.4.1 Memory cost of FIPWANOS for smallest box size of $\lambda/10$

Cost comparison

Tables 9.3 and 9.4 list the cost to store the O2I and O2I-SM translation matrices, the leafy box radiation patterns, and the total memory for uniform (FIPWA) and nonuniform (FIPWANOS) sampling methods, respectively. In each case, the smallest box is $\lambda/10$ for small to moderate sized problems ($N < 500\,000$). Memory is measured in MBytes and the mat-vec time is measured in CPU seconds.

When the smallest box is $\lambda/10$, the costs to store the radiation patterns are identical. The reason the storage is only reduced by a small amount and the cost of the radiation patterns are equal is because of the smallest box size. As the box size becomes smaller, the number of samples for ϕ reduces to a constant. In other words, there is always a minimum number of plane waves that are needed for the radiation patterns and translation matrices. The minimum number is used at every latitude on the sphere.

Table 9.4: Cost of nonuniform sampling for smallest box size of $\lambda/10$. Results for various 1-m spheres over a half space.

	Number of unknowns			
	1 200	10 092	101 568	250 300
Frequency (GHz)	0.103	0.297	0.943	1.481
Max level	3	5	6	7
O2I (MB)	0.3	1.7	4.1	11.3
O2I-SM (MB)	1.9	11.4	84.7	200.7
Radiation Patterns (MB)	1.7	14.2	143.0	355.2
Total memory (MB)	5.8	42.8	381.2	943.2
Mat-vec (CPU sec)	0.09	0.97	11.3	29.5

Savings in memory and mat-vec time

The percentage of reduction for each item in Tables 9.3 and 9.4 is listed in Table 9.5. There is an average reduction of 19% in the mat-vec time, and an average reduction of 5% in the storage costs. The mat-vec time often constitutes a large portion of the total simulation time, so even with a small savings in memory, the savings in mat-vec time make the nonuniform sampling method worth exploring.

9.4.2 Memory cost of FIPWANOS for smallest box size of $\lambda/5$

Cost comparison

Higher accuracy typically costs more time, so it is desirable to save computational time for the case when the smallest box size is $\lambda/5$. By comparing FIPWA and FIPWANOS for this case, it becomes clear that there is potential for a large reduction in the simulation time.

Table 9.5: Percentage of reduction for smallest box size of $\lambda/10$. Results for various 1-m spheres over a half space.

Percentage of Reduction					
	Number of unknowns				Mean
	1 200	10 092	101 568	250 300	
Frequency (GHz)	0.206	0.594	1.887	2.963	
Max level	3	5	6	7	
O2I	25.00	26.09	24.07	26.62	25.45
O2I-SM	5.00	9.52	6.82	6.74	7.02
Radiation Patterns	0	0	0	0	0
Total memory	4.92	6.14	4.32	4.54	4.98
Mat-vec	18.18	17.80	19.29	20.70	18.99

Similar to the previous section, Tables 9.6 and 9.7 list the cost to store the O2I and O2I-SM translation matrices, the radiation patterns, and the direct interaction matrices for cases when the smallest box is $\lambda/5$. The mat-vec time is also given. The difference in memory and time is very noticeable.

Savings in memory and mat-vec time

Previous estimates of the memory savings were approximately 20%, Table 9.8, page 126, shows that the average reduction in storage for the case when the smallest box dimension is $a = \lambda/5$ is 22%, which agrees with the earlier estimates. Yet, the average reduction in mat-vec time is 31%. As a matter of perspective, with a smallest box size of $a = \lambda/5$ for $N = 1\,000\,000$, the simulation can take approximately 80 hours to complete, and cost nearly 8 GBytes of RAM. With the non-uniform sampling method, the same problem can be solved in nearly the same time and cost as the case where $a = \lambda/10$ (approximately 60 hours with

4 GBytes of RAM). The benefit is that the setup with a larger size of a results in a more accurate solution.

Table 9.6: Cost of uniform sampling with 3-D FIPWA when smallest box size is $\lambda/5$. Results for various 1-m spheres over a half space.

	Number of unknowns			
	1 200	10 092	101 568	250 300
Frequency (GHz)	0.206	0.594	1.887	2.963
Max level	3	5	6	7
O2I (MB)	1.0	5.3	15.3	50.5
O2I-SM (MB)	3.4	19.3	130.6	320.2
Radiation Patterns (MB)	4.5	37.5	377.0	936.5
Total memory (MB)	11.7	86.9	769.7	1 949
Mat-vec (CPU sec)	0.23	2.55	31.5	87.2

Table 9.7: Cost of nonuniform sampling with 3-D FIPWA when smallest box size is $\lambda/5$. Results for various 1-m spheres over a half space.

	Number of unknowns			
	1 200	10 092	101 568	250 300
Frequency (GHz)	0.206	0.594	1.887	2.963
Max level	3	5	6	7
O2I (MB)	0.7	4.0	11.1	34.5
O2I-SM (MB)	2.6	15.9	112.3	271.7
Radiation Patterns (MB)	3.2	27.1	273.0	678.1
Total memory (MB)	9.0	68.2	604.2	1 519
Mat-vec (CPU sec)	0.15	1.80	22.2	61.1

Table 9.8: Percentage of reduction for smallest box size of $\lambda/5$. Results for various 1-m spheres over a half space.

Percentage of Reduction					
	Number of unknowns				Mean
	1 200	10 092	101 568	250 300	
Frequency (GHz)	0.206	0.594	1.887	2.963	
Max level	3	5	6	7	
O2I	30.00	24.53	27.45	31.68	28.42
O2I-SM	23.53	17.62	14.01	15.15	17.58
Radiation Patterns	28.89	27.73	27.59	27.59	27.95
Total memory	23.08	21.52	21.50	22.06	22.04
Mat-vec	34.78	29.41	29.52	29.93	30.91

It should be noted that the lower sampling rate near the poles may need to be accounted for in the interlevel interpolation process. Otherwise, the accuracy in the mat-vec product may degrade and the iterative solution could require more iterations. This would be self-defeating. However, the results suggest that it is worth exploring.

9.5 Summary

It was shown that the bottle-neck in memory and setup time of the fast algorithm is the radiation patterns at the leafy level. By reducing the number of samples needed to store the translation matrices and radiation patterns with nonuniform sampling of the $\theta - \phi$ pairs, the total storage, setup time, and time per matrix-vector product are reduced up to 30%. While complete solutions are not presented, it is clear that there is a large savings in memory, setup time, and time to compute the matrix-vector product.

CHAPTER 10

CONCLUSIONS

10.1 Conclusions

The multilevel multipole-free fast algorithm was presented with $\mathcal{O}(N \log N)$ cost in memory and processing time in the iterative solver, making it on par with other multilevel fast algorithms. The new algorithm evolves directly from FIPWA for layered media, where the 3-D translators are constructed with a multipole expansion of the 2-D Green's function. In the new algorithm, the 2-D FIPWA, for complex media, replaces the multipole expansion to form a completely multipole-free fast algorithm called the multipole-free fast inhomogeneous plane-wave algorithm (MF-FIPWA). MF-FIPWA was validated by comparing the results of a benchmark problem with results from FIPWA and a full matrix solver.

In addition, FIPWA was compared to MF-FIPWA in terms of setup memory and CPU costs, and accuracy. Various scattering problems were studied to reveal when MF-FIPWA outperforms FIPWA and vice-versa. It was shown that without optimization, MF-FIPWA has lower setup time for large boxes in the tree, and FIPWA has lower setup time for small boxes in the tree. The accuracies are comparable.

To solve large scale problems, various debugging methods were used and discussed to aid future developers of MF-FIPWA or other codes. The debugging routines gave a new look

at the 2-D FIPWA translator that was summarily substituted as a black box for the 2-D FMA translator. Upon exploring alternative methods of implementing MF-FIPWA, so as to reduce setup time, it became apparent that additional savings could be obtained for both FIPWA and MF-FIPWA by using nonuniform sampling.

The nonuniform sampling method was briefly studied to gauge the potential savings in memory, setup time, and time to compute the matrix-vector product. Substantial savings of up to 30% can be achieved with nonuniform sampling, making the method an excellent topic of further research.

10.2 Future Work

Implementation of MF-FIPWA was challenging even with a black box approach. Physical differences exist between the two mathematical formulations that require alternative methods for interpolating and extrapolating to the steepest descent paths and computing highly accurate numerical integration. Future work with MF-FIPWA should explore novel methods to perform both, and also to implement the nonuniform sampling method. Storage of the radiation patterns of the leafy boxes and computing the matrix vector product constitute the majority of the simulation cost and storage requirements. Hence, there is potential to make large improvements to fast algorithms in general by using the nonuniform sampling methods for interpolation between levels in the tree.

APPENDIX A

SPECIAL INTERPOLATION FUNCTION

The interpolation function is derived for the 2-D translator in MF-FIPWA.

Given the function $g(t)$ that is band-limited to $-M < \omega < M$, $g(t)$ can be reconstructed exactly when convolved with the sinc function or the Dirac delta function as

$$g(t) = \int_{-\infty}^{\infty} d\tau g(\tau) I(\tau - t), \quad (\text{A.1})$$

where $I(t) = \frac{M}{\pi} \text{sinc} Mt = \frac{M}{\pi} \frac{\sin(Mt)}{Mt}$, or $I(t) = \delta(t)$.

If $g(t)$ is convolved with a Dirac comb function with N_L spacings in the period L , it can be made to be periodic with period L as

$$g_L(t) = \int_{-\infty}^{\infty} d\tau g(\tau) T_L(\tau - t), \quad (\text{A.2})$$

where $g_L(t)$ represents the periodic form of $g(t)$, $T_L(t) = \sum_{p=-\infty}^{\infty} \delta(t - pl)$, $l = L/N_L$, and $p \in \mathbb{Z}$.

The special interpolation function must be periodic so that

$$g_L(t) = \int_{-\infty}^{\infty} d\tau g(\tau) I_L(t - \tau), \quad (\text{A.3})$$

and is obtained by substituting (A.1) into (A.2) to form the double convolution

$$g_L(t) = \int_{-\infty}^{\infty} d\tau \left[\int_{-\infty}^{\infty} ds g(s) I(s - \tau) \right] T_L(\tau - t), \quad (\text{A.4})$$

$$= \int_{-\infty}^{\infty} d\tau \left[\int_{-\infty}^{\infty} ds g(s) I(s - \tau) \right] \sum_{p=-\infty}^{\infty} \delta(\tau - pl). \quad (\text{A.5})$$

The order of convolution is easily swapped in the Fourier space with the convolution theorem

$$g_L(t) = g(t) * I(t) * T_L(t), \quad (\text{A.6})$$

$$= \mathcal{F}.\mathcal{T}^{-1} \left\{ \tilde{g}(\omega) \tilde{I}(\omega) \tilde{T}_L(\omega) \right\}, \quad (\text{A.7})$$

$$= \mathcal{F}.\mathcal{T}^{-1} \left\{ \tilde{g}(\omega) \tilde{I}_L(\omega) \right\}, \quad (\text{A.8})$$

where $\tilde{g}(\omega)$, $\tilde{I}(\omega)$, $\tilde{I}_L(\omega)$, and $\tilde{T}_L(\omega)$ represent the Fourier transforms of $g(t)$, $I(t)$, $I_L(t)$ and $T_L(t)$, respectively.

In (A.8), $\tilde{I}_L(\omega) = \tilde{I}(\omega) \tilde{T}_L(\omega)$, for which the Fourier transforms are well known as

$$\tilde{I}(\omega) = \begin{cases} 1, & -M < \omega < M \\ 0, & \text{otherwise} \end{cases}, \quad (\text{A.9})$$

$$\tilde{T}_L(\omega) = \frac{1}{l} \sum_{m=-\infty}^{\infty} \delta(\omega - m/l). \quad (\text{A.10})$$

Starting with the inverse Fourier transform

$$I_L(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega e^{-i\omega t} \tilde{I}(\omega) \tilde{T}_L(\omega), \quad (\text{A.11})$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega e^{-i\omega t} \frac{1}{l} \sum_{m=-N}^N \delta(\omega - m/l), \quad (\text{A.12})$$

where $N = \lfloor lM \rfloor$, and $\lfloor \cdot \rfloor$ rounds the argument down to the nearest integer, the inverse Fourier transform of $\tilde{T}_L(\omega)$ results in

$$I_L(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega e^{-i\omega t} \tilde{I}_L(\omega), \quad (\text{A.13})$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega e^{-i\omega t} \frac{1}{l} \sum_{n=-N}^N \delta(\omega - n/l), \quad (\text{A.14})$$

$$= \frac{1}{2\pi l} \sum_{n=-N}^N \int_{-\infty}^{\infty} d\omega e^{-i\omega t} \delta(\omega - n/l), \quad (\text{A.15})$$

$$= \frac{1}{\pi l} \sum_{n=-N}^N \frac{1}{2} e^{-i \frac{n}{l} t}. \quad (\text{A.16})$$

The sum is recognized as a geometric series with the closed form solution

$$\sum_{n=-N}^N \frac{1}{2} e^{-i \frac{n}{l} t} = \sum_{n=-N}^N \frac{1}{2} \left(e^{-i \frac{t}{l}} \right)^n \quad (\text{A.17})$$

$$= \frac{1}{2} + \sum_{n=1}^N \cos \left(n \frac{t}{l} \right), \quad (\text{A.18})$$

$$= \frac{\sin \left[(2N+1) \frac{t}{2l} \right]}{2 \sin \left(\frac{t}{2l} \right)}. \quad (\text{A.19})$$

Hence,

$$I_L(t) = \frac{1}{2\pi l} \frac{\sin \left[(2N+1) \frac{t}{2l} \right]}{\sin \left(\frac{t}{2l} \right)}, \quad (\text{A.20})$$

where $N = \lfloor lM \rfloor$, $l = L/N_L$, and L and M are the period and single-sided bandwidth of $g(t)$, respectively. The form of $I_L(t)$ is sometimes called a digital sinc, or periodic sinc function [41–43].

APPENDIX B

COMPARATIVE RESULTS OF SCALING STUDY

This appendix presents the set of data used in the scaling study of Chapters 5 and 6. The data is created with FIPWA and MF-FIPWA for a 1-meter sphere at various frequencies. The spheres are always located 0.2 m above the half space with $\epsilon = (6.5, 0.6)$. In Figures B.1–B.6, the smallest box size is $a = 0.10\lambda$. Data in Figs. B.7–B.10 are the cases when the smallest box size is $a = 0.20\lambda$. When a is smaller, the simulation requires more memory, but it produces more accurate results.

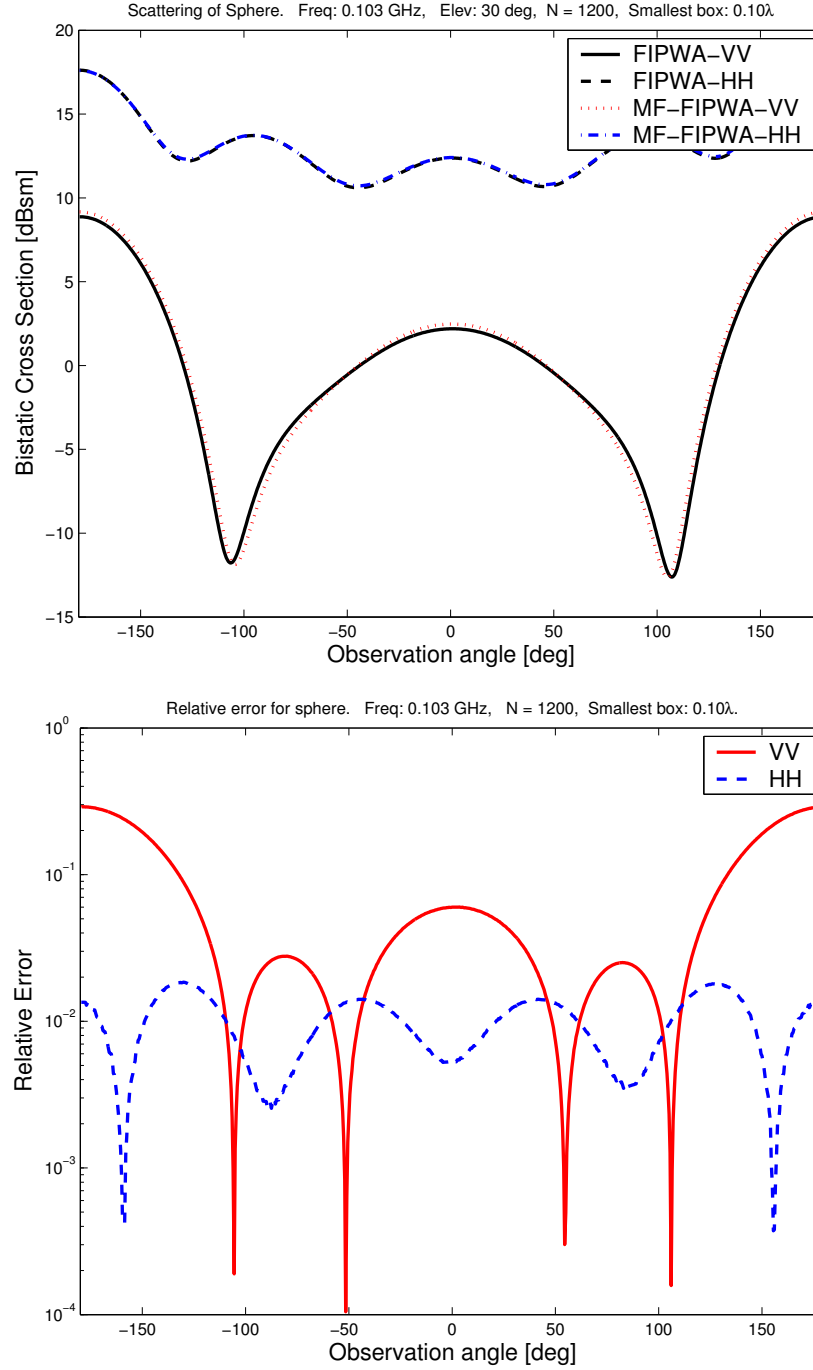


Figure B.1: Bistatic scattering and relative error of sphere at 0.103 GHz. Top: Scattering solution of FIPWA and MF-FIPWA. Bottom: Relative error of MF-FIPWA with FIPWA. The smallest box size is 0.10 λ .

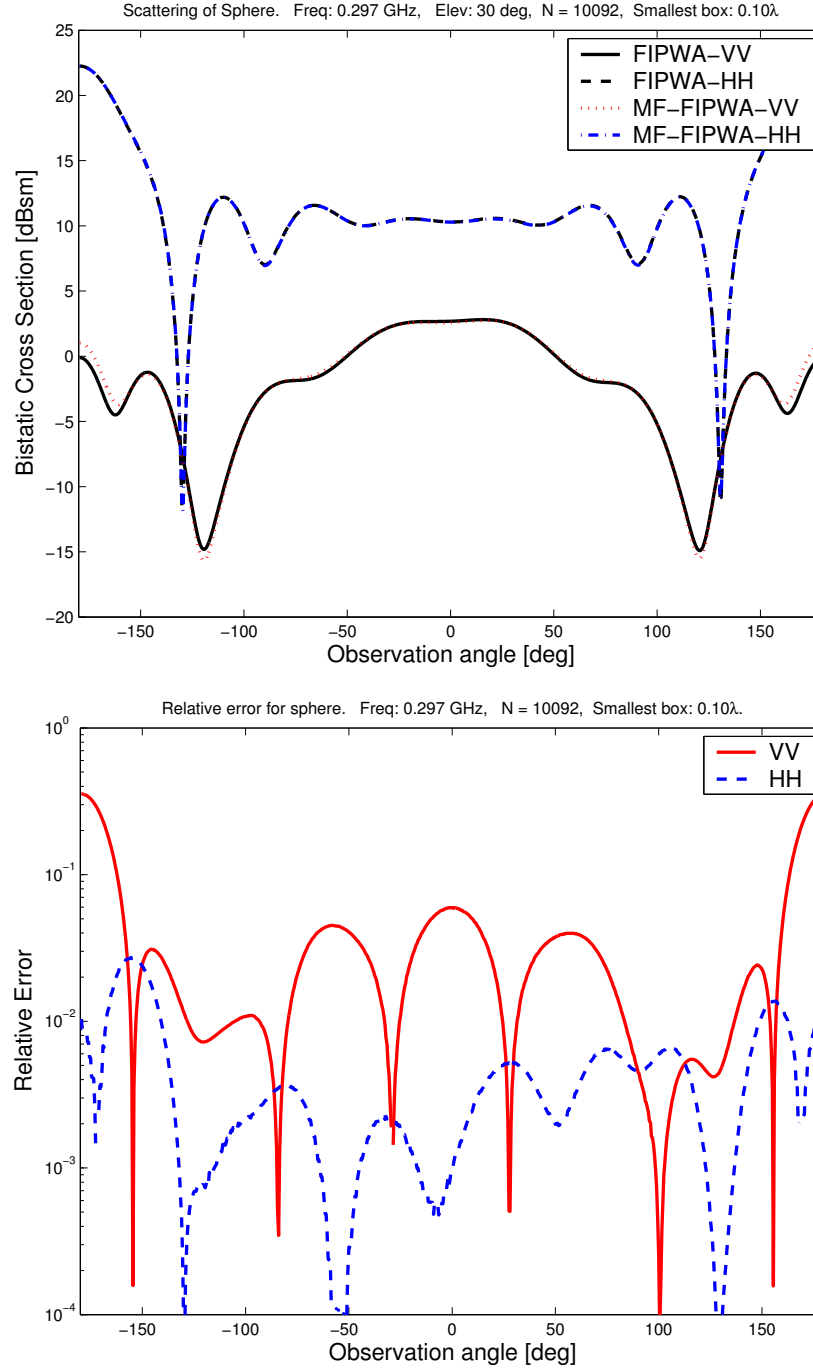


Figure B.2: Bistatic scattering and relative error of sphere at 0.297 GHz. Top: Scattering solution of FIPWA and MF-FIPWA. Bottom: Relative error of MF-FIPWA with FIPWA. The smallest box size is 0.10 λ .

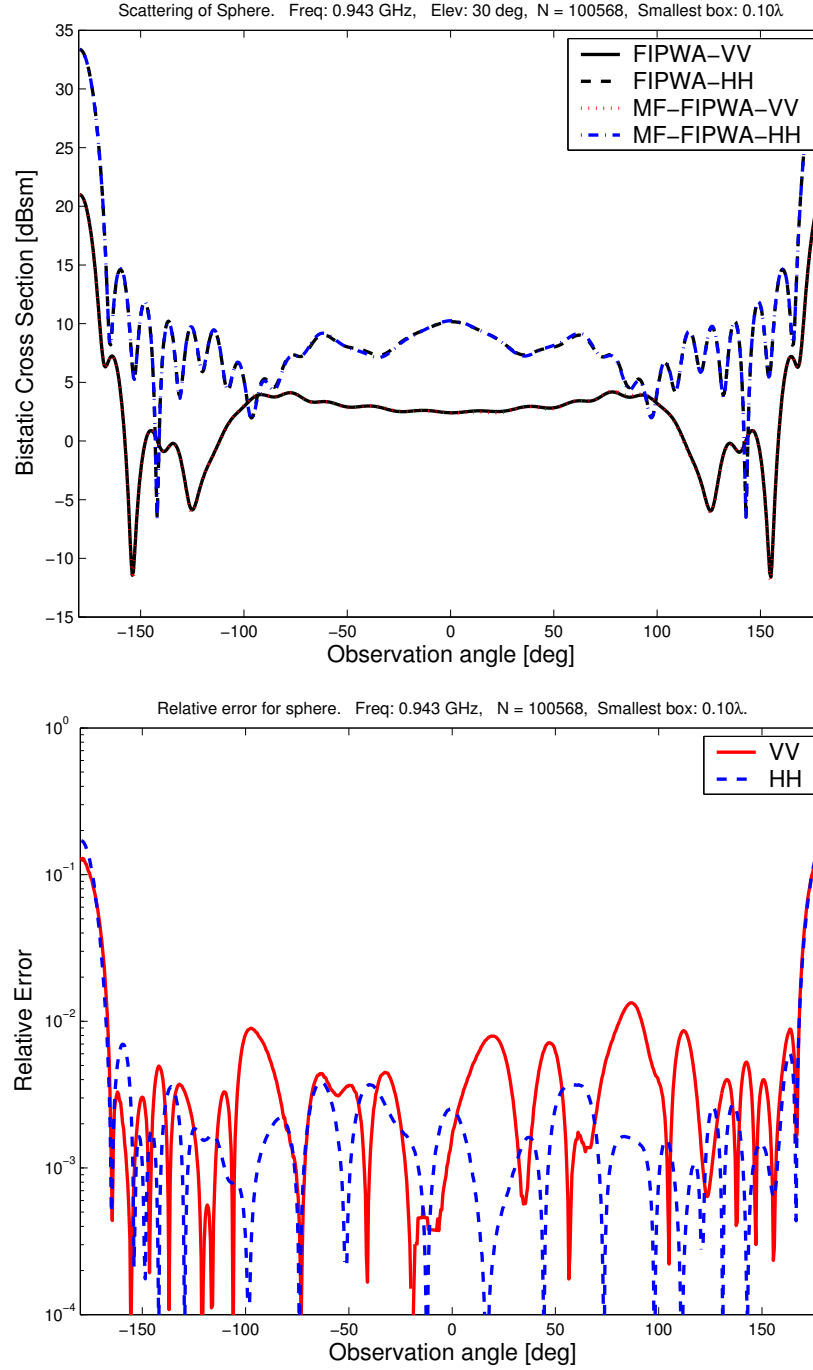


Figure B.3: Bistatic scattering and relative error of sphere at 0.943 GHz. Top: Scattering solution of FIPWA and MF-FIPWA. Bottom: Relative error of MF-FIPWA with FIPWA. The smallest box size is 0.10 λ .

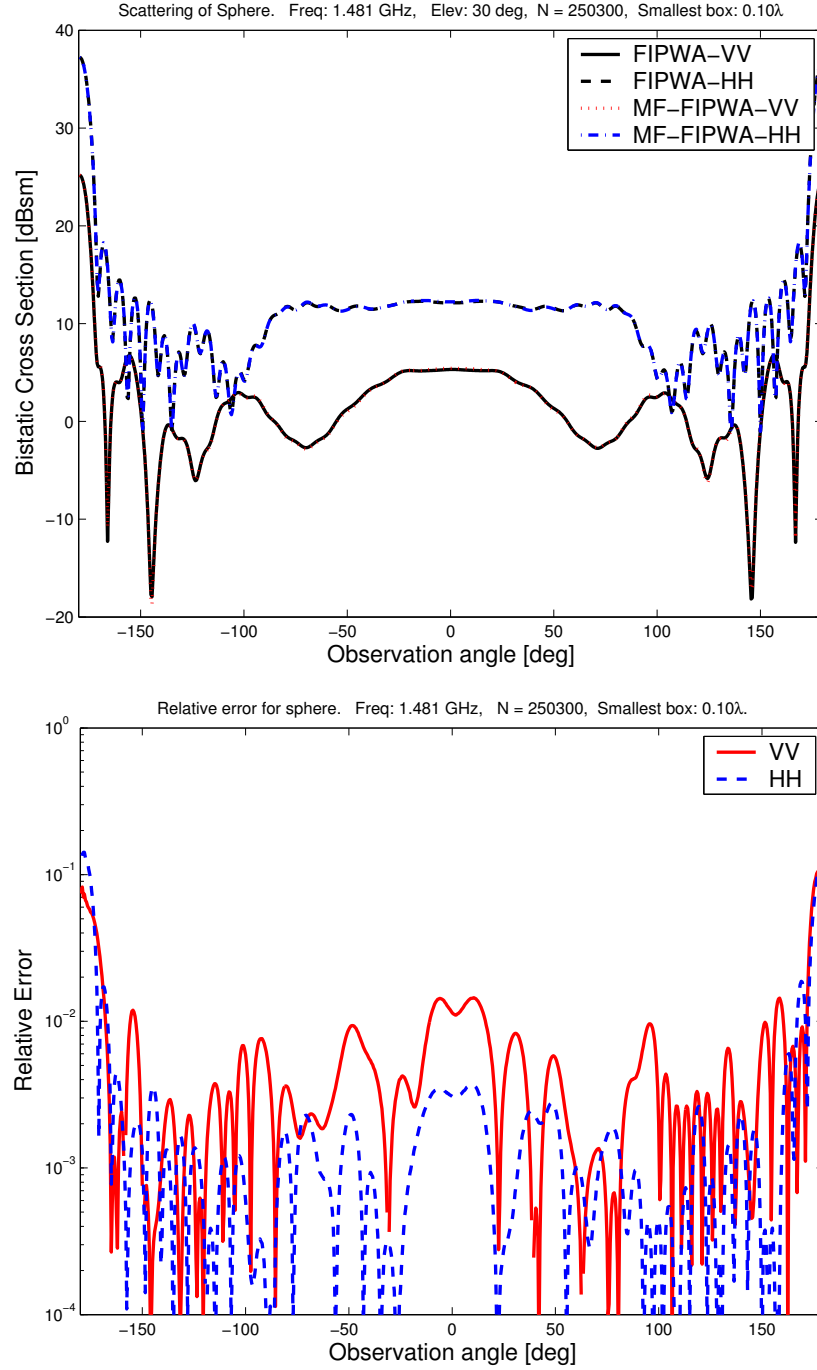


Figure B.4: Bistatic scattering and relative error of sphere at 1.481 GHz. Top: Scattering solution of FIPWA and MF-FIPWA. Bottom: Relative error of MF-FIPWA with FIPWA. The smallest box size is 0.10 λ .

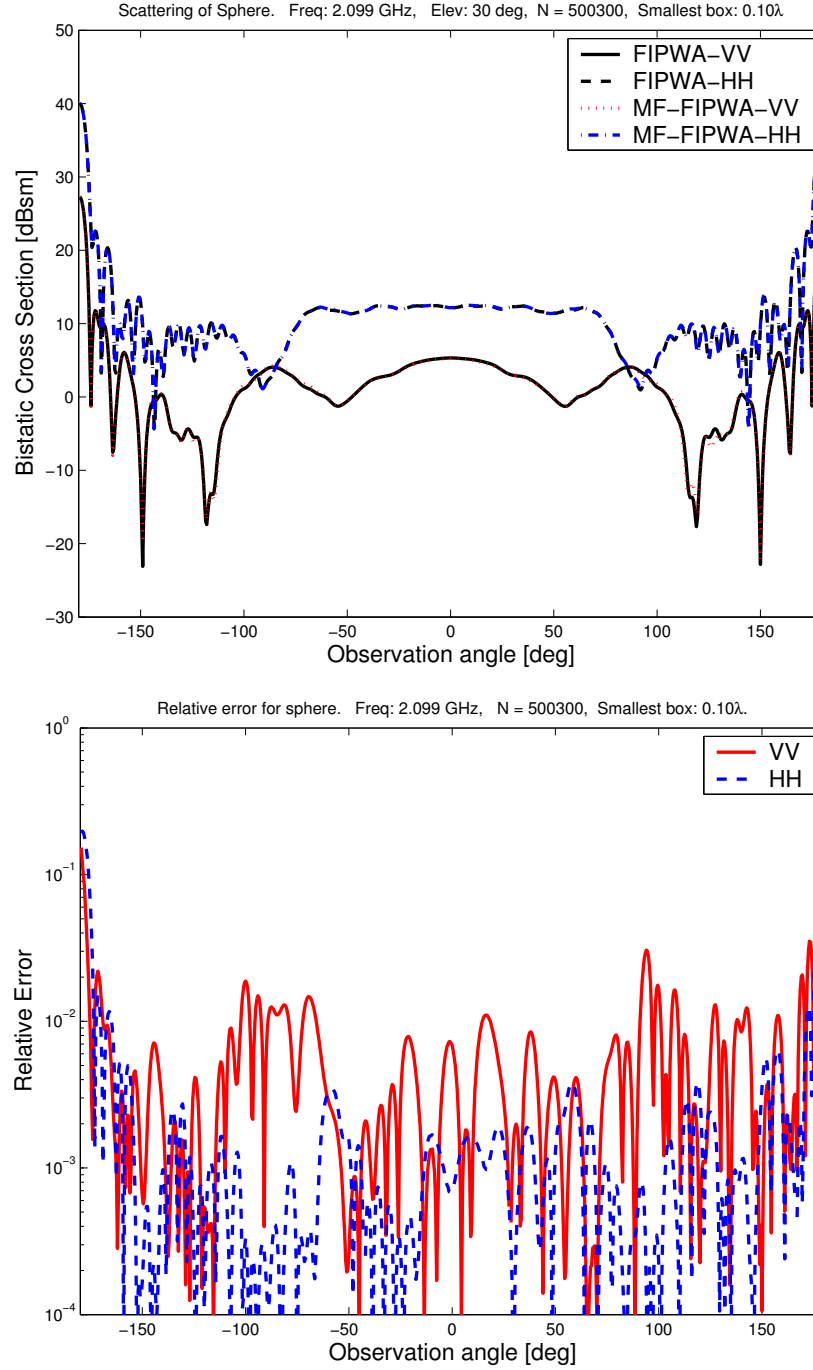


Figure B.5: Bistatic scattering and relative error of sphere at 2.099 GHz. Top: Scattering solution of FIPWA and MF-FIPWA. Bottom: Relative error of MF-FIPWA with FIPWA. The smallest box size is 0.10 λ .

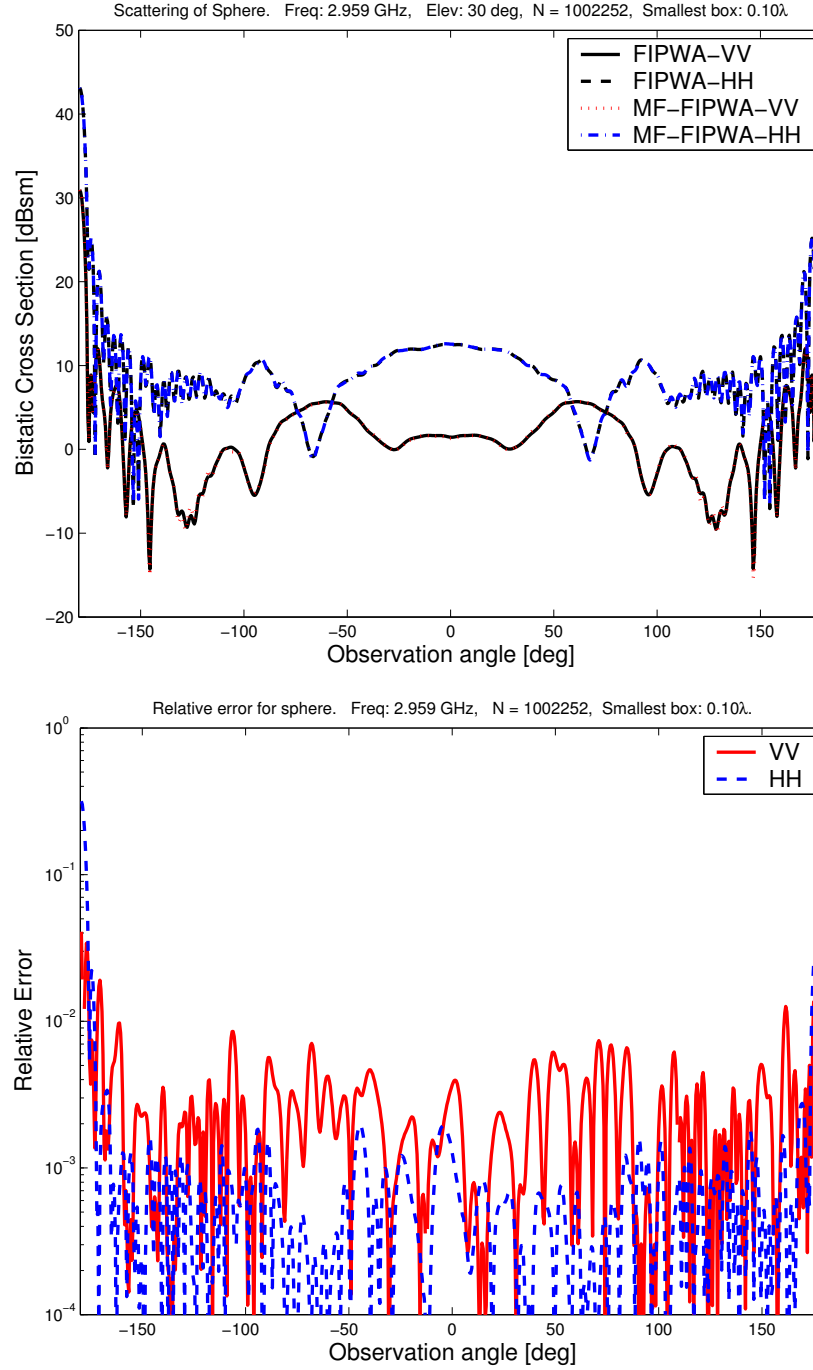


Figure B.6: Bistatic scattering and relative error of sphere at 2.959 GHz. Top: Scattering solution of FIPWA and MF-FIPWA. Bottom: Relative error of MF-FIPWA with FIPWA. The smallest box size is 0.10 λ .

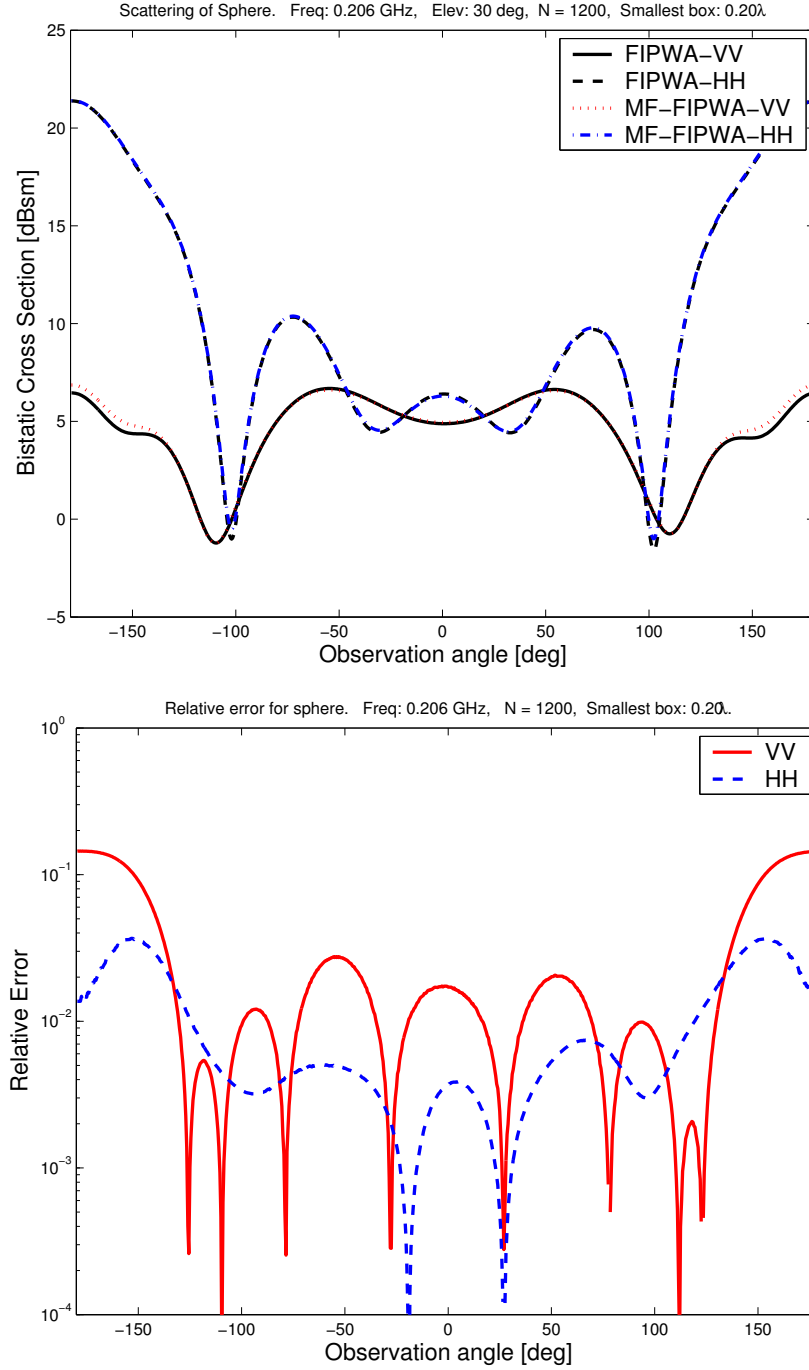


Figure B.7: Bistatic scattering and relative error of sphere at 0.206 GHz. Top: Scattering solution of FIPWA and MF-FIPWA. Bottom: Relative error of MF-FIPWA with FIPWA. The smallest box size is 0.20 λ .

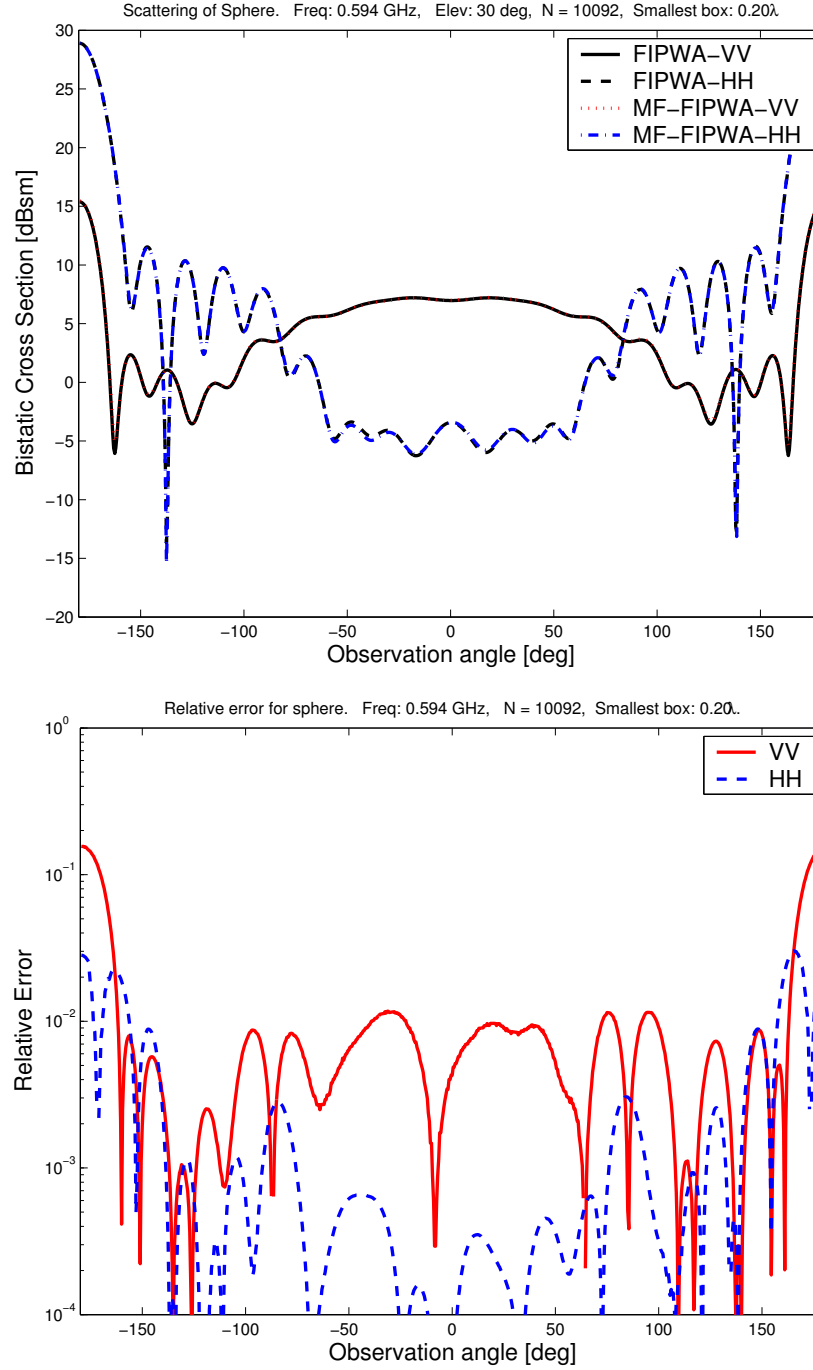


Figure B.8: Bistatic scattering and relative error of sphere at 0.594 GHz. Top: Scattering solution of FIPWA and MF-FIPWA. Bottom: Relative error of MF-FIPWA with FIPWA. The smallest box size is 0.20 λ .

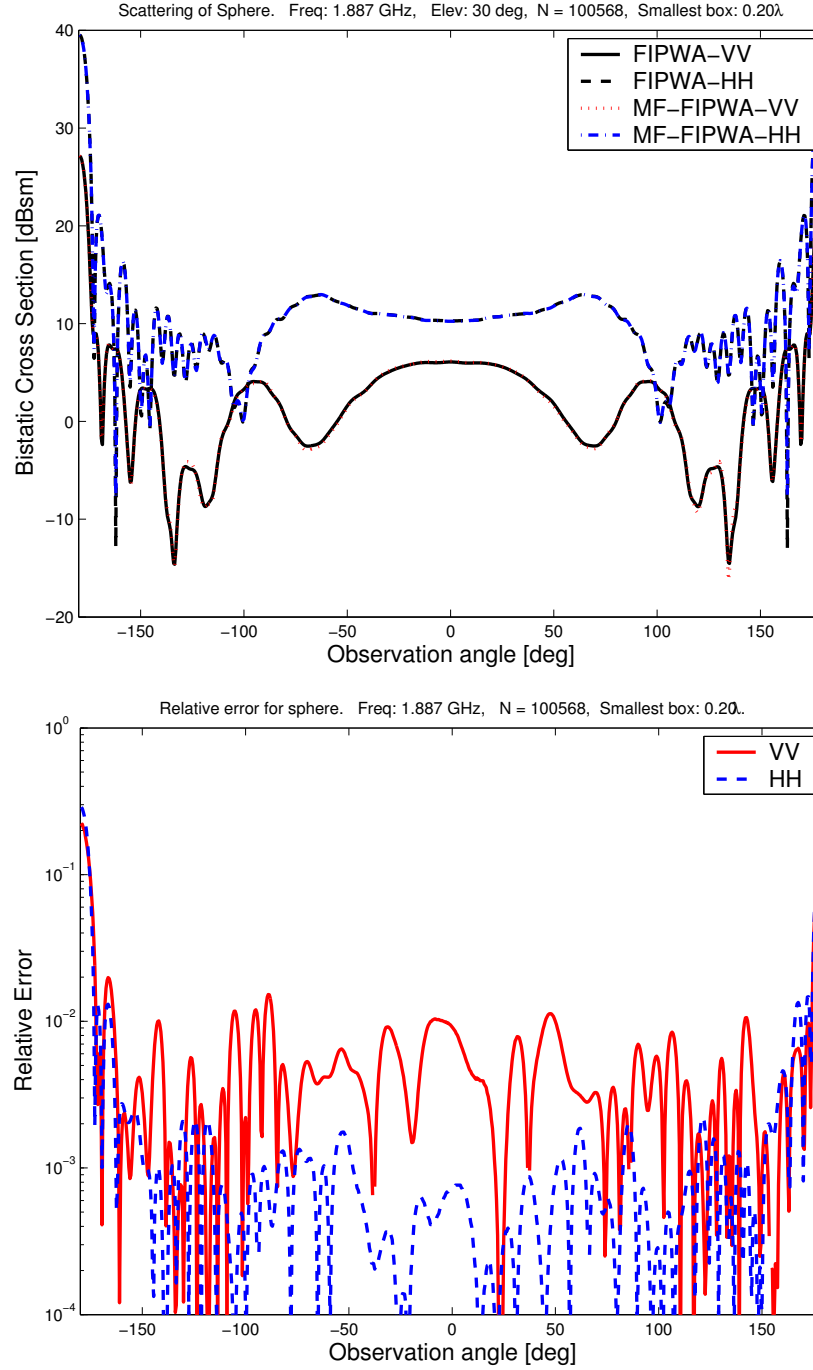


Figure B.9: Bistatic scattering and relative error of sphere at 1.887 GHz. Top: Scattering solution of FIPWA and MF-FIPWA. Bottom: Relative error of MF-FIPWA with FIPWA. The smallest box size is 0.20 λ .

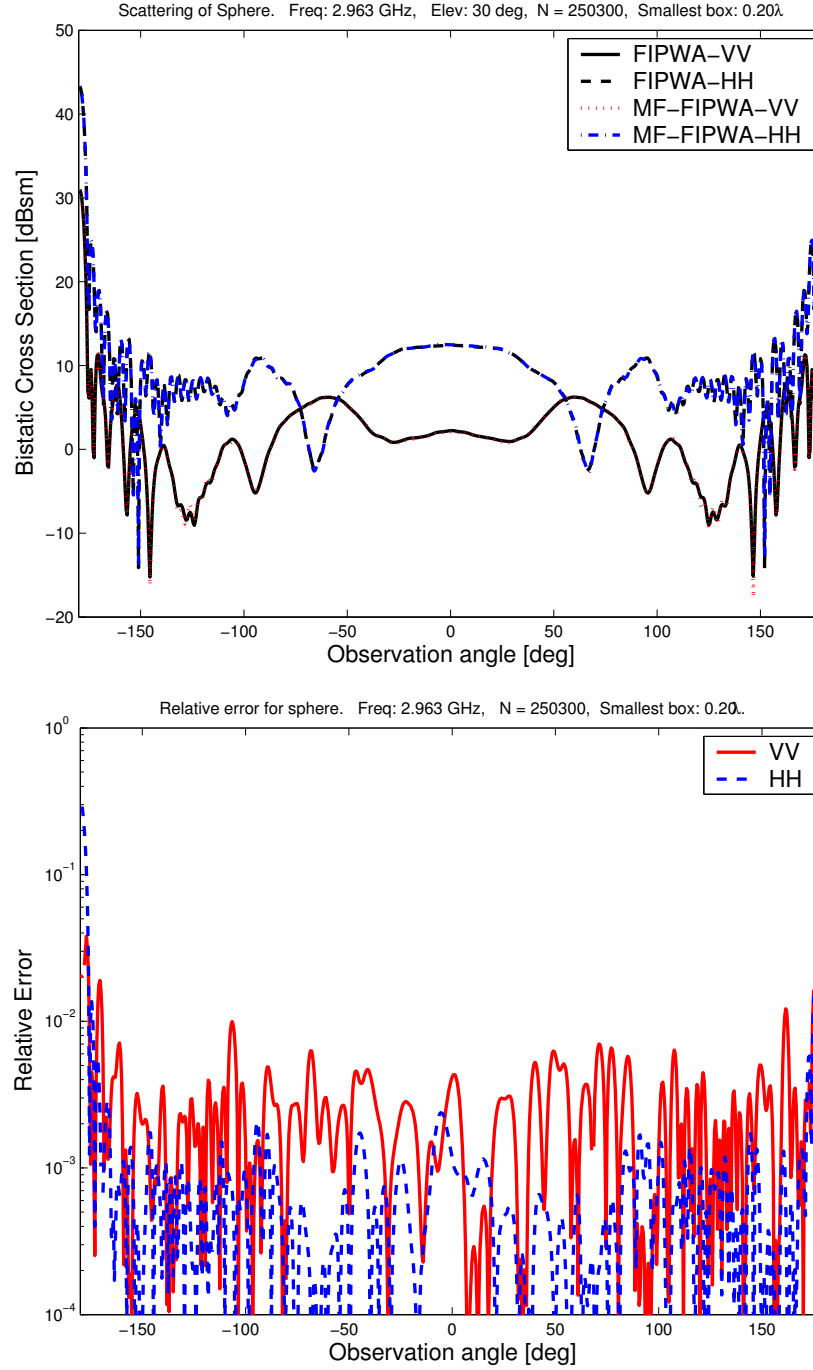


Figure B.10: Bistatic scattering and relative error of sphere at 2.963 GHz. Top: Scattering solution of FIPWA and MF-FIPWA. Bottom: Relative error of MF-FIPWA with FIPWA. The smallest box size is 0.20 λ .

REFERENCES

- [1] W. C. Chew, *Waves and Fields in Inhomogeneous Media*. Piscataway, NJ: IEEE Press, 1995.
- [2] V. Rokhlin, “Rapid solution of integral equations of scattering theory in two dimensions,” *J. Comput. Phys.*, vol. 86, pp. 414–439, 1990.
- [3] L. Greengard and V. Rokhlin, “A fast algorithm for particle simulations,” *J. Comput. Phys.*, vol. 73, pp. 325–348, 1987.
- [4] R. Coifman, V. Rokhlin, and S. Wandzura, “The fast multipole method for the wave equation a pedestrian prescription,” *IEEE Trans. Magn.*, vol. 35, no. 3, pp. 7–12, June 1993.
- [5] C. C. Lu and W. C. Chew, “Fast algorithm for solving hybrid integral equations,” *IEEE, Proceedings-H*, vol. 140, no. 6, Dec. 1993.
- [6] —, “A multilevel algorithm for solving boundary-value scattering,” *Microwave and Optical Technology Letters*, vol. 7, no. 10, pp. 466–470, July 1994.
- [7] J. M. Song and W. C. Chew, “Multilevel fast multipole algorithm for solving combined field integral equations of electromagnetic scattering,” *Microwave and Optical Technology Letters*, vol. 10, no. 1, pp. 14–19, Sept. 1995.
- [8] J. M. Song, C. C. Lu, and W. C. Chew, “Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects,” *IEEE Trans. Antennas Propagat.*, vol. 45, no. 10, pp. 1488–1493, Oct. 1997.
- [9] R. F. Harrington, *Field Computation by Moment Method*. Piscataway, NJ: IEEE Press, 1993.
- [10] B. Hu, W. C. Chew, and S. Velampambil, “Fast inhomogeneous plane wave algorithm (FIPWA) for analysis of electromagnetic scattering,” *Radio Science*, vol. 36, no. 6, pp. 1327–1340, 2001.
- [11] A. Brandt, “Multilevel computations of integral transforms and particle interactions with oscillatory kernels,” *Comp. Phys. Comm.*, vol. 65, pp. 24–38, 1991.

- [12] L. J. Jiang and W. C. Chew, "The mixed-form fast multipole algorithm for broadband electromagnetic simulations," *IEEE Antennas and Propagation Symposium, Washington DC*, July 2005.
- [13] —, "Low frequency fast inhomogeneous plane wave algorithm (LF-FIPWA)," *Microwave Opt. Technol. Lett.*, vol. 40, no. 2, pp. 117–122, 2004.
- [14] B. Hu and W. C. Chew, "Fast inhomogeneous plane wave algorithm for scattering from objects above the multi-layered medium," *IEEE Trans. Geosci. Remote Sensing*, vol. 39, no. 5, pp. 1028–1038, 2001.
- [15] N. Geng, A. Sullivan, and L. Carin, "Multilevel fast-multipole algorithm for scattering from conducting targets above or embedded in a lossy half-space," *IEEE Trans. Geosci. Remote Sensing*, vol. 38, no. 4, pp. 1561–1573, July 2000.
- [16] L. Li, Z. Liu, Z. Dong, J. Thompson, and L. Carin, "Scalable multilevel fast multipole method for multiple targets in the vicinity of a half space," *IEEE Trans. Geosci. Remote Sensing*, vol. 41, no. 4, pp. 791–802, Apr. 2003.
- [17] W. C. Chew, J. Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*. Norwood, MA: Artech House, Inc., 2000.
- [18] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propagat.*, vol. 30, no. 3, pp. 409–418, Mar. 1982.
- [19] J. Zhao, W. C. Chew, C. C. Lu, E. Michielssen, and J. Song, "Thin-stratified medium fast-multipole algorithm for solving microstrip structures," *IEEE Trans. Microwave Theory Tech.*, vol. 46, no. 4, pp. 395–403, Apr. 1998.
- [20] M. A. Saville and W. C. Chew, "Error control for 2D FIPWA in complex, homogeneous media," *Journal of Electromagnetic Waves and Applications*, vol. 20, no. 5, pp. 567–581, 2006.
- [21] B. Hu, "Fast inhomogeneous plane wave algorithm for electromagnetic scattering problems," Ph.D. dissertation, University of Illinois at Urbana-Champaign, Department of Electrical and Computer Engineering, 2001.
- [22] B. Hu, W. C. Chew, E. Michielssen, and J. Zhao, "Fast inhomogeneous plane wave algorithm for the fast analysis of two-dimensional scattering problems," *Radio Science*, vol. 34, no. 4, pp. 759–772, July-Aug. 1999.
- [23] B. Hu and W. C. Chew, "Fast inhomogeneous plane wave algorithm for electromagnetic solutions in layered medium structures: Two-dimensional scattering problems," *Radio Science*, vol. 35, no. 1, pp. 31–43, Jan.-Feb. 2000.

- [24] E. Michielssen and W. C. Chew, "Fast steepest descent path algorithm for analyzing scattering from two-dimensional objects," *Radio Science*, vol. 31, no. 5, pp. 1215–1224, Sept.-Oct. 1996.
- [25] L. Greengard, J. Huang, V. Rokhlin, and S. Wandzura, "Accelerating fast multipole methods for the Helmholtz equation at low frequencies," *IEEE Comput. Sci. Eng.*, vol. 5, pp. 32–38, July-Sept. 1998.
- [26] E. Darve and P. Havé, "A fast multipole method for Maxwell equations stable at all frequencies," *Phil. Trans. Royal Soc. Lon., A*, vol. 362, no. 1816, pp. 603–628, Mar. 2004.
- [27] H. Wallén and J. Sarvas, "Translation procedures for broadband MLFMA," *Progress in Electromagnetics Research*, vol. 55, pp. 47–78, 2005.
- [28] S. Ohnuki and W. C. Chew, "A study of the error controllability of the fast inhomogeneous plane wave algorithm for a 2-D free space case," *Antennas and Propagation Society International Symposium, 2003*, pp. 326–329, June 2003.
- [29] —, "Error analysis of the fast inhomogeneous plane wave algorithm for 2-D free space cases," *Microwave Opt. Technol. Lett.*, vol. 38, no. 4, Aug. 2003.
- [30] J. Song and W. C. Chew, "Error analysis for the truncation of multipole expansion of vector Green's functions," *IEEE Microwave Wireless Compon. Lett.*, vol. 11, no. 7, July 2001.
- [31] M. Saville and W. C. Chew, "Error control of the 2-D fast inhomogeneous plane wave algorithm for complex media," Center for Computational Electromagnetics and Electromagnetics Laboratory, University of Illinois at Urbana-Champaign, Tech. Rep. 0805, Oct. 2005.
- [32] E. Jorgensen, O. S. Kim, P. Meincke, and O. Breinbjerg, "Higher order hierarchical discretization scheme for surface integral equations for layered media," *IEEE Trans. Geosci. Remote Sensing*, vol. 42, no. 4, pp. 764–772, Apr. 2004.
- [33] N. Geng, A. Sullivan, and L. Carin, "Fast multipole method for scattering from 3-D PEC targets situated in a half-space environment," *Microwave and Optical Technology Letters*, vol. 21, no. 6, pp. 399–405, June 1999.
- [34] W. C. Chew, J. L. Xiong, and M. A. Saville, "A matrix-friendly formulation of layered medium green's function," Accepted for publication in *Journal of Antennas and Wireless Propagation Letters*, 2006.
- [35] M. L. Hastriter, S. Ohnuki, and W. C. Chew, "Error control of the translation operator in 3D MLFMA," *Microwave Opt. Technol. Lett.*, vol. 37, no. 3, May 2003.

- [36] W. C. Chew, J. C. Song, T. J. Cui, S. Velamparambil, M. L. Hastriter, and B. Hu, "Review of large scale computing in electromagnetics with fast integral equation solvers," *Comput. Model. Eng. Sci. (USA)*, vol. 5, no. 4, pp. 361–372, 2004.
- [37] M. A. Saville and W. C. Chew, "Multilevel multipole free fast algorithm for layered media," *IEEE International Symposium and USNC/URSI National Radio Science Meeting*, vol. 5, July 2006.
- [38] O. M. Bucci, C. Gennarelli, and C. Savarese, "Optimal interpolation of radiation fields over a sphere," *IEEE Trans. Antennas Propagat.*, vol. 39, pp. 1633–1643, 1991.
- [39] W. C. Chew, "Personal notes on FIPWA versus MF-FIPWA," July 2006.
- [40] M. A. Saville, "Draft user manual for the multipole-free fast inhomogeneous plane-wave algorithm," Sept. 2006.
- [41] T. Schanze, "Sinc interpolation of discrete periodic signals," *IEEE Trans. Signal Processing*, vol. 43, no. 6, pp. 1502–1503, June 1995.
- [42] F. Candocia and J. C. Principe, "Comments on sinc interpolation of discrete periodic signals," *IEEE Trans. Signal Processing*, vol. 46, no. 7, pp. 2044–2047, July 1998.
- [43] S. Dooley and A. K. Nandi, "Notes on the interpolation of discrete periodic signals using sinc function related approaches," *IEEE Trans. Signal Processing*, vol. 48, no. 4, pp. 1201–1203, Apr. 2000.
- [44] M. A. Saville and W. C. Chew, "Error control for the 3-D FIWPA in complex media," *IEEE International Symposium and USNC/URSI National Radio Science Meeting*, vol. 5, July 2005.

AUTHOR'S BIOGRAPHY

Michael Andrew Saville received the B.S.E.E. degree in 1997 from Texas A&M University, and the M.S.E.E. from Air Force Institute of Technology in 2000. As a 21-year veteran of the United States Air Force, he has worked as a ground radar technician, calibration laboratory technician, electronic warfare and systems engineer, and team leader for various government and contractor personnel. Since 2003, he has been a Ph.D. student at the University of Illinois at Urbana-Champaign, and upon graduating, he will be reassigned to the Air Force Institute of Technology as an assistant professor of electrical engineering.