

FAST MULTIPOLE METHOD FOR THE SOLUTION OF ELECTROMAGNETIC SCATTERING PROBLEMS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Özgür Salih Ergül

June 2003

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Levent Gürel (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Ergin Atalar

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Vakur B. Ertürk

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

FAST MULTIPOLE METHOD FOR THE SOLUTION OF ELECTROMAGNETIC SCATTERING PROBLEMS

Özgür Salih Ergül

M.S. in Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Levent Gürel

June 2003

The fast multipole method (FMM) is investigated in detail for the solution of electromagnetic scattering problems involving arbitrarily shaped three-dimensional conducting surfaces. This method is known to reduce the computational complexity and the memory requirement of the solution without sacrificing the accuracy. Therefore, it achieves the solution of large problems with less computational resources compared to the other traditional solution algorithms. However, the expected efficiency of the FMM may not be obtained unless the appropriate choices of the components are made. The types of the employed integral equation, iterative algorithm, and preconditioning technique directly affect the efficiency of the implementations. Performances of these components are also related to each other, and their simultaneous optimization creates a challenging task in the design of an efficient solver.

Keywords: Fast multipole method (FMM), electromagnetic scattering, integral equations, iterative algorithms, preconditioning techniques.

ÖZET

ELEKTROMANYETİK SAÇINIM PROBLEMLERİ İÇİN HIZLI ÇOKKUTUP YÖNTEMİ

Özgür Salih Ergül

Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans

Tez Yöneticisi: Doç. Dr. Levent Gürel

Haziran 2003

Gelişigüzel şekilli üç boyutlu iletken yüzeyleri içeren elektromanyetik saçınım problemlerinin çözümü için hızlı çokkutup yöntemi (HÇY) incelenmiştir. Bu yöntemin, hesapsal karmaşıklığı ve bellek gereksinimini, çözümün doğruluğundan taviz vermeden düşürdüğü bilinmektedir. Böylece, büyük problemlerin çözümü geleneksel çözüm algoritmalarına göre bilgisayar kaynaklarının daha az kullanımıyla yapılabilmektedir. Öte yandan, HÇY'nin beklenen verimliliği, bileşenleri uygun seçilmediği sürece elde edilemeyebilir. Kullanılan integral denklemi, iteratif algoritma ve preconditioning tekniği uygulamanın verimliliğini doğrudan etkiler. Bunların kullanımının birbirine bağlı olmasından dolayı, verimli bir çözücünün tasarımı zorlu bir işe dönüşmektedir.

Anahtar kelimeler: Hızlı çokkutup yöntemi (HÇY), elektromanyetik saçınım, integral denklemleri, iteratif algoritmalar, preconditioning teknikleri.

ACKNOWLEDGMENTS

I gratefully thank my supervisor Dr. Levent Gürel for his supervision, guidance, and suggestions throughout the development of this thesis. I would also like to express my deepest gratitude to him for supporting my studies on the computational electromagnetics.

I also thank Prof. Dr. Ergin Atalar and Dr. Vakur B. Ertürk for reading and commenting on this thesis.

I also thank Prof. Dr. Ayhan Altıntaş for commenting on this thesis.

I also thank all undergraduate and graduate students of Bilkent Computational Electromagnetics Laboratory for their cooperation and accompaniment in my studies.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Historical Background	2
1.3	Contributions	5
1.4	Simulation Environment and Computational Resources	6
2	Integral Equations	7
2.1	Electric-Field Integral Equation	7
2.2	Magnetic-Field Integral Equation	8
2.2.1	Radiation Integral	9
2.2.2	Derivation by Using Interior Boundary Condition	12
2.3	Combined-Field Integral Equation	13
2.4	Method of Moments	14
2.4.1	Geometry Modelling and Meshing	16
2.4.2	Triangular Rooftop Basis Functions	19
3	MOM Implementations with the EFIE	22
3.1	Formulation	22
3.2	Singularity Extraction	27
3.3	Evaluation of Impedance-Matrix Elements	32
3.4	Numerical Evaluation of Integrals	33
3.4.1	Adaptive Integration: Method 1	34

3.4.2	Adaptive Integration: Method 2	36
3.5	Excitation	40
3.5.1	Plane-Wave Excitation	40
3.5.2	Delta-Gap Excitation	41
3.5.3	Current Sources	42
3.6	Computational Analysis of the MOM	43
3.7	RCS Calculations for RWG Basis Function	44
4	MOM Implementations with the MFIE and the CFIE	46
4.1	MFIE-MOM Implementation with RWG Testing Function	47
4.1.1	Formulation	48
4.1.2	Singularity Extraction	50
4.1.3	Evaluation of the Impedance-Matrix Elements	56
4.1.4	Singularity Extraction for Outer Integrals	63
4.1.5	Numerical Integration of Inner and Outer Integrals	66
4.1.6	Excitation	67
4.2	MFIE-MOM Implementation with Point Testing Functions	67
4.2.1	Formulation	68
4.2.2	Evaluation of the Impedance-Matrix Elements	70
4.3	CFIE-MOM Implementation with RWG Basis and Testing Functions	75
4.4	CFIE-MOM Implementation with Point Testing Function for the MFIE	76
4.5	Results	76
5	FMM Implementations	86
5.1	Theory of the FMM	87
5.1.1	Factorization of the Green's Function	87
5.1.2	Calculation of an Interaction	88
5.1.3	Clustering and Evaluation of the Interactions	91

5.1.4	Truncation Number	95
5.1.5	Integration Points	98
5.1.6	Computational Analysis of the FMM	99
5.2	FMM Implementation with RWG Basis Functions	102
5.3	Results	103
5.3.1	Accuracy of the FMM Implementations	104
5.3.2	Efficiency of the FMM Implementations	104
6	Iterative Methods and Preconditioning Techniques	109
6.1	Iterative Solution of the Linear Systems	110
6.1.1	Bi-Conjugate Gradient (BiCG) Algorithm	111
6.1.2	Conjugate Gradient Squared (CGS) Algorithm	115
6.2	Preconditioning Techniques	116
6.2.1	Types of Preconditioners	117
6.2.2	Comparisons of Preconditioners and Iterative Solvers	121
6.3	Comparisons of the Integral Equations	122
6.3.1	Efficiency of the FMM for Different Formulations	124
6.3.2	Choice of α in the CFIE	126
6.4	Effect of Initial Guess on the Convergence	126
6.5	Multiple Excitations	127
7	MLFMA Implementations	132
7.1	Evaluation of Interactions with the MLFMA	133
7.1.1	Aggregation in the MLFMA	133
7.1.2	Translation in the MLFMA	139
7.1.3	Disaggregation in the MLFMA	140
7.2	Clustering and Structure of the MLFMA	146
7.3	Computational Analysis of the MLFMA	148
7.3.1	Computational Analysis of Aggregation and Disaggregation	149

7.3.2	Computational Analysis of Translation	150
7.4	Results	152
7.4.1	Accuracy of the MLFMA Implementations	153
7.4.2	Efficiency of the MLFMA Implementations	153
8	Conclusion	155
	Bibliography	158
	Glossary	162

List of Figures

2.1	Observation point approaching a smooth portion of the surface from outside.	10
2.2	Surface models of various closed geometries with triangular meshing.	17
2.3	Surface models of various open geometries with triangular meshing.	18
2.4	RWG function defined on the triangular domains.	19
3.1	RWG function does not have current flow across its boundaries. .	24
3.2	Location of the basis and testing triangles after the coordinate transformation.	26
3.3	Source triangle on the x - y plane and projection of the observation point.	28
3.4	Geometric variables introduced to express the results of analytical integrals.	29
3.5	Different cases for the projection of the observation point.	31
3.6	Adaptive integration method 1.	34
3.7	Function to be integrated on the triangular domain.	36
3.8	Integration points taken by the first adaptive method to evaluate the integral of the function shown in Figure 3.7	37
3.9	Adaptive integration method 2.	38
3.10	Integration points taken by the second adaptive method to evaluate the integral of the function shown in Figure 3.7	39
3.11	Delta-gap source defined on the m^{th} edge.	41

3.12	Current sources defined with the RWG functions.	42
3.13	Corresponding physical equivalences for the current sources shown in Figure 3.12.	43
4.1	Limit cases for observation points approaching the basis triangle. .	52
4.2	Observation points approaching the edge at different angles. . . .	54
4.3	Magnitude of the integral $\mathbf{I} = \int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z f_{m3}(R)$ for the scenario in Figure 4.2.	54
4.4	Interaction between the neighboring basis and testing triangles. .	61
4.5	Limit case as the observation point approaches the edge of the basis function.	62
4.6	Second coordinate transformation used to annihilate the singular- ity of the outer integral.	64
4.7	Point testing functions.	68
4.8	Point testing function approaching the center of the edge of the basis function from outside.	70
4.9	Point testing function approaching the edge of the basis triangle. .	71
4.10	The testing function residing on one of the edges of the basis tri- angle, on which the basis function is not defined.	73
4.11	Models of a sphere having a radius of 0.3 m with mesh sizes (a) 6 cm, (b) 5 cm, (c) 4 cm, and (d) 3 cm.	79
4.12	RCS values obtained by the EFIE-MOM implementation for a sphere with a radius of (a) 0.5λ , (b) 0.6λ , (c) 0.75λ , and (d) 1.0λ . 81	
4.13	RCS values obtained by the MFIE-MOM implementation using point testing functions for a sphere with a radius of (a) 0.5λ , (b) 0.6λ , (c) 0.75λ , and (d) 1.0λ	82
4.14	RCS values obtained by the MFIE-MOM implementation using the RWG functions and Galerkin method for a sphere with a radius of (a) 0.5λ , (b) 0.6λ , (c) 0.75λ , and (d) 1.0λ	83

4.15	RCS values obtained by the CFIE-MOM implementation using the RWG functions and Galerkin method for a sphere with a radius of (a) 0.5λ , (b) 0.6λ , (c) 0.75λ , and (d) 1.0λ	84
4.16	RCS values obtained by the EFIE, the MFIE, and the CFIE formulations for a sphere with a radius of 0.8λ . The MFIE suffers from the internal resonance problem.	85
5.1	Dividing $\mathbf{r} - \mathbf{r}'$ into segments.	89
5.2	Clustering of basis and testing functions provides a single translation between the clusters.	91
5.3	Aggregation process in basis clusters.	92
5.4	Translation from a basis cluster to testing clusters.	93
5.5	Disaggregation process in testing clusters.	94
5.6	Magnitude of the imaginary part of the Hankel function, $h_l^{(1)}(x)$ with respect to l for different values of x	96
5.7	Values of L obtained with the formula given in Equation (5.24) and direct test of Equation (5.8) for an accuracy of three digits.	97
5.8	Angular integration points for $L = 5$	99
5.9	RCS values obtained by the EFIE-FMM implementation using the RWG functions and the Galerkin method for a sphere having a radius of (a) 0.75λ , (b) 1.0λ , (c) 1.5λ , and (d) 2.0λ	107
5.10	RCS values obtained by the CFIE-FMM implementation using the RWG functions and the Galerkin method for a sphere having a radius of (a) 0.75λ , (b) 1.0λ , (c) 1.5λ , and (d) 2.0λ	108
6.1	General form of an iterative solution.	110
6.2	Pseudocode of a general BiCG algorithm for complex systems.	113
6.3	Pseudocode of the reduced form of the BiCG algorithm for complex symmetric systems.	115

6.4	Pseudocode of the general CGS algorithm for complex systems.	116
6.5	Self-cluster interactions in the impedance matrix of an 72-unknown problem.	119
6.6	Self-cluster interactions in the impedance matrix of an 72-unknown problem after the renumbering operation.	120
6.7	Plots of the residual versus the iteration number for different types of preconditioners implemented with CGS and BiCG for the solution of (a) 1302-unknown, and (b) 8364-unknown scattering problems of the sphere shown in Figure 4.11. The EFIE is used to formulate the problems.	128
6.8	Plots of the residual versus the iteration number for the solution of sphere problems shown in Figure 4.11 using different integral-equation formulations with CGS employing (a) diagonal and (b) block-diagonal preconditioners.	129
6.9	Number of iterations with respect to α to reach various residual errors for the sphere problems having (a) 1302 unknowns, (b) 2076 unknowns, (c) 3723 unknowns, and (d) 8364 unknowns.	130
6.10	Effect of the improved initial guess on the convergence for different types of preconditioners implemented with BiCG for the solution of (a) 1302-unknown and (b) 8364-unknown scattering problems of the sphere shown in Figure 4.11.	131
7.1	Aggregation process in the MLFMA.	134
7.2	Fine and coarse grids on the unit sphere.	135
7.3	2-D Interpolation.	136
7.4	Interpolation points after the first step.	136
7.5	Extended sample space of unit sphere.	138
7.6	Translation process in the MLFMA.	140
7.7	Disaggregation process in the MLFMA.	141

7.8	Operations on function samplings: (a) interpolation, (b) antepola- tion from coarse to fine grid, (c) decimation, and (d) antepolation from fine to coarse grid.	144
7.9	Clustering with cubes in the MLFMA: (a) the main cube, (b) subcubes of the main cube.	146
7.10	Near and far clusters in the clustering method with cubes. In the worst case, the distance between two far-field clusters is $2a$	147
7.11	Tree structure of the MLFMA.	148
7.12	RCS values obtained by the CFIE-MLFMA implementation for a sphere having a radius of (a) 1.5λ , (b) 2.0λ , (c) 3.0λ , and (d) 6.0λ	154

List of Tables

4.1	Sphere problems.	77
5.1	Comparison of the processing times of the EFIE-MOM and the EFIE-FMM implementations.	105
5.2	Comparison of the memory requirements of the EFIE-MOM and the EFIE-FMM implementations.	105
6.1	Processing time of the EFIE-FMM implementation.	124
6.2	Memory requirement of the EFIE-FMM implementation.	124
6.3	Processing time of the CFIE-FMM implementation.	126
6.4	Memory requirement of the CFIE-FMM implementation.	126
7.1	Comparison of the processing times of the CFIE-FMM and the CFIE-MLFMA implementations.	153
7.2	Comparison of the memory requirements of the CFIE-FMM and the CFIE-MLFMA implementations.	153

To My Ayça and My Family

Chapter 1

Introduction

Maxwell's equations

$$\nabla \times \mathbf{H}(\mathbf{r}) = \frac{\partial \mathbf{D}(\mathbf{r})}{\partial t} + \mathbf{J}(\mathbf{r}) \quad (1.1)$$

$$\nabla \times \mathbf{E}(\mathbf{r}) = -\frac{\partial \mathbf{B}(\mathbf{r})}{\partial t} \quad (1.2)$$

$$\nabla \cdot \mathbf{D}(\mathbf{r}) = \rho(\mathbf{r}) \quad (1.3)$$

$$\nabla \cdot \mathbf{B}(\mathbf{r}) = 0 \quad (1.4)$$

and the associated boundary conditions are the basic tools for the formulation of the electromagnetic problems. However, analytic treatment of these equations is limited to some canonical geometries. For arbitrary geometries, numerical methods have been developed in recent years. With the aid of the advances in the computer technology, it has become possible to apply these numerical techniques to very large electromagnetic problems.

1.1 Motivation

Most of the real-life problems in the electromagnetics are large, requiring immense computational resources in their solutions. In addition, the solutions of these problems are usually required to be performed for many times. The advances

in the computer technology make it possible to solve larger problems in reduced processing times. On the other hand, fast and efficient algorithms are developed to perform these solutions with restricted computational resources. The necessity of these algorithms is not negated by the advances in the computer technology because of the following reasons:

1. They can be integrated with the latest advances in the computer hardware technology so that it becomes possible to solve even larger problems.
2. They are designed to perform the solutions with less number of operations and they minimize the round-off errors introduced by the computers.
3. The improvement introduced by the fast algorithms usually provides greater advances than the improvement introduced by the computer hardware technology in the solution of large problems.

As a results, fast and efficient algorithms should be used to increase the ability of computer technology in the solution of large electromagnetic problems.

1.2 Historical Background

Numerical techniques in computational electromagnetics are generally divided into two groups: partial-differential-equation (PDE) and integral-equation (IE) techniques. The finite-difference time-domain technique (FDTD) and the finite element method (FEM) are the most popular PDE techniques, while the method of moments (MOM) [1], the fast multiple method (FMM) [2, 3] and its extension, the multi-level fast multipole algorithm (MLFMA) [4]–[6], recursive T-matrix algorithms (RTMA's) [7, 8], and the fast far-field approximation (FAFFA) [9] are the most popular IE techniques.

In this thesis, the MOM and the FMM will be investigated in detail for the

solution of electromagnetic scattering problems. These methods rely on the application of the discrete versions of the integral equations on the surfaces of the geometries. Three integral equations will be used: The electric-field integral equation (EFIE), the magnetic-field integral equation (MFIE), and the combined-field integral equation (CFIE). Following the discretization, a linear system of equations is obtained. The unknown of the system is a coefficient vector for the given expansion of the induced current on the surface.

For problems that include large geometries in terms of the wavelength, the size of the linear system becomes large, too. On the other hand, the MOM is not suitable for solving large problems. For a linear system with N unknowns, this method has $O(N^2)$ complexity for constructing the system, while the memory requirement is also $O(N^2)$. In addition, the direct solution of the system requires $O(N^3)$ processing time, and $O(IN^2)$ for an iterative solution with I iterations. Consequently, as the number of unknowns increases, this method becomes very inefficient in terms of the memory usage and the processing time. It becomes impossible to increase the problem size with the existing computational resources. However, the MOM forms the basis for the FMM, which is much more efficient.

The FMM is based on the iterative solution of the linear system. This method provides an efficient calculation of the matrix-vector multiplications required by the iterative solver. With this method, the memory requirement drops to $O(N^{3/2})$ and the processing time becomes $O(IN^{3/2})$. Consequently, it becomes possible to solve larger problems. However, some FMM implementations may not show the expected efficiency. It will be shown that the type of the integral equation to formulate the problem directly affects the behavior of the iterative solver. Then, the iterative solver may require large preconditioners to reach the solution and this leads to increased memory usage and processing time.

Linear systems obtained by the EFIE are usually ill-conditioned, especially for the problems involving closed surfaces. Therefore, their usage in the FMM reduces the efficiency significantly. Using the MFIE or the CFIE leads to better results and the efficiency can be increased to expected levels. It will be shown that the CFIE is the most suitable integral equation to formulate large problems. On the other hand, these two integral equations cannot be applied to geometries with open surfaces. Therefore, the EFIE becomes the only choice for those problems.

The extension of the FMM is the multi-level fast multipole algorithm (MLFMA), which further reduces the processing time to $O(IN \log N)$ and memory requirement to $O(N \log N)$. The MOM, FMM and MLFMA implementations will be applied on scattering-from-sphere problems to demonstrate the improvement on the efficiency. It will be seen that, with the existing computational resources, the MOM is limited to about 14,000-unknown problems, while it becomes possible to solve a 130,000-unknown problem with the MLFMA.

The integrals arising in the MOM are evaluated by numerical techniques. This causes a controllable error in the solutions. In the MLFMA, there are three additional error sources that may also affect the accuracy of the final result. These errors can also be controlled; however, the trade-off between the efficiency and the accuracy should be balanced. The accuracy of the implementations will be presented by comparisons to the analytical results.

1.3 Contributions

The main objective of this thesis is to solve large electromagnetic scattering problems with high accuracy. A brief overview on the integral equations will be given in Chapter 2. In Chapters 3 and 4, MOM implementations will be shown in detail to form the basis for the FMM implementations. Chapter 5 introduces the FMM implementations using three different integral equations. The efficiency considerations, with a focus on the iterative solutions of the problems, are investigated in Chapter 6. Finally, the MLFMA will be presented in Chapter 7 as the final stage of this work. Progress towards the MLFMA will be accompanied by the results on accuracy and efficiency.

An efficient implementation of the MOM using the MFIE formulation and the Rao-Wilton-Glisson (RWG) functions is given in [10]. An alternative implementation is presented in Chapter 4, which divides the expression for the impedance-matrix elements into basic integrals and evaluates these basic integrals in a parallel manner. Adaptive integration algorithms are given in Chapter 3 to perform the numerical integrations efficiently, with an adjustable accuracy.

The evaluation of the analytical integrals appearing in the MFIE formulation is given in [10]-[12]; however, an investigation on the limit values will be presented in Chapter 4. It will be shown that the solid angle expression in the MFIE can be derived in these evaluations. In addition, for the unbounded integrand of the testing integrals in the neighboring interactions, a singularity extraction method will be suggested.

Although the CFIE is known to give better-conditioned systems for the closed geometries [4], Chapter 6 presents a detailed investigation on the efficiency of

the FMM implementations, according to the choices of the integral equation, iterative solver, and preconditioning technique. It will be shown that the EFIE may not be used with efficient preconditioners, such as the diagonal and block-diagonal preconditioners. It will be also shown that the increase in the number of iterations is very limited for the CFIE as the number of unknown increases.

1.4 Simulation Environment and Computational Resources

To model the geometries, a commercial program, *I-DEAS*, has been used. This program is also able to mesh the geometries, which is required by the numerical implementations. Various geometries meshed with linear triangles will be shown at the end of Chapter 2.

It should be noted that the modelling stage includes an error due to the inexact modelling of the curved surfaces with the linear elements. In addition, the integral equations are used in discrete form, which leads to another error source. These errors cannot be avoided; however, they can be made ignorably small by choosing the size of the elements carefully, before the numerical treatment.

The solvers are implemented by Fortran programs, which are able to use the exported data of the *I-DEAS*. The solutions are performed on Digital Alpha processors with 2 GB memory. In addition to the solvers, other Fortran programs have also been implemented to display the current distribution by the *I-DEAS* and to calculate the radar cross section (RCS).

Chapter 2

Integral Equations

This chapter introduces the surface integral equations to solve the scattering and radiation problems of objects having arbitrarily shaped geometries. For a conducting surface, application of the boundary conditions leads to reduction of the three-dimensional problem into a two-dimensional one involving the surface. In this thesis, the objects are assumed to be perfectly conducting; however, integral equations are also applicable to penetrable scatterers.

For the numerical solution of the problem involving the continuous fields and current density, the method of moments (MOM) will be introduced. This method expands the current density in terms of known basis functions, and tests the integral equation as many times as the number of unknown coefficients. The result of this method is a linear system to be solved by a suitable technique.

2.1 Electric-Field Integral Equation

The electric-field integral equation (EFIE) can be derived by applying the boundary condition about the tangential component of the electric field on a surface.

For a perfectly conducting surface,

$$\hat{\mathbf{t}} \cdot \mathbf{E}^{inc}(\mathbf{r}) + \hat{\mathbf{t}} \cdot \mathbf{E}^{sca}(\mathbf{r}) = 0, \quad (2.1)$$

where the observation point, \mathbf{r} , is located on the surface. In the equation, \mathbf{E}^{inc} represents the incident field due to external sources and \mathbf{E}^{sca} represents the scattered field, which can be written in terms of the induced current as

$$\mathbf{E}^{sca}(\mathbf{r}) = i\omega\mu \int_{S'} d\mathbf{r}' \overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}'), \quad (2.2)$$

where

$$\overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}') = \left[\overline{\mathbf{I}} + \frac{\nabla\nabla}{k^2} \right] g(\mathbf{r}, \mathbf{r}') \quad (2.3)$$

is the dyadic Green's function and

$$g(\mathbf{r}, \mathbf{r}') = \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} \quad (2.4)$$

is the Green's function for the three-dimensional scalar Helmholtz equation. Then, the EFIE can be formed by substituting the scattered field expression into the boundary condition as

$$\hat{\mathbf{t}} \cdot \int_{S'} d\mathbf{r}' \overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}') = \frac{i}{k\eta} \hat{\mathbf{t}} \cdot \mathbf{E}^{inc}(\mathbf{r}). \quad (2.5)$$

2.2 Magnetic-Field Integral Equation

The magnetic-field integral equation (MFIE) can be derived by applying the boundary condition about the tangential component of the magnetic field on closed surfaces of objects. For a conducting surface, this boundary condition can be written as

$$\hat{\mathbf{n}} \times \mathbf{H}^{inc}(\mathbf{r}) + \hat{\mathbf{n}} \times \mathbf{H}^{sca}(\mathbf{r}) = \mathbf{J}(\mathbf{r}) \quad (2.6)$$

if the observation point, \mathbf{r} , approaches the surface from outside. In this equation, \mathbf{H}^{inc} represents the incident magnetic field due to external sources, while \mathbf{H}^{sca}

represents the magnetic field scattered due to the induced current on the surface and can be written as

$$\mathbf{H}^{sca}(\mathbf{r}) = \int_{S'} d\mathbf{r}' \mathbf{J}(\mathbf{r}') \times \nabla' g(\mathbf{r}, \mathbf{r}'), \quad (2.7)$$

where $g(\mathbf{r}, \mathbf{r}')$ is the Green's function defined in (2.4). Then, it is convenient to rewrite Equation (2.6) as

$$\mathbf{J}(\mathbf{r}) - \hat{\mathbf{n}} \times \int_{S'} d\mathbf{r}' \mathbf{J}(\mathbf{r}') \times \nabla' g(\mathbf{r}, \mathbf{r}') = \hat{\mathbf{n}} \times \mathbf{H}^{inc}(\mathbf{r}), \quad (2.8)$$

which is the general expression for the MFIE for conducting closed surfaces when the observation point approaches from the outside.

2.2.1 Radiation Integral

As the observation point approaches the surface, the radiation integral can be divided into two integrals as [13]

$$\begin{aligned} \int_{S'} d\mathbf{r}' \mathbf{J}(\mathbf{r}') \times \nabla' g(\mathbf{r}, \mathbf{r}') &= \lim_{S'_\varepsilon \rightarrow 0} \int_{S' - S'_\varepsilon} d\mathbf{r}' \mathbf{J}(\mathbf{r}') \times \nabla' g(\mathbf{r}, \mathbf{r}') \\ &+ \lim_{S'_\varepsilon \rightarrow 0} \int_{S'_\varepsilon} d\mathbf{r}' \mathbf{J}(\mathbf{r}') \times \nabla' g(\mathbf{r}, \mathbf{r}'), \end{aligned} \quad (2.9)$$

which can be written shortly as

$$\mathbf{H}^{sca}(\mathbf{r}) = \mathbf{H}_{PV}^{sca}(\mathbf{r}) + \mathbf{H}_{\varepsilon \rightarrow 0}^{sca}(\mathbf{r}). \quad (2.10)$$

S'_ε in Equation (2.9) is the infinitesimal surface in the vicinity of the observation point, and the rest of the surface is represented by the first integral. The limits in the equation indicate that S'_ε shrinks to zero. However, its contribution to the overall integral does not evaluate to zero since the integrand tends to diverge when the observation point, \mathbf{r} , approaches the source point, \mathbf{r}' . Figure 2.1 shows that the observation point at $(0, 0, z)$ approaches a smooth portion of the surface as z goes to zero. In the figure, V_i represents the interior of the closed object, so that the observation point approaches the surface from outside. The infinitesimal

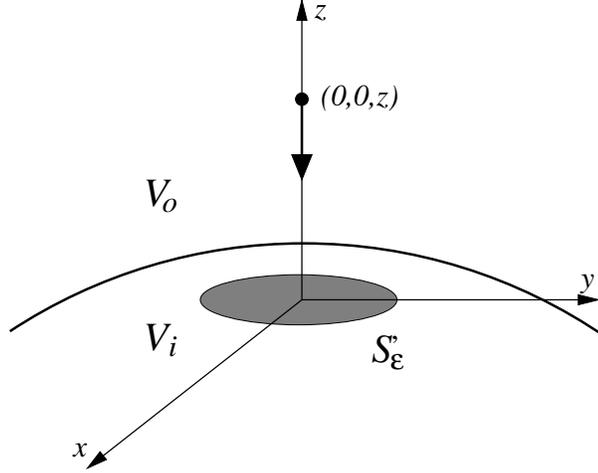


Figure 2.1: Observation point approaching a smooth portion of the surface from outside.

surface S'_ϵ is locally planar having its normal in the z direction and in the shape of a circle with radius ϵ .

Since the observation point becomes close to the source, Green's function given in Equation (2.4) can be approximated as

$$g(\mathbf{r}, \mathbf{r}') = \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} \approx \frac{1}{4\pi|\mathbf{r}-\mathbf{r}'|}. \quad (2.11)$$

In addition, source area is small enough that the current density can be assumed to be constant over the entire surface with a value of $\mathbf{J}(\mathbf{r}_0)$, where $\mathbf{r}_0 = (0, 0, 0)$ is the point on the surface, which the observation point approaches in the limit. Then, the current density can be taken out of the integral and the expression for the scattered magnetic field from the infinitesimal surface becomes

$$\mathbf{H}_\epsilon^{sca}(0, 0, z) \approx \mathbf{J}(\mathbf{r}_0) \times \int_{S_\epsilon} d\mathbf{r}' \nabla' \left(\frac{1}{4\pi|\mathbf{r}-\mathbf{r}'|} \right). \quad (2.12)$$

If the gradient expression is written explicitly as

$$\nabla' \left(\frac{1}{|\mathbf{r}-\mathbf{r}'|} \right) = \frac{\mathbf{R}}{R^3}, \quad (2.13)$$

where

$$\mathbf{R} = \mathbf{r} - \mathbf{r}' = \hat{\mathbf{z}}z - \boldsymbol{\rho}', \quad (2.14)$$

then

$$\begin{aligned}
\mathbf{H}_\varepsilon^{sca}(0, 0, z) &\approx \frac{\mathbf{J}(\mathbf{r}_0)}{4\pi} \times \int_{S_\varepsilon} d\mathbf{r}' \frac{\mathbf{R}}{R^3} \\
&= \frac{\mathbf{J}(\mathbf{r}_0)}{4\pi} \times \hat{\mathbf{z}} \int_0^{2\pi} d\phi' \int_0^\varepsilon d\rho' \frac{\rho' z}{[(\rho')^2 + z^2]^{3/2}} \\
&\quad - \frac{\mathbf{J}(\mathbf{r}_0)}{4\pi} \times \hat{x} \int_0^{2\pi} d\phi' \int_0^\varepsilon d\rho' \frac{(\rho')^2 \cos \phi'}{[(\rho')^2 + z^2]^{3/2}} \\
&\quad - \frac{\mathbf{J}(\mathbf{r}_0)}{4\pi} \times \hat{y} \int_0^{2\pi} d\phi' \int_0^\varepsilon d\rho' \frac{(\rho')^2 \sin \phi'}{[(\rho')^2 + z^2]^{3/2}} \\
&= -\frac{\mathbf{J}(\mathbf{r}_0)}{2} \times \hat{\mathbf{z}} \left. \frac{z}{[(\rho')^2 + z^2]^{1/2}} \right|_0^\varepsilon \\
&= -\frac{\mathbf{J}(\mathbf{r}_0)}{2} \times \hat{\mathbf{z}} \left[\frac{z}{[\varepsilon^2 + z^2]^{1/2}} - \frac{z}{|z|} \right]. \tag{2.15}
\end{aligned}$$

As the observation point approaches the surface, z goes to zero and

$$\mathbf{H}_\varepsilon^{sca}(0, 0, z = 0) \approx \frac{\mathbf{J}(\mathbf{r}_0)}{2} \times \hat{\mathbf{z}}. \tag{2.16}$$

This result can be generalized as

$$\mathbf{H}_{\varepsilon \rightarrow 0}^{sca}(\mathbf{r}_0) = \frac{\mathbf{J}(\mathbf{r}_0)}{2} \times \hat{\mathbf{n}}, \tag{2.17}$$

where $\hat{\mathbf{n}}$ represents the normal of the surface in general.

When the observation point approaches a smooth portion of the surface, Equation (2.17) indicates that the limit value of the magnetic field shown in Equation (2.10) has a magnitude equal to half of the current density and has a direction perpendicular to and in the same plane with the current flow. Consequently, if some current flow exists on a planar surface, the principle value of the magnetic field radiated by this current density is perpendicular to that plane, while the limit value forms the parallel component.

In general, the limit value depends on the solid angle of the surface at the observation point, which is simply 2π for smooth surfaces. This dependence will

be shown explicitly in Chapter 4, but Equation (2.8) should be rewritten here as [14]

$$\frac{\Omega_o}{4\pi} \mathbf{J}(\mathbf{r}) - \hat{\mathbf{n}} \times \int_{S'(PV)} d\mathbf{r}' \mathbf{J}(\mathbf{r}') \times \nabla' g(\mathbf{r}, \mathbf{r}') = \hat{\mathbf{n}} \times \mathbf{H}^{inc}(\mathbf{r}) \quad (2.18)$$

with

$$\mathbf{H}_{\varepsilon \rightarrow 0}^{sca}(\mathbf{r}_0) = \frac{\Omega_i}{4\pi} \mathbf{J}(\mathbf{r}_0) \times \hat{\mathbf{n}}, \quad (2.19)$$

where Ω_o represent the external and Ω_i represents the internal solid angles of the surface at the observation point, which may take values between 0 and 4π . It should be indicated that the Equation (2.18) relies on the assumption that the current density is continuous at the observation point.

2.2.2 Derivation by Using Interior Boundary Condition

If the observation point approaches the boundary from the interior of the closed surface, the boundary condition about the tangential component of the magnetic field should be written as

$$\hat{\mathbf{n}} \times \mathbf{H}^{inc}(\mathbf{r}) + \hat{\mathbf{n}} \times \mathbf{H}^{sca}(\mathbf{r}) = 0 \quad (2.20)$$

if the surface is perfectly conducting. Using the expression for scattered magnetic field given in Equation (2.7),

$$\hat{\mathbf{n}} \times \mathbf{H}^{inc}(\mathbf{r}) + \hat{\mathbf{n}} \times \int_{S'} d\mathbf{r}' \mathbf{J}(\mathbf{r}') \times \nabla' g(\mathbf{r}, \mathbf{r}') = 0, \quad (2.21)$$

which seems to be different from (2.8). However, if the same procedure was applied to the radiation integral, then the limit value would be evaluated for smooth surfaces as

$$\mathbf{H}_{\varepsilon \rightarrow 0}^{sca}(\mathbf{r}_0) = -\frac{\mathbf{J}(\mathbf{r}_0)}{2} \times \hat{\mathbf{n}}, \quad (2.22)$$

while it becomes

$$\mathbf{H}_{\varepsilon \rightarrow 0}^{sca}(\mathbf{r}_0) = -\frac{\Omega_o}{4\pi} \mathbf{J}(\mathbf{r}_0) \times \hat{\mathbf{n}} \quad (2.23)$$

in general. Using these limit expressions, Equation (2.21) changes into Equation (2.18) again, although the derivation was based on a different form of the boundary condition.

2.3 Combined-Field Integral Equation

When the EFIE or the MFIE are used for the scattering problems of closed objects, internal resonance problems arise at resonance frequencies. Both integral equations have nonzero null-space solutions, which adds extra current on the object in addition to the correct current distribution. The extra current from the solution of the EFIE does not radiate so that the far-field radiation can be calculated correctly, even when the current is incorrect. However, null-space solution of the MFIE radiates and leads to incorrect calculation of far-field radiation [15].

To avoid the internal resonance problem, the combined-field integral equation (CFIE) can be used. This integral equation is simply the linear combination of the EFIE and the MFIE, and can be represented as

$$\text{CFIE} = \alpha\text{EFIE} + (1 - \alpha)\text{MFIE}, \quad (2.24)$$

where α may take values between 0 and 1. Then, the CFIE can be written by combining the Equations (2.5) and (2.8) as

$$\begin{aligned} & \alpha \left[\hat{\mathbf{t}} \cdot \int_{S'} d\mathbf{r}' \overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}') \right] \\ & + \frac{i}{k} (1 - \alpha) \left[\mathbf{J}(\mathbf{r}) - \hat{\mathbf{n}} \times \int_{S'} d\mathbf{r}' \mathbf{J}(\mathbf{r}') \times \nabla' g(\mathbf{r}, \mathbf{r}') \right] \\ & = \frac{i}{k} \left[\frac{\alpha}{\eta} \hat{\mathbf{t}} \cdot \mathbf{E}^{inc}(\mathbf{r}) + (1 - \alpha) \hat{\mathbf{n}} \times \mathbf{H}^{inc}(\mathbf{r}) \right]. \end{aligned} \quad (2.25)$$

It should be noted that the MFIE part is multiplied by the factor of i/k , in order to weight the equations equally before the linear combination.

Using the CFIE, two boundary conditions are applied at the same time and null-space solution of this equation is zero for all frequencies. In other words, the CFIE is free of internal resonance problem and it has unique solution for all frequencies. However, the CFIE has another advantage that makes it more preferable than the EFIE or the MFIE. As it will be shown later, matrix equations obtained by using the CFIE are generally better conditioned than the systems obtained by using the EFIE or the MFIE. This leads to relatively faster convergence of the iterative methods in the solution of the system, and gives superiority to the CFIE over the others [16]. Nevertheless, the MFIE and the CFIE cannot be used for the scattering problems of objects with open surfaces, and the EFIE becomes the only choice for those problems.

2.4 Method of Moments

Integral equations introduced in the previous sections can be represented in general as

$$L\{\mathbf{f}(x)\} = \mathbf{g}(x), \quad (2.26)$$

where L is the linear operator of the equation, while $\mathbf{f}(x)$ is the unknown function that stands for the current distribution and $\mathbf{g}(x)$ is the excitation. To solve the problem, the unknown function can be expanded in a series of basis functions as

$$\mathbf{f}(x) \approx \sum_{n=1}^N a_n \mathbf{b}_n(x), \quad (2.27)$$

where a_n is the coefficient of the n^{th} basis function. The basis functions should be linearly independent and they have to be chosen appropriately, so that they can expand the current distribution. Then, defining the residual error as

$$\begin{aligned} R(x) &= L\left\{\sum_{n=1}^N a_n \mathbf{b}_n(x)\right\} - \mathbf{g}(x) \\ &= \left[\sum_{n=1}^N a_n L\{\mathbf{b}_n(x)\}\right] - \mathbf{g}(x), \end{aligned} \quad (2.28)$$

the aim becomes to make this error arbitrarily small, in order to solve the problem not exactly, but with a small error. For this purpose, another set of functions called the testing functions, $\mathbf{t}_m(x)$, can be used to weight both sides of (2.28). Defining an inner product as

$$\langle \mathbf{a}(x), \mathbf{b}(x) \rangle = \int dx \mathbf{a}(x) \cdot \mathbf{b}(x), \quad (2.29)$$

Equation (2.28) can be tested for $m = 1, \dots, N$ as

$$\int dx \mathbf{t}_m(x) \cdot \sum_{n=1}^N a_n L\{\mathbf{b}_n(x)\} = \int dx \mathbf{t}_m(x) \cdot \mathbf{g}(x). \quad (2.30)$$

Finally, interchanging the order of summation and integration, the set of equations becomes

$$\sum_{n=1}^N a_n \int dx \mathbf{t}_m(x) \cdot L\{\mathbf{b}_n(x)\} = \int dx \mathbf{t}_m(x) \cdot \mathbf{g}(x) \quad (2.31)$$

and a linear system can be formed as

$$\sum_{n=1}^N a_n Z_{mn} = v_m, \quad (2.32)$$

where the matrix elements are

$$Z_{mn} = \int dx \mathbf{t}_m(x) \cdot L\{\mathbf{b}_n(x)\} \quad (2.33)$$

and the vector elements are

$$v_m = \int dx \mathbf{t}_m(x) \cdot \mathbf{g}(x). \quad (2.34)$$

It is more evident in Equation (2.32) that N linearly independent equations are used to find the unknown coefficients of N basis function. In this equation, the matrix $\overline{\mathbf{Z}}$ is usually called the impedance matrix, and the vector \mathbf{v} is called the excitation vector. An element of the $\overline{\mathbf{Z}}$ matrix at (m, n) is referred to as the interaction between the m^{th} testing and n^{th} basis functions.

The method described above is called as the method of moments (MOM) [1], which has been extensively used by many scientist in the solution of various types of scattering problems. By this method, an operator equation is reduced to a matrix equation. Therefore, it becomes possible to solve the linear equations involving continuous functions numerically with small errors.

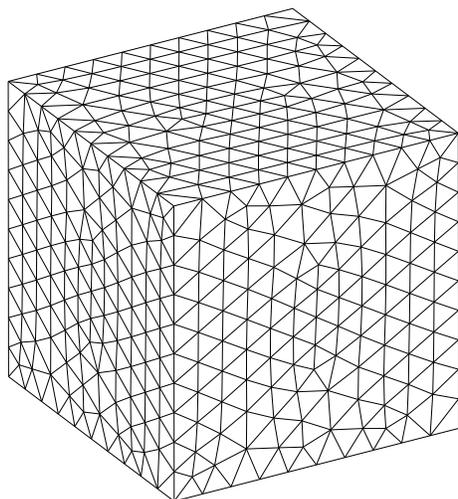
2.4.1 Geometry Modelling and Meshing

The application of the MOM requires the modelling of the problem geometries in the computer environment. Then, the surface models have to be meshed according to the type of the basis function to be used. Figures 2.2 and 2.3 show triangular meshing applied on various models of closed and open geometries. For high accuracy, the size of the triangles have to be small, which leads to large number of triangles or number of unknown coefficients, N . However, it becomes difficult to solve the linear system in (2.32) when N gets larger. The rule of thumb is to choose the average size of the mesh about 1/10 of the wavelength so that the MOM can be applied efficiently and with small error.

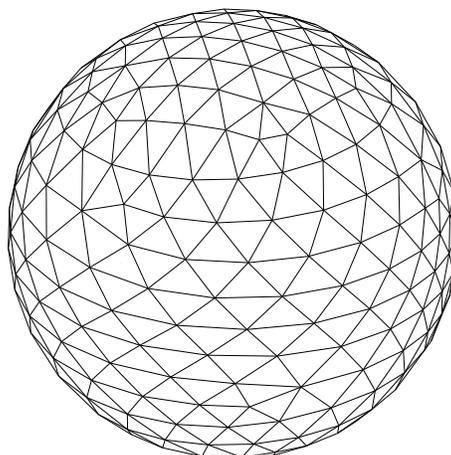
There are actually two error sources appearing at this stage. The first one is the approximation of the continuous current density with discrete functions. The second one is the modelling of the geometry with linear elements. The first error always exists, but it can be reduced by decreasing the element size. The second error, which also depends on the element size, exists only for the objects having curved surfaces, such as the sphere. In Figure 2.2(b), the smooth surface of the sphere is approximated by the combination of linear triangles.

Various types of basis functions can be defined on the triangular domains. For small error in the expansion of the current, high-order functions should be preferred. In addition, the type of the linear operator also affects the choice [17], as

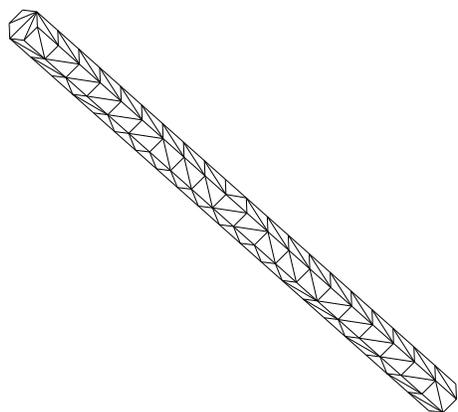
will be shown in the next subsection.



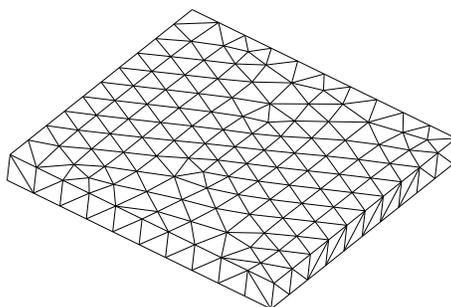
(a) Cube



(b) Sphere



(c) Rod



(d) Box

Figure 2.2: Surface models of various closed geometries with triangular meshing.

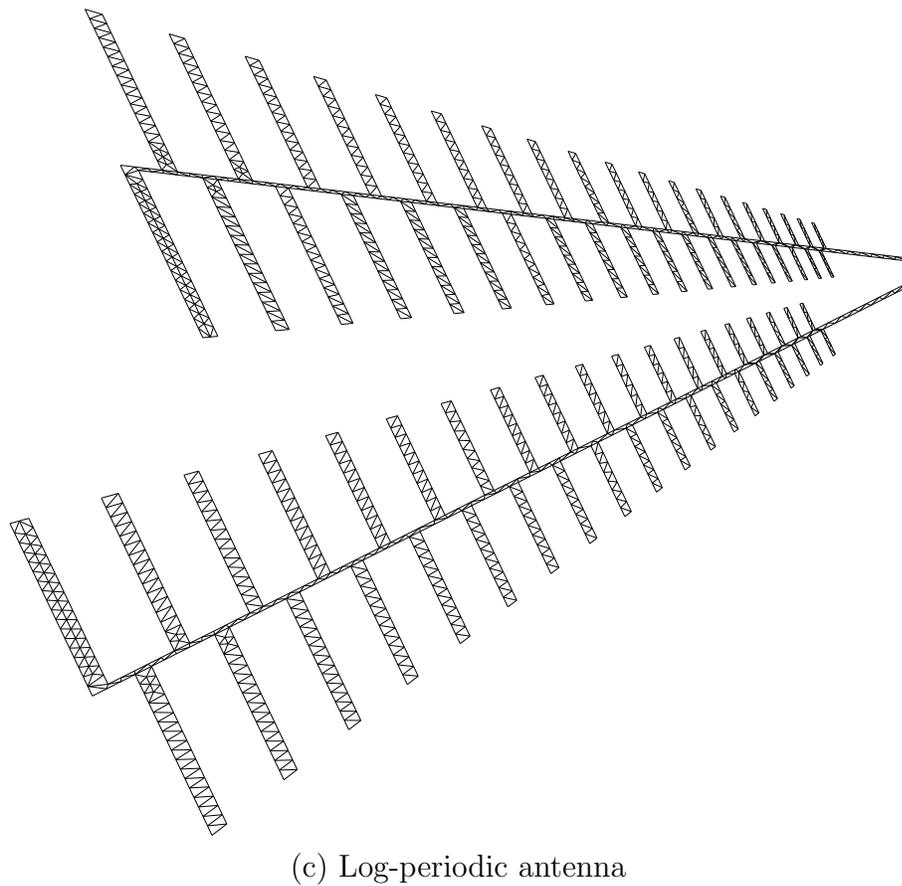
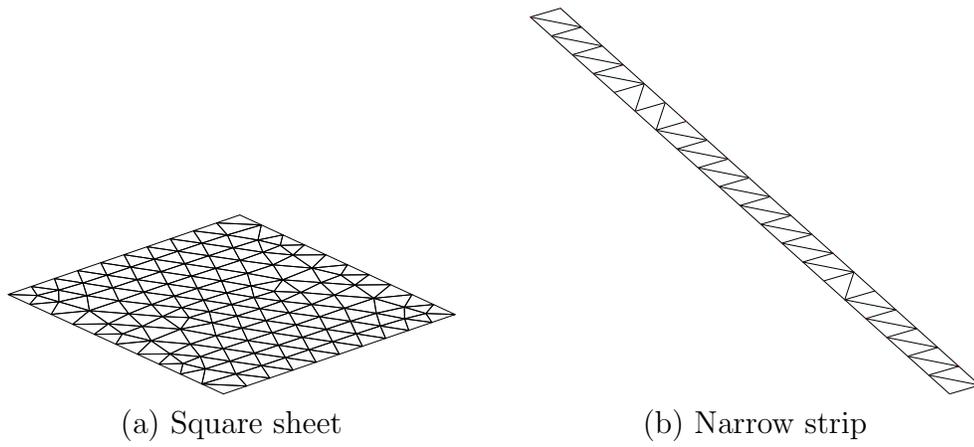


Figure 2.3: Surface models of various open geometries with triangular meshing.

2.4.2 Triangular Rooftop Basis Functions

Rao-Wilton-Glisson (RWG) functions are linearly varying vector functions defined on planar triangular domains [18]. Due to their useful properties, they have been widely used as basis and testing functions in MOM applications.

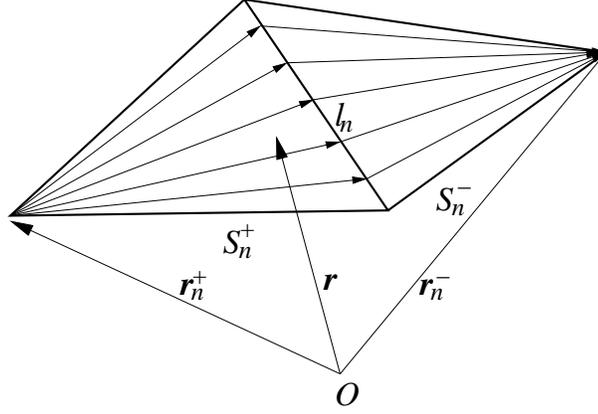


Figure 2.4: RWG function defined on the triangular domains.

Figure 2.4 shows an RWG function defined on two triangles having a common edge of length l_n . This RWG function associated with the n^{th} edge is given as

$$\mathbf{b}_n(\mathbf{r}) = \left\{ \begin{array}{ll} \frac{l_n}{2A_n^+}(\mathbf{r} - \mathbf{r}_n^+), & \mathbf{r} \in S_n^+ \\ \frac{l_n}{2A_n^-}(\mathbf{r}_n^- - \mathbf{r}), & \mathbf{r} \in S_n^- \\ 0, & \text{otherwise} \end{array} \right\}, \quad (2.35)$$

where A_n^+ and A_n^- are the areas of the triangle surfaces S_n^+ and S_n^- , respectively.

Since the function is zero outside the two triangles and due to the symmetry of the expressions in the two triangular domains, an alternative definition can be attempted for the function on the i^{th} triangle as

$$\mathbf{b}_{ik}(\mathbf{r}) = \pm \frac{l_{ik}}{2A_i}(\mathbf{r} - \mathbf{r}_{ik})\delta_i(\mathbf{r}), \quad (2.36)$$

where

$$\delta_i(\mathbf{r}) = \begin{cases} 1, & \mathbf{r} \in A_i \\ 0, & \textit{otherwise} \end{cases} \quad (2.37)$$

is used to indicate that the value is zero outside the triangle. The alignment of the function on the triangle is represented by the index $k = 1, 2, 3$, which associates the function to one of the three edges of the triangle.

An important property of the RWG functions is that their divergence is finite everywhere. This comes from the fact that there is no discontinuity in the current flow that is crossing the boundaries of the triangles. Considering a single triangle, the current flow is purely tangential to two of the edges with no normal component. At the edge, to which the function is associated, the current flow has normal component; but it is continuous due to the corresponding current flow on the other triangle of the same function. It should be noted that, the value of the normal component becomes unity on the edge due to the normalization factors in (2.35). Therefore, the coefficient related to the n^{th} basis function can be interpreted as the value of the current flow across the n^{th} edge.

In general, divergence of the RWG function can be written as

$$\nabla \cdot \mathbf{b}_n(\mathbf{r}) = \begin{cases} \frac{l_n}{A_n^+}, & \mathbf{r} \in S_n^+ \\ -\frac{l_n}{A_n^-}, & \mathbf{r} \in S_n^- \\ 0, & \textit{otherwise} \end{cases} \quad (2.38)$$

and the total charge associated with the function becomes

$$A_n^+ \frac{l_n}{A_n^+} - A_n^- \frac{l_n}{A_n^-} = 0. \quad (2.39)$$

Equations (2.3) and (2.5) show that the implementation of the MOM with the EFIE requires the divergence operation on the current density. Therefore, it is appropriate to use the RWG functions to expand the current density, so that the

divergence operation is guaranteed to give bounded values. In addition, it will be shown that an efficient implementation of the MOM with the EFIE also requires the divergence of the testing functions. Therefore, it is also suitable to use the RWG functions as the testing functions. In general, the strategy of choosing the same type of basis and testing functions is called the Galerkin method. This method will be extensively used in the MOM and the fast-multipole-method (FMM) implementations presented in this thesis.

Chapter 3

MOM Implementations with the EFIE

This chapter introduces the application of the MOM on the EFIE. The implementation uses the RWG functions for both basis and testing functions. For efficiency, the impedance-matrix expression will be modified and divided into smaller integrals. In the evaluation of these integrals, it will be seen that the integrands tend to diverge as the observation point approaches the source point, due to the singularity of the Green's function. To avoid numerical problems, singularity extraction method will be introduced to divide the problematic inner integral into analytic and numerical parts, each of which can be evaluated without any problem. Various methods will be explained to manage the numerical parts of the inner and outer integrals. Numerical results demonstrating the accuracy of the implementation will be presented at the end of Chapter 4.

3.1 Formulation

For conducting objects, the EFIE is given in Equation (2.5) as

$$\hat{\mathbf{t}} \cdot \int_{S'} d\mathbf{r}' \overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}') = \frac{i}{k\eta} \hat{\mathbf{t}} \cdot \mathbf{E}^{inc}(\mathbf{r}), \quad (3.1)$$

where the observation point is on the surface. To apply the MOM, current density is expressed in terms of the basis functions as

$$\mathbf{J}(\mathbf{r}') = \sum_{n=1}^N a_n \mathbf{b}_n(\mathbf{r}'), \quad (3.2)$$

where $\mathbf{b}_n(\mathbf{r}')$ represents the n^{th} basis function and a_n represents the coefficient determining the weight of this basis function. Then,

$$\hat{\mathbf{t}} \cdot \int_{S'} d\mathbf{r}' \overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}') \cdot \sum_{n=1}^N a_n \mathbf{b}_n(\mathbf{r}') = \frac{i}{k\eta} \hat{\mathbf{t}} \cdot \mathbf{E}^{inc}(\mathbf{r}) \quad (3.3)$$

and this equality can be tested as

$$\int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \int_{S'} d\mathbf{r}' \overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}') \cdot \sum_{n=1}^N a_n \mathbf{b}_n(\mathbf{r}') = \frac{i}{k\eta} \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \mathbf{E}^{inc}(\mathbf{r}), \quad (3.4)$$

where $\mathbf{t}_m(\mathbf{r})$ represents the m^{th} testing function. Rearranging the order of summation and integrals, the MOM system can be formed as

$$\sum_{n=1}^N a_n Z_{mn}^E = v_m^E, \quad (3.5)$$

where

$$Z_{mn}^E = \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \int_{S_n} d\mathbf{r}' \overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{b}_n(\mathbf{r}') \quad (3.6)$$

and

$$v_m^E = \frac{i}{k\eta} \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \mathbf{E}^{inc}(\mathbf{r}). \quad (3.7)$$

Writing the dyadic Green's function explicitly, the expression for the impedance-matrix element can be further divided as

$$\begin{aligned} Z_{mn}^E &= \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \int_{S_n} d\mathbf{r}' \left[\overline{\mathbf{I}} + \frac{\nabla \nabla}{k^2} \right] g(\mathbf{r}, \mathbf{r}') \cdot \mathbf{b}_n(\mathbf{r}') \\ &= \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \int_{S_n} d\mathbf{r}' g(\mathbf{r}, \mathbf{r}') \mathbf{b}_n(\mathbf{r}') \\ &+ \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \int_{S_n} d\mathbf{r}' \frac{\nabla \nabla}{k^2} g(\mathbf{r}, \mathbf{r}') \cdot \mathbf{b}_n(\mathbf{r}'). \end{aligned} \quad (3.8)$$

If the basis and testing functions are chosen to be the RWG functions, the second part of the expression can be evaluated as

$$\begin{aligned}
& \frac{1}{k^2} \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \nabla \left\{ \int_{S_n} d\mathbf{r}' \nabla g(\mathbf{r}, \mathbf{r}') \cdot \mathbf{b}_n(\mathbf{r}') \right\} \\
&= \frac{1}{k^2} \int_{S_m} d\mathbf{r} \nabla \cdot \left\{ \mathbf{t}_m(\mathbf{r}) \int_{S_n} d\mathbf{r}' \nabla g(\mathbf{r}, \mathbf{r}') \cdot \mathbf{b}_n(\mathbf{r}') \right\} \\
&- \frac{1}{k^2} \int_{S_m} d\mathbf{r} \nabla \cdot \mathbf{t}_m(\mathbf{r}) \int_{S_n} d\mathbf{r}' \nabla g(\mathbf{r}, \mathbf{r}') \cdot \mathbf{b}_n(\mathbf{r}') \\
&= \frac{1}{k^2} \int_{\partial S_m} dl' \hat{\mathbf{u}} \cdot \mathbf{t}_m(\mathbf{r}) \int_{S_n} d\mathbf{r}' \nabla g(\mathbf{r}, \mathbf{r}') \cdot \mathbf{b}_n(\mathbf{r}') \\
&- \frac{1}{k^2} \int_{S_m} d\mathbf{r} \nabla \cdot \mathbf{t}_m(\mathbf{r}) \int_{S_n} d\mathbf{r}' \nabla g(\mathbf{r}, \mathbf{r}') \cdot \mathbf{b}_n(\mathbf{r}'), \tag{3.9}
\end{aligned}$$

where $\hat{\mathbf{u}}$ represents the normal direction around the triangle pair of the function, as shown in Figure 3.1.

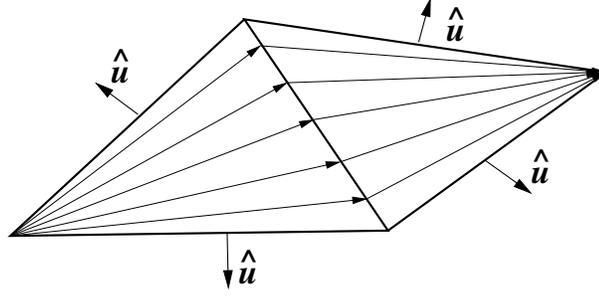


Figure 3.1: RWG function does not have current flow across its boundaries.

Since the RWG function does not have normal current flow across its boundaries,

$$\frac{1}{k^2} \int_{\partial S_m} dl' \hat{\mathbf{u}} \cdot \mathbf{t}_m(\mathbf{r}) \int_{S_n} d\mathbf{r}' \nabla g(\mathbf{r}, \mathbf{r}') \cdot \mathbf{b}_n(\mathbf{r}') = 0 \tag{3.10}$$

and

$$\begin{aligned}
& \frac{1}{k^2} \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \nabla \left\{ \int_{S_n} d\mathbf{r}' \nabla g(\mathbf{r}, \mathbf{r}') \cdot \mathbf{b}_n(\mathbf{r}') \right\} \\
&= \frac{1}{k^2} \int_{S_m} d\mathbf{r} \nabla \cdot \mathbf{t}_m(\mathbf{r}) \int_{S_n} d\mathbf{r}' \nabla g(\mathbf{r}, \mathbf{r}') \cdot \mathbf{b}_n(\mathbf{r}') \tag{3.11}
\end{aligned}$$

since

$$\nabla g(\mathbf{r}, \mathbf{r}') = -\nabla' g(\mathbf{r}, \mathbf{r}'). \tag{3.12}$$

A similar procedure can be applied on the inner integral, so that

$$\begin{aligned}
& \frac{1}{k^2} \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \nabla \left\{ \int_{S_n} d\mathbf{r}' \nabla g(\mathbf{r}, \mathbf{r}') \cdot \mathbf{b}_n(\mathbf{r}') \right\} \\
&= \frac{1}{k^2} \int_{S_m} d\mathbf{r} \nabla \cdot \mathbf{t}_m(\mathbf{r}) \int_{S_n} d\mathbf{r}' \nabla' \cdot \left\{ \mathbf{b}_n(\mathbf{r}') g(\mathbf{r}, \mathbf{r}') \right\} \\
&- \frac{1}{k^2} \int_{S_m} d\mathbf{r} \nabla \cdot \mathbf{t}_m(\mathbf{r}) \int_{S_n} d\mathbf{r}' g(\mathbf{r}, \mathbf{r}') \nabla' \cdot \mathbf{b}_n(\mathbf{r}') \\
&= \frac{1}{k^2} \int_{S_m} d\mathbf{r} \nabla \cdot \mathbf{t}_m(\mathbf{r}) \int_{\partial S_n} dl' \hat{\mathbf{u}} \cdot \left\{ \mathbf{b}_n(\mathbf{r}') g(\mathbf{r}, \mathbf{r}') \right\} \\
&- \frac{1}{k^2} \int_{S_m} d\mathbf{r} \nabla \cdot \mathbf{t}_m(\mathbf{r}) \int_{S_n} d\mathbf{r}' g(\mathbf{r}, \mathbf{r}') \nabla' \cdot \mathbf{b}_n(\mathbf{r}') \\
&= -\frac{1}{k^2} \int_{S_m} d\mathbf{r} \nabla \cdot \mathbf{t}_m(\mathbf{r}) \int_{S_n} d\mathbf{r}' g(\mathbf{r}, \mathbf{r}') \nabla' \cdot \mathbf{b}_n(\mathbf{r}'). \tag{3.13}
\end{aligned}$$

Then, the impedance-matrix expression can be rewritten as

$$\begin{aligned}
Z_{mn}^E &= \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \int_{S_n} d\mathbf{r}' g(\mathbf{r}, \mathbf{r}') \mathbf{b}_n(\mathbf{r}') \\
&- \frac{1}{k^2} \int_{S_m} d\mathbf{r} \nabla \cdot \mathbf{t}_m(\mathbf{r}) \int_{S_n} d\mathbf{r}' g(\mathbf{r}, \mathbf{r}') \nabla' \cdot \mathbf{b}_n(\mathbf{r}'). \tag{3.14}
\end{aligned}$$

Inserting the functions and their divergences explicitly by using Equations (2.36) and (2.38),

$$\begin{aligned}
Z_{ik,jl}^E &= \frac{l_{ik} l_{jl}}{4A_i A_j} \int_{S_i} d\mathbf{r} (\mathbf{r} - \mathbf{r}_{ik}) \cdot \int_{S_j} d\mathbf{r}' g(\mathbf{r}, \mathbf{r}') (\mathbf{r}' - \mathbf{r}_{jl}) \\
&- \frac{1}{k^2} \frac{l_{ik} l_{jl}}{A_i A_j} \int_{S_m} d\mathbf{r} \int_{S_n} d\mathbf{r}' g(\mathbf{r}, \mathbf{r}'), \tag{3.15}
\end{aligned}$$

where i and j indicate that the interaction is between i^{th} and j^{th} triangles, while l and k represent the alignment of the basis and testing functions on these triangles. For an efficient implementation, it is essential to form the calculation loop over the triangles, instead of calculating the interactions between the unknowns. It will be shown that the basic integrals used in forming the impedance-matrix expression can be calculated without any alignment information. Then, constructing the loop over the triangles makes it possible to avoid calculating the same integrals for nine times.

Before going on further, it should be noted that the divergence of a RWG function is finite, so that no problem occurs in the second integral in (3.14). The advantage of using divergence-conforming functions in the EFIE-MOM appears more evidently at this stage. Since the divergence is also a constant, the integral reduces to a very simple form.

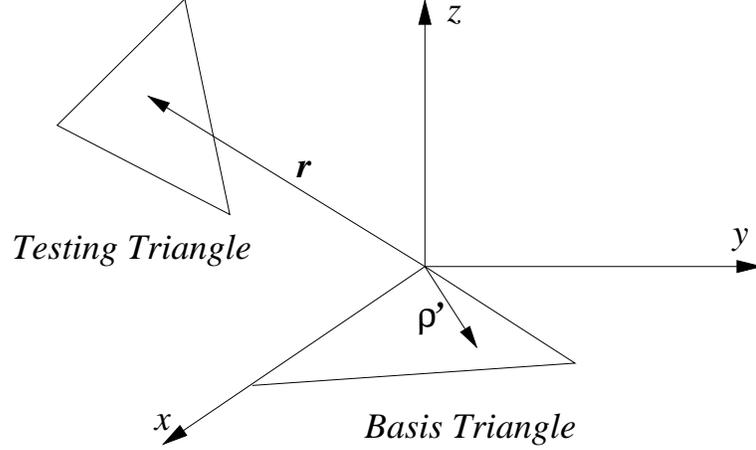


Figure 3.2: Location of the basis and testing triangles after the coordinate transformation.

Finally, a coordinate transformation can be applied, so that the basis triangle lies on the x - y plane with one of its edges on the x -axis, as shown in Figure 3.2. Such a transformation is essential in order to easily evaluate the analytic integrals appearing in the singularity extraction. The final expression for the interaction can be written as

$$\begin{aligned}
Z_{ik,jl}^E &= \frac{l_{ik}l_{jl}}{4A_iA_j} \int_{S_i} d\mathbf{r}(\mathbf{r} - \mathbf{r}_{ik}) \cdot \int_{S_j} d\mathbf{r}'(\boldsymbol{\rho}' - \boldsymbol{\rho}_{jl}) \frac{e^{ikR}}{4\pi R} \\
&\quad - \frac{1}{k^2} \frac{l_{ik}l_{jl}}{A_iA_j} \int_{S_m} d\mathbf{r} \int_{S_n} d\mathbf{r}' \frac{e^{ikR}}{4\pi R} \\
&= \frac{l_{ik}l_{jl}}{4A_iA_j} \int_{S_m} d\mathbf{r}(\boldsymbol{\rho} - \boldsymbol{\rho}_{ik}) \cdot \int_{S_n} d\mathbf{r}'\boldsymbol{\rho}' \frac{e^{ikR}}{4\pi R} \\
&\quad - \frac{l_{ik}l_{jl}}{4A_iA_j} \boldsymbol{\rho}_{jl} \cdot \int_{S_m} d\mathbf{r}(\boldsymbol{\rho} - \boldsymbol{\rho}_{ik}) \int_{S_n} d\mathbf{r}' \frac{e^{ikR}}{4\pi R} \\
&\quad - \frac{1}{k^2} \frac{l_{ik}l_{jl}}{A_iA_j} \int_{S_m} d\mathbf{r} \int_{S_n} d\mathbf{r}' \frac{e^{ikR}}{4\pi R}, \tag{3.16}
\end{aligned}$$

where

$$R = |\mathbf{r} - \boldsymbol{\rho}'| \quad (3.17)$$

is the distance between the observation point and the source point.

3.2 Singularity Extraction

When the observation and source points approach each other, R goes to zero and the integrands of the inner integrals in Equation (3.16) diverges. For a numerical evaluation of these inner integrals, a suitable singularity extraction method has to be applied as

$$\int_{S_j} d\mathbf{r}' \frac{e^{ikR}}{R} = \int_{S_j} d\mathbf{r}' \frac{e^{ikR} - 1}{R} + \int_{S_j} d\mathbf{r}' \frac{1}{R} \quad (3.18)$$

$$\int_{S_j} d\mathbf{r}' \boldsymbol{\rho}' \frac{e^{ikR}}{R} = \int_{S_j} d\mathbf{r}' \boldsymbol{\rho}' \frac{e^{ikR} - 1}{R} + \int_{S_j} d\mathbf{r}' \frac{\boldsymbol{\rho}'}{R}. \quad (3.19)$$

It should be noted that

$$\lim_{R \rightarrow 0} \frac{e^{ikR} - 1}{R} = ik \quad (3.20)$$

so that the first integrals on the right sides of Equations (3.18) and (3.19) can be evaluated numerically without any singularity problem, while other two integrals can be evaluated analytically [11, 12].

Evaluation of $\int_{S_j} d\mathbf{r}' 1/R$

Figure 3.3 shows the source triangle lying on the x - y plane with one of its edges on the x axis, and the projection of the observation point on the x - y plane, which may be located inside or outside the triangle. The integral can be divided into two parts as

$$\int_{S_j} d\mathbf{r}' \frac{1}{R} = \int_{S_j - S_\varepsilon} d\mathbf{r}' \frac{1}{R} + \int_{S_\varepsilon} d\mathbf{r}' \frac{1}{R}, \quad (3.21)$$

where the region S_ε represents the infinitesimal circular area centered at the projection and lying within the source triangle. Therefore, it does not actually exist

when the projection is outside, but it exists as a full circle when the projection is inside the triangle.

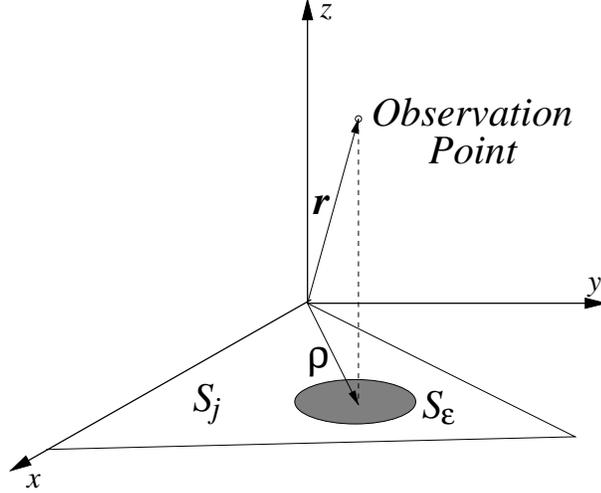


Figure 3.3: Source triangle on the x - y plane and projection of the observation point.

By using the identity

$$\nabla'_S \cdot \left(\frac{R}{P} \hat{\mathbf{P}} \right) = \frac{1}{R}, \quad (3.22)$$

where

$$P = |\mathbf{P}| = |\boldsymbol{\rho} - \boldsymbol{\rho}'|, \quad (3.23)$$

the integral can be evaluated as

$$\begin{aligned} \int_{S_j} d\mathbf{r}' \frac{1}{R} &= \lim_{\varepsilon \rightarrow 0} \int_{S_j - S_\varepsilon} d\mathbf{r}' \nabla'_S \cdot \left(\frac{R}{P} \hat{\mathbf{P}} \right) + \lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} d\mathbf{r}' \frac{1}{R} \\ &= \lim_{\varepsilon \rightarrow 0} \int_{\partial(S_j - S_\varepsilon)} \frac{R}{P} \hat{\mathbf{P}} \cdot \hat{\mathbf{u}} dl' + \lim_{\varepsilon \rightarrow 0} \int_0^{\alpha(\rho)} d\phi \int_0^\varepsilon dP \frac{P}{(P^2 + z^2)^{1/2}} \\ &= \lim_{\varepsilon \rightarrow 0} \int_{\partial(S_j - S_\varepsilon)} \frac{R}{P} \hat{\mathbf{P}} \cdot \hat{\mathbf{u}} dl' + \alpha(\rho) \lim_{\varepsilon \rightarrow 0} [(\varepsilon^2 + z^2)^{1/2} - |z|] \\ &= \sum_{i=1}^3 \mathbf{P}_i^0 \cdot \hat{\mathbf{u}}_i \int_{\partial_i S_j} \frac{R}{P^2} dl' + \lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} \frac{R}{P} \hat{\mathbf{P}} \cdot \hat{\mathbf{u}} dl' \\ &\quad + \alpha(\rho) \lim_{\varepsilon \rightarrow 0} [(\varepsilon^2 + z^2)^{1/2} - |z|] \end{aligned} \quad (3.24)$$

$$\begin{aligned}
&= \sum_{i=1}^3 \mathbf{P}_i^0 \cdot \hat{\mathbf{u}}_i \int_{\partial_i S_j} \frac{R}{P^2} dl' \\
&\quad + \alpha(\rho) \lim_{\varepsilon \rightarrow 0} \left[-(\varepsilon^2 + z^2)^{1/2} + (\varepsilon^2 + z^2)^{1/2} - |z| \right] \\
&= \sum_{i=1}^3 \mathbf{P}_i^0 \cdot \hat{\mathbf{u}}_i \int_{\partial_i S_j} \frac{R}{P^2} dl' - \alpha(\rho) |z|. \tag{3.25}
\end{aligned}$$

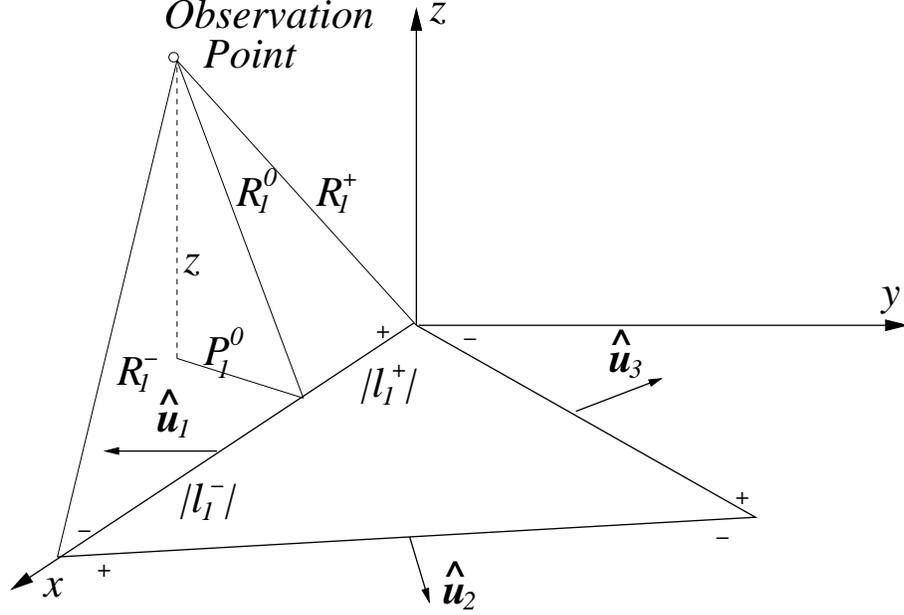


Figure 3.4: Geometric variables introduced to express the results of analytical integrals.

Finally, the integral can be written in terms of the geometric variables shown in Figure 3.4 as

$$\begin{aligned}
\int_{S_j} d\mathbf{r}' \frac{1}{R} &= -\alpha(\rho) |z| + \sum_i \hat{\mathbf{P}}_i^0 \cdot \hat{\mathbf{u}}_i |z| \left[\tan^{-1} \frac{|z| l_i^+}{P_i^0 R_i^+} - \tan^{-1} \frac{|z| l_i^-}{P_i^0 R_i^-} \right] \\
&\quad + \sum_i \hat{\mathbf{P}}_i^0 \cdot \hat{\mathbf{u}}_i \left[P_i^0 \ln \left(\frac{R_i^+ + l_i^+}{R_i^- + l_i^-} \right) \right], \tag{3.26}
\end{aligned}$$

where the variables can be summarized as follows:

- R_i^+ and R_i^- are the distances between the observation point and the end points of i^{th} edge. The superscripts “+” and “-” referring to the end points

of the edge are determined by the right-hand rule applied on the triangle in the z direction.

- R_i^0 is the distance between the observation point and the i^{th} edge, which can be written as

$$R_i^0 = [z^2 + (P_i^0)^2]^{1/2}, \quad (3.27)$$

where P_i^0 is the distance between the projection of the observation point on the x - y plane and the i^{th} edge. In addition, $\hat{\mathbf{P}}_i^0$ is the unit vector pointing along the line between the projections of the observation point on the x - y plane and on the i^{th} edge. The direction of this vector is towards the edge.

- l_i^+ and l_i^- have magnitudes equal to the “+” and “-” segments of the i^{th} edge. These segments are formed by the projection of the observation point on the edge. The signs of l_i^+ and l_i^- are determined by the relative position of this projection compared to the “+” and “-” ends of the edge.

In general,

$$\frac{l_i^\pm}{|l_i^\pm|} = \frac{(\boldsymbol{\rho}_{ei} - \boldsymbol{\rho}_\pm) \cdot (\boldsymbol{\rho}_- - \boldsymbol{\rho}_+)}{|(\boldsymbol{\rho}_{ei} - \boldsymbol{\rho}_\pm) \cdot (\boldsymbol{\rho}_- - \boldsymbol{\rho}_+)|}, \quad (3.28)$$

where $\boldsymbol{\rho}_{ei}$ is the projection of the observation point on the i^{th} edge, while $\boldsymbol{\rho}_+$ and $\boldsymbol{\rho}_-$ represent the “+” and “-” ends of the edge, respectively.

In Equation (3.26), α represents the angular extent of ∂S_ε lying within the triangle, so that it becomes 2π when the projection of the observation point is inside the triangle and zero when it is outside. If the projection is on an edge or vertex, then α takes values between 0 and 2π . These cases are shown in Figure 3.5. In general,

$$\alpha(\rho) = \sum_{i=1}^3 \hat{\mathbf{P}}_i^0 \cdot \hat{\mathbf{u}}_i \left[\tan^{-1} \frac{l_i^+}{P_i^0} - \tan^{-1} \frac{l_i^-}{P_i^0} \right] \quad (3.29)$$

and Equation (3.26) can be rewritten as

$$\int_{S_j} d\mathbf{r}' \frac{1}{R} = - \sum_i \hat{\mathbf{P}}_i^0 \cdot \hat{\mathbf{u}}_i |z| \left[\tan^{-1} \frac{P_i^0 l_i^+}{(R_i^0)^2 + |z| R_i^+} - \tan^{-1} \frac{P_i^0 l_i^-}{(R_i^0)^2 + |z| R_i^-} \right] + \sum_i \hat{\mathbf{P}}_i^0 \cdot \hat{\mathbf{u}}_i \left[P_i^0 \ln \left(\frac{R_i^+ + l_i^+}{R_i^- + l_i^-} \right) \right] \quad (3.30)$$

with the aid of the identity

$$\tan^{-1} \frac{l_i^\pm}{P_i^0} - \tan^{-1} \frac{|z| l_i^\pm}{P_i^0 R_i^\pm} = \tan^{-1} \frac{P_i^0 l_i^\pm}{(R_i^0)^2 + |z| R_i^\pm}. \quad (3.31)$$

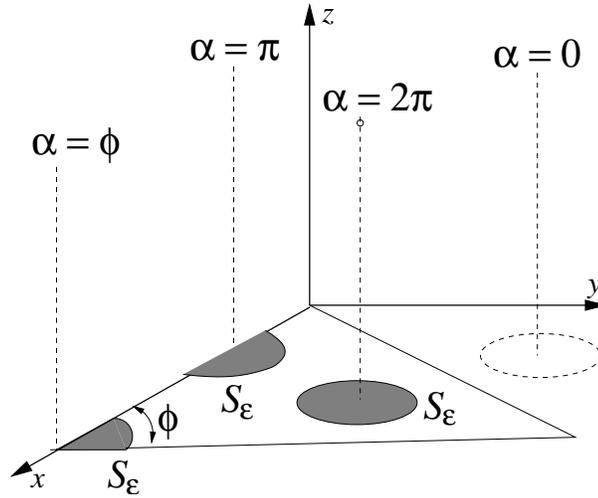


Figure 3.5: Different cases for the projection of the observation point.

Evaluation of $\int_{S_j} d\mathbf{r}' \rho' / R$

This integral can be divided as

$$\int_{S_j} d\mathbf{r}' \frac{\rho'}{R} = \int_{S_j} d\mathbf{r}' \frac{\rho' - \rho}{R} + \rho \int_{S_j} d\mathbf{r}' \frac{1}{R} \quad (3.32)$$

and it should be noted that the second integral has been evaluated. The first integral can further be divided into two integrals as

$$\int_{S_j} d\mathbf{r}' \frac{\rho' - \rho}{R} = \int_{S_j - S_\epsilon} d\mathbf{r}' \frac{\rho' - \rho}{R} + \int_{S_\epsilon} d\mathbf{r}' \frac{\rho' - \rho}{R} \quad (3.33)$$

and the integrand can be written as

$$\frac{\boldsymbol{\rho}' - \boldsymbol{\rho}}{R} = -\nabla'_S R \quad (3.34)$$

so that

$$\begin{aligned} \int_{S_j} d\mathbf{r}' \frac{\boldsymbol{\rho}' - \boldsymbol{\rho}}{R} &= -\lim_{\varepsilon \rightarrow 0} \int_{S-S_\varepsilon} d\mathbf{r}' \nabla'_S R - \lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} d\mathbf{r}' \frac{\boldsymbol{\rho}' - \boldsymbol{\rho}}{R} \\ &= -\lim_{\varepsilon \rightarrow 0} \int_{\partial(S-S_\varepsilon)} R \hat{\mathbf{u}} dl' - \lim_{\varepsilon \rightarrow 0} \int_0^{2\pi} \int_0^\varepsilon \frac{P^2}{[P^2 + z^2]^{1/2}} dP d\phi \\ &= -\sum_{i=1}^3 \hat{\mathbf{u}}_i \int_{\partial_i S} R dl' \\ &= -\frac{1}{2} \sum_{i=1}^3 \hat{\mathbf{u}}_i \left[(R_i^0)^2 \ln \frac{R_i^+ + l_i^+}{R_i^- + l_i^-} + l_i^+ R_i^+ - l_i^- R_i^- \right], \end{aligned} \quad (3.35)$$

where the final expression is reached by considering the geometric entities shown in Figure 3.4. Finally, Equations (3.35) and (3.30) can be combined as shown in (3.32) to complete the evaluation.

3.3 Evaluation of Impedance-Matrix Elements

For an efficient implementation, the expression in Equation (3.16) has to be divided into smaller integrals as

$$\begin{aligned} Z_{ik,jl}^E &= \frac{l_{ik} l_{jl}}{4A_i A_j} \left\{ \left[-\frac{4}{k^2} + x_{ik} x_{jl} + y_{ik} y_{jl} \right] I_{e1} + I_{e2} + I_{e3} \right. \\ &\quad \left. - x_{jl} I_{e4} - y_{jl} I_{e5} - x_{ik} I_{e6} - y_{ik} I_{e7} \right\}, \end{aligned} \quad (3.36)$$

where the basic integrals can be listed as

$$\begin{aligned} I_{e1} &= \int_{S_m} d\mathbf{r} \int_{S_n} I_{e1}^{in}, & I_{e2} &= \int_{S_m} d\mathbf{r} \int_{S_n} x I_{e2}^{in}, & I_{e3} &= \int_{S_m} d\mathbf{r} \int_{S_n} y I_{e3}^{in}, \\ I_{e4} &= \int_{S_m} d\mathbf{r} \int_{S_n} x I_{e1}^{in}, & I_{e5} &= \int_{S_m} d\mathbf{r} \int_{S_n} y I_{e1}^{in}, & I_{e6} &= \int_{S_m} d\mathbf{r} \int_{S_n} I_{e2}^{in}, \\ I_{e7} &= \int_{S_m} d\mathbf{r} \int_{S_n} I_{e3}^{in}, \end{aligned} \quad (3.37)$$

and the inner integrals can be written as

$$\begin{aligned}
 I_{e1}^{in} &= \int_{S_n} d\mathbf{r}' \frac{e^{ikR}}{4\pi R}, \\
 I_{e2}^{in} &= \int_{S_n} d\mathbf{r}' x' \frac{e^{ikR}}{4\pi R}, \\
 I_{e3}^{in} &= \int_{S_n} d\mathbf{r}' y' \frac{e^{ikR}}{4\pi R}.
 \end{aligned} \tag{3.38}$$

Equation (3.36) indicates that the integrals can be evaluated without any alignment information of the basis and testing functions. The evaluation of these seven basic integrals require the evaluation of three inner integrals. Then, their values can be used in forming the contributions to different locations in the impedance matrix. It should be noted that a triangle may belong up to three functions so that the contribution may be added up to nine different locations.

3.4 Numerical Evaluation of Integrals

The inner integrals in Equation (3.38) are divided into numerical and analytical parts as shown in Equation (3.18). The numerical parts of these integrals can be evaluated by using Gaussian quadrature rules. There are many of rules employing different numbers of points on the triangle. In addition, the rules using different or same number of points can be combined to obtain new rules using larger number of points.

On the other hand, using a static rule in the implementation may result in incorrect integrations or inefficiencies. For an integration, if the number of points are less than the required, the final value may not converge to the exact value within a small error. In the other extreme, many integration points may be used, although few of them could already satisfy the convergence criteria. In addition, the distribution of the points may not be uniform and it would be better to take more samples, where the function changes rapidly, while taking less points for

smoother portions. As a result, instead of using high-ordered Gaussian quadrature rules, an adaptive algorithm can be used to control the integration error efficiently. Two different methods of adaptive integration will be presented here.

3.4.1 Adaptive Integration: Method 1

As shown in Figure 3.6, following steps are applied to evaluate the inner integrals on the basis triangle:

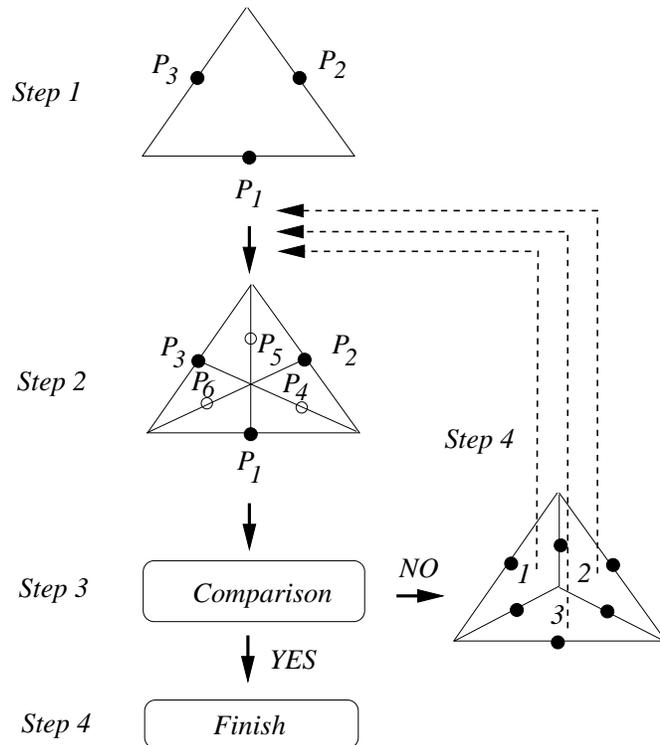


Figure 3.6: Adaptive integration method 1.

1. On the triangle, three points are chosen, each of which is at the middle of an edge. As the Gaussian quadrature rule with three points states, the value of the integral is

$$I_3 = \left[f(p1) + f(p2) + f(p3) \right] \frac{A_j}{3}, \quad (3.39)$$

where A_j is the area of the triangle.

2. Three more points are taken on the medians of the triangle. These points are located at $1/3$ of the median nearer to the vertices of the triangle. The value of the integral with six points becomes

$$I_6 = \left[f(p1) + f(p2) + f(p3) + 2f(p4) + 2f(p5) + 2f(p6) \right] \frac{A_j}{9}. \quad (3.40)$$

3. A comparison is performed on the 6-point versus 3-point integration values. If the error, which can be defined as

$$E_{3,6} = \frac{|I_6 - I_3|}{|I_6|}, \quad (3.41)$$

is less than the given error threshold, then the integration can be assumed to converge. Then, I_6 can be returned as the value of the integral. If the error is large, more points should be sampled in order to achieve the desired convergence.

4. If the convergence is not satisfied, each of the three subtriangles can be considered as separate domains, on which the integrations are to converge individually, similar to the main triangle after first step. Three points are already evaluated on each of these subtriangles, and the process continues by taking three more samples to reach 6-point integrations.
5. The adaptive routine continues by comparing the results of the 3-point and 6-point integrations on subtriangles. Whenever convergence is reached in a subtriangle, one branch of the adaptive integration stops there, but the algorithm may go on in other subtriangles.

It should be noted that the algorithm completely stops only when the convergence is accomplished in all of the subtriangles of the basis triangle. In addition, the distribution of the points on the triangle is not uniform and more points are taken, where the function changes more rapidly. This is due to the fact that the algorithm continues to sample denser only on the non-converged regions.

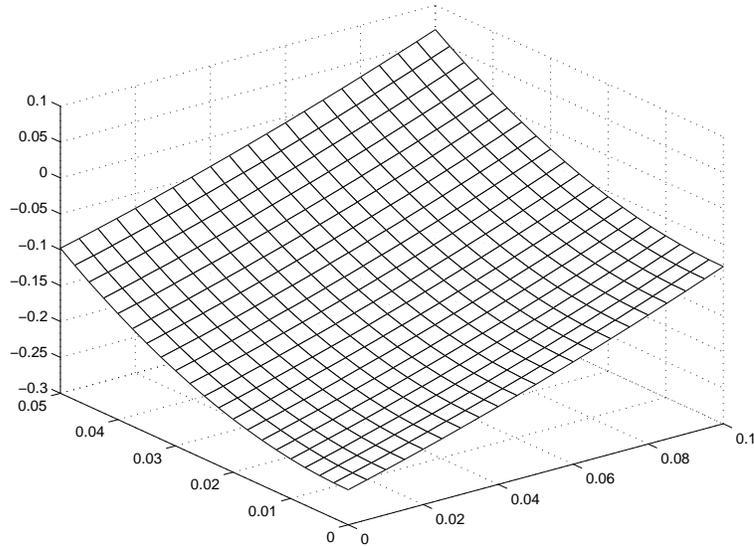


Figure 3.7: Function to be integrated on the triangular domain.

A disadvantage of this method appears when the aspect ratio of the integration domain gets larger. Figure 3.7 shows a function to be integrated on the triangular domain with node coordinates $(0, 0, 0)$, $(0.083, 0, 0)$, and $(0.042, 0.048, 0)$. As shown in Figure 3.8, the method continues to make divisions towards the edge on the x axis, but does not obtain a convergence. Actually, on the subtriangles near the edge, the new points are not located efficiently and they are very near to previous points. Therefore, the method tends to continue infinitely until an external stop. The second adaptive integration method avoids such divisions that create domains with high aspect ratios.

3.4.2 Adaptive Integration: Method 2

This method, described in Figure 3.9, consists of the following steps:

1. The algorithm starts with taking five points on the main triangle. The

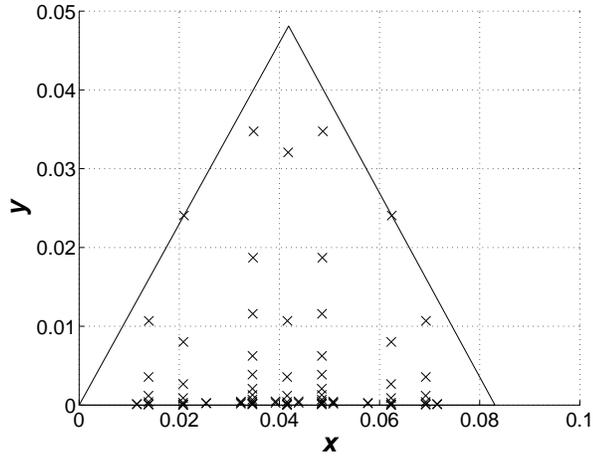


Figure 3.8: Integration points taken by the first adaptive method to evaluate the integral of the function shown in Figure 3.7

value of the integral with five points can be written as

$$I_5 = \left[f(p1) + f(p2) + f(p3) + f(p4) + 2f(p5) \right] \frac{A_j}{6}. \quad (3.42)$$

It should be indicated that two of the points are located on the longest edge of the triangle.

2. Six more points are taken on the triangle. To determine the location of these new points, the medians are drawn to the longest edge in each subtriangle. Then 9-point integration can be written as

$$I_9 = \left[f(p1) + f(p2) + 2f(p5) + f(p6) + f(p7) + 2f(p8) + f(p9) + f(p10) + 2f(p11) \right] \frac{A_j}{12}. \quad (3.43)$$

It should be noted that the third and fourth points are not used in the 9-point integration.

3. A comparison on 5-point and 9-point integrations is made to determine the error as in the previous algorithm. If the convergence criterion is not satisfied, the process continues.

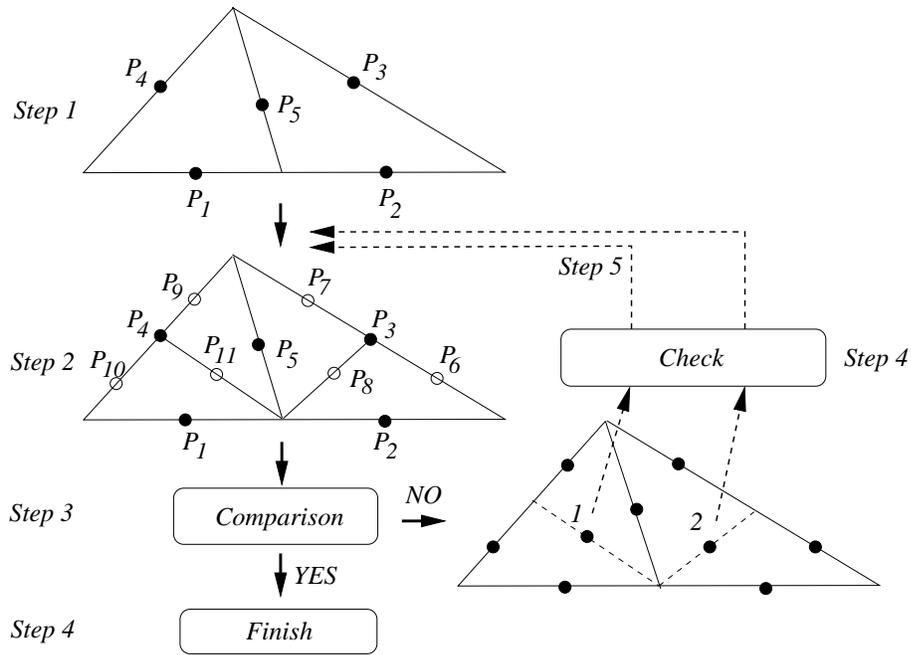


Figure 3.9: Adaptive integration method 2.

4. Before taking more points, each subtriangle is checked by comparing their 3-point and 5-point integrations. It should be realized that the points to calculate these integrations have already been sampled in the previous steps.
5. The algorithm continues on the subtriangles that have large errors between their 3-point and 5-point integrations. The subtriangle at this stage is similar to the main triangle after the first step.
6. The algorithm continues with the comparisons of 5-point and 9-point integrations on subtriangles. Different from the first algorithm, if a parent triangle does not satisfy the convergence criteria, the procedure does not continue on all of the subtriangles.

This algorithm guarantees the divisions to be done appropriately so that the aspect ratios of the sub-domains do not grow. Figure 3.10 shows the integration points taken on the triangle for the function in Figure 3.7, for which the first algorithm fails. As it can be seen, there is no inefficient crowding of the integration

points with the new algorithm. The control stage to determine the longest edge of the triangles creates an extra computational load, but this load is negligible when compared to the inefficient sampling of the integrand.

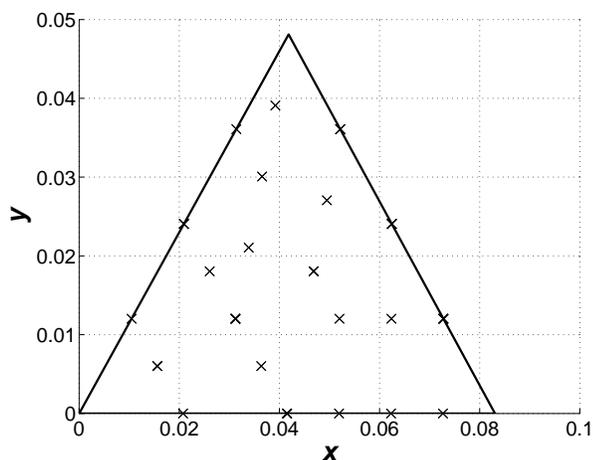


Figure 3.10: Integration points taken by the second adaptive method to evaluate the integral of the function shown in Figure 3.7

In both of the algorithms, the process continues until convergence in every region is guaranteed, which may lead to unnecessary usage of time. Convergence in a small region may not be so important since the result of this region may be negligible compared to the overall integration. For the first algorithm, this inefficiency can be avoided by carrying the 6-point integration result of the main triangle to each level of the adaptive method in order to compare the integration results of the subtriangles and decide on the necessity of further division. In the second algorithm, the carried value is the 9-point integration result of the main triangle, which is more reliable.

Numerical parts of the inner integrals given in (3.38) are evaluated in a parallel manner, which means that they are calculated simultaneously within the same adaptive algorithm. Choosing the source point, the integrands of these

integrals are determined with a single evaluation of the extracted Green's function. After the evaluation of the numerical parts, they are combined with the analytical parts to complete the inner integrals. Then, these three inner integrals are used to form the seven outer integrals given in (3.37). The evaluation of the outer integrals can be performed either by high-ordered Gaussian quadratures or adaptive integration. These integrals are also evaluated simultaneously, since they have also related integrands.

3.5 Excitation

There are various excitation types that can be implemented in the MOM using the EFIE formulation. For the scattering problems, plane-wave excitation can be used, while the radiation problems can be implemented with either delta-gap voltage sources or current sources defined on the geometry.

3.5.1 Plane-Wave Excitation

In the plane-wave excitation, the external source is a plane wave having a certain propagation direction. Rewriting Equation (3.7),

$$v_m^E = \frac{i}{k\eta} \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \mathbf{E}^{inc}(\mathbf{r}). \quad (3.44)$$

Incident electric field in the above can be written as

$$\mathbf{E}^{inc}(\mathbf{r}) = \mathbf{E}_0 e^{i\mathbf{k} \cdot \mathbf{r}}, \quad (3.45)$$

where \mathbf{E}_0 is a complex vector, which satisfies

$$\mathbf{E}_0 \cdot \mathbf{k} = 0. \quad (3.46)$$

Then, the equation

$$v_m^E = \frac{i}{k\eta} \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \mathbf{E}_0 e^{i\mathbf{k} \cdot \mathbf{r}} \quad (3.47)$$

can be used to fill the right-side vector in the MOM system.

3.5.2 Delta-Gap Excitation

Delta-gap sources can be located on the geometry to create voltage differences between two separate parts of the geometry. Figure 3.11 shows a delta-gap source located on the m^{th} edge of the geometry. For this type of source to be meaningful, the only connection between the geometry parts, separated by the source, should be through this edge. In the infinitely narrow gap between the triangles of the m^{th} edge, some electric field is assumed to exist. The voltage difference is induced by this electric field to excite the conductor. Incident field for delta-gap source

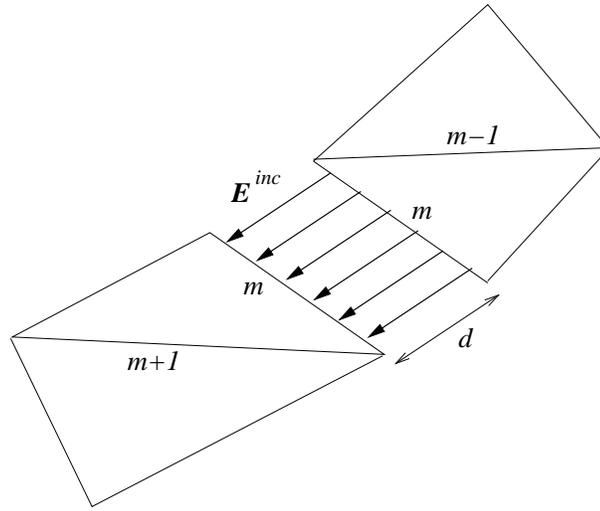


Figure 3.11: Delta-gap source defined on the m^{th} edge.

can be written as

$$\mathbf{E}^{\text{inc}}(\mathbf{r}) = \hat{\mathbf{u}}_e \delta(\mathbf{r} - \mathbf{r}_m) / d \quad (3.48)$$

where d represents the gap distance and $\hat{\mathbf{u}}_e$ is the unit vector perpendicular to the edge. $\delta(\mathbf{r} - \mathbf{r}_m)$ in the expression indicates that the electric field is only located on the m^{th} edge, inside the infinitely narrow gap between the triangles. Then,

$$v_m^E = \lim_{d \rightarrow 0} \left\{ \frac{i}{k\eta} \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \hat{\mathbf{u}}_e \delta(\mathbf{r} - \mathbf{r}_m) / d \right\} = \frac{il_m}{k\eta} \quad (3.49)$$

since the normal component of RWG function becomes unity on its edge.

3.5.3 Current Sources

Similar to the delta-gap source, various current sources can be located on the geometry. Figure 3.12 shows four different implementations that can be summarized as follows:

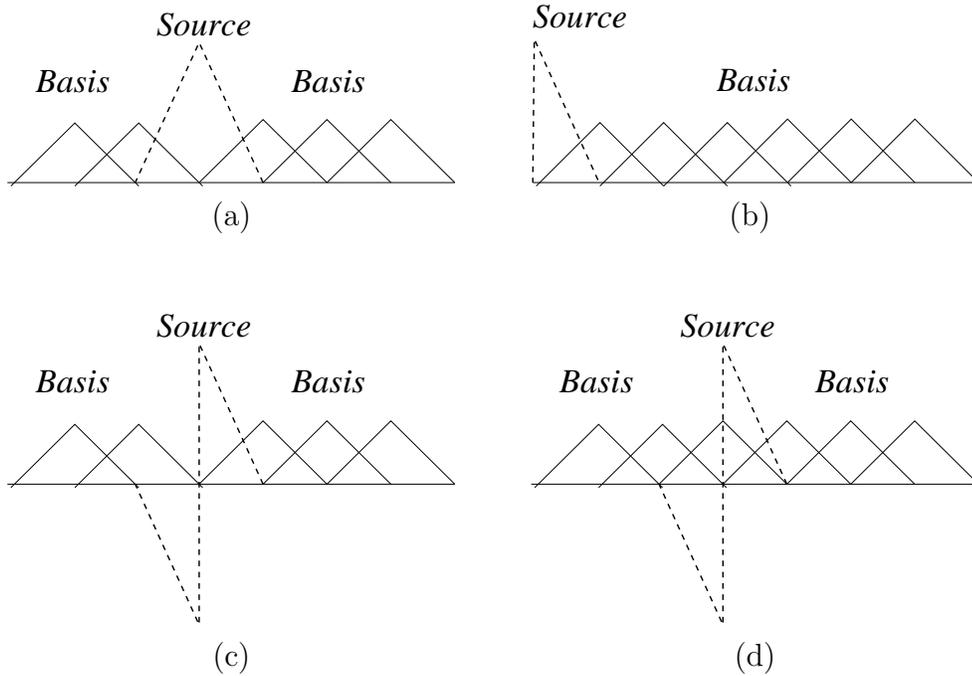


Figure 3.12: Current sources defined with the RWG functions.

- Type 1: This type of current source is implemented by forcing one of the basis functions to have a certain coefficient.
- Type 2: This type of current source is similar to the first type, but the current source is now located on an edge. This can be accomplished by defining a half basis function and assigning a coefficient for it.
- Type 3: This type of current source is implemented by dividing one of the basis functions into two, and reversing the current flow in one of them. The currents on the two triangles may flow towards the common edge (sink) or away from the edge.

- Type 4: If another basis function is defined at the source location in type 3, the electrical connectivity is maintained between the parts of the geometry on the different sides of the source and this results in another type of current sources.

In Figure 3.12, the basis functions are represented by triangles. Corresponding physical equivalences for the types of the current sources are shown in Figure 3.13.

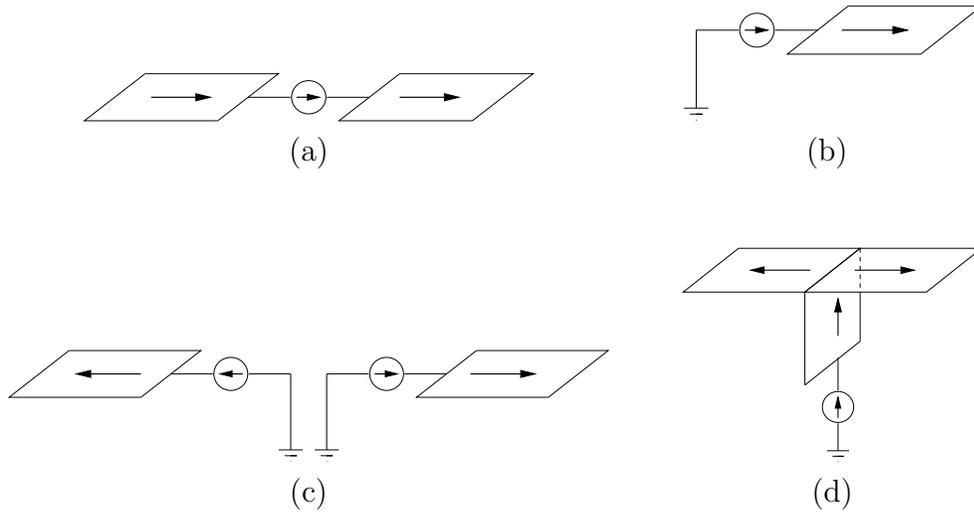


Figure 3.13: Corresponding physical equivalences for the current sources shown in Figure 3.12.

3.6 Computational Analysis of the MOM

For an N -unknown problem, the MOM generates an impedance matrix that has N^2 elements, each of which is calculated separately. Then, both the processing time and the memory requirement appears to be $O(N^2)$ to form this matrix. After that, the system has to be solved either by using a direct method such as Gaussian elimination, or by using an iterative solver. The direct solution involves $O(N^3)$ floating-point operations (FLOPs), while the iterative solution requires

matrix-vector multiplication in each iteration, which can be done with $O(N^2)$ operations. Therefore, processing time for the solution becomes $O(N^3)$ for direct methods and $O(IN^2)$ for iterative methods, where I is the number of iterations. On the other hand, it is usually observed that $I = O(N)$ and $O(IN^2) = O(N^3)$.

In conclusion, computational requirements of the MOM increases very rapidly as the problem size gets larger. The requirements will be quantified at the end of Chapter 5, together with the requirements for the fast multipole method in order to make comparisons.

3.7 RCS Calculations for RWG Basis Function

This section presents the summary for calculating the radar cross section (RCS) of an object due to the induced current represented by the RWG the functions. In the far-field, $r \gg \lambda$ and

$$\mathbf{E}(\mathbf{r}) = ik\eta[\hat{\boldsymbol{\theta}}\hat{\boldsymbol{\theta}} \cdot \mathbf{F}(\theta, \phi) + \hat{\boldsymbol{\phi}}\hat{\boldsymbol{\phi}} \cdot \mathbf{F}(\theta, \phi)]g(r), \quad (3.50)$$

where

$$g(r) = \frac{e^{ikr}}{r} \quad (3.51)$$

and $\mathbf{F}(\theta, \phi)$ is the vector current moment, which can be written as

$$\mathbf{F}(\theta, \phi) = \int_{S'} \mathbf{J}(\mathbf{r}') e^{-ik\hat{\mathbf{r}} \cdot \mathbf{r}'}. \quad (3.52)$$

If the RWG functions are used to expand the current, the expression for vector current moment for a basis triangle can be written with the aid of Equation (2.36) as

$$\mathbf{F}(\theta, \phi) = \pm a_{jl} \frac{l_{jl}}{2A_j} \int_{S'_j} (\mathbf{r}' - \mathbf{r}_{jl}) e^{-ik\hat{\mathbf{r}} \cdot \mathbf{r}'}, \quad (3.53)$$

where a_{jl} indicates the coefficient related to the basis function that resides on the l^{th} edge of the j^{th} triangle. This expression can be evaluated analytically

without any numerical treatment. In addition, the vector current moments for the basis functions can be summed directly to find the vector current moment of the overall current distribution.

Finally, the RCS can be calculated as

$$\begin{aligned}
 RCS(\theta, \phi) &= 4\pi r^2 P^{rad}(\mathbf{r}) = 4\pi r^2 |\mathbf{E}(\mathbf{r})|^2 \\
 &= \frac{k^2 \eta^2}{4\pi} |\hat{\boldsymbol{\theta}} \hat{\boldsymbol{\theta}} \cdot \mathbf{F}(\theta, \phi) + \hat{\boldsymbol{\phi}} \hat{\boldsymbol{\phi}} \cdot \mathbf{F}(\theta, \phi)|^2. \quad (3.54)
 \end{aligned}$$

To check the accuracy of the implementations, one way is to calculate the RCS of the problem by using the induced current on the geometry. This will be done in the next chapter to compare the MOM implementations formulated by various integral equations.

Chapter 4

MOM Implementations with the MFIE and the CFIE

The MOM can also be used to solve electromagnetic scattering problems formulated by the MFIE. However, the MFIE introduces additional difficulties that do not appear in the EFIE formulation. Due to the cross product term, efficient implementation of the integrals become more difficult and the number of basic integrals, each of which has to be evaluated independently, increases drastically. Moreover, the MFIE contains a $1/R^2$ singularity, which causes extra numerical problems that was not encountered in the EFIE [10].

This chapter introduces two different implementations of the MOM using the MFIE formulation. The first implementation uses the Galerkin method and the RWG functions, while the second one uses point testing functions. Singularity extraction for the new integrals will be introduced in the first implementation. However, it can be omitted entirely in the second implementation, which has simpler expressions for the matrix elements but lower accuracy. These two will also be combined with the EFIE implementations to form the CFIE implementations and the results will be shown at the end of the chapter to make comparisons on

accuracy and efficiency.

4.1 MFIE-MOM Implementation with RWG Testing Function

The MFIE formulation can be used in the MOM in a manner similar to the EFIE formulation. Impedance-matrix expression includes double integration and the inner part contains a singularity that should be extracted. It will be observed that the solid angle dependence of the MFIE appears at this stage, in the evaluation of the analytic integrals. Different from the EFIE formulation, some of the outer integrals may also have a singularity in the MFIE formulation and it may be chosen to apply singularity extraction for these integrals. This singularity is weak so that is also possible to overcome it by high sampling of the integrands, while controlling the location of the samples to avoid numerical errors.

For efficiency, the overall expression for an impedance-matrix element will be divided into basic integrals and interactions between pairs of triangles will be calculated instead of the interactions between pairs of unknowns. It will also be observed that the self-interaction of a triangle does not require explicit implementation of singularity extraction. This is due to the fact that an infinitesimal current density on a planar surface produces only perpendicular magnetic field on that plane, except in its own location. Therefore, in self-interactions, only the limit value in Equation (2.10) is tested by the testing function, while the principle value is not.

4.1.1 Formulation

The MFIE for closed conducting surfaces was shown in Equation (2.8) as

$$\mathbf{J}(\mathbf{r}) - \hat{\mathbf{n}} \times \int_S d\mathbf{r}' \mathbf{J}(\mathbf{r}') \times \nabla' g(\mathbf{r}, \mathbf{r}') = \hat{\mathbf{n}} \times \mathbf{H}^{inc}(\mathbf{r}) \quad (4.1)$$

when the observation point approaches the surface from outside. The MOM can be applied to this equation by expressing the current density in terms of basis functions (\mathbf{b}_n) and then testing both sides of the equation by using testing functions (\mathbf{t}_m) as

$$\begin{aligned} \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \hat{\mathbf{n}} \times \mathbf{H}^{inc}(\mathbf{r}) &= \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \sum_{n=1}^N a_n \mathbf{b}_n(\mathbf{r}) \\ &- \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \hat{\mathbf{n}} \times \int_S d\mathbf{r}' \sum_{n=1}^N a_n \mathbf{b}_n(\mathbf{r}') \times \nabla' g(\mathbf{r}, \mathbf{r}'), \end{aligned} \quad (4.2)$$

where N is the number of unknown coefficients. Interchanging the order of summations and integrations, the right side of the equation can be combined and the MOM system can be formed as

$$\sum_{n=1}^N a_n Z_{mn}^M = v_m^M, \quad (4.3)$$

where

$$\begin{aligned} Z_{mn}^M &= \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \mathbf{b}_n(\mathbf{r}) \\ &- \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \hat{\mathbf{n}} \times \int_{S_n} d\mathbf{r}' \mathbf{b}_n(\mathbf{r}') \times \nabla' g(\mathbf{r}, \mathbf{r}') \end{aligned} \quad (4.4)$$

and

$$v_m^M = \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \hat{\mathbf{n}} \times \mathbf{H}^{inc}(\mathbf{r}). \quad (4.5)$$

If the RWG basis and testing functions are used, interaction between a basis and a testing triangle (part of an impedance-matrix element) can be written by using the expression for the RWG functions in Equation (2.36) as

$$\begin{aligned} Z_{ik,jl}^M &= \delta_{ij} \int_{S_i} d\mathbf{r} \frac{l_{ik}(\mathbf{r} - \mathbf{r}_{ik})}{2A_i} \cdot \frac{l_{jl}(\mathbf{r} - \mathbf{r}_{jl})}{2A_j} \\ &- \int_{S_i} d\mathbf{r} \frac{l_{ik}(\mathbf{r} - \mathbf{r}_{ik})}{2A_i} \cdot \hat{\mathbf{n}} \times \int_{S_j} d\mathbf{r}' \frac{l_{jl}(\mathbf{r}' - \mathbf{r}_{jl})}{2A_j} \times \nabla' g(\mathbf{r}, \mathbf{r}'), \end{aligned} \quad (4.6)$$

where i and j are used to indicate that the interaction is between the i^{th} testing and the j^{th} basis triangles, while k and l represent the alignment of basis and testing functions on these triangles. The Kronecker delta function in the first term indicates that the basis and testing functions have nonzero values only inside their respective triangles.

At this stage, it can be assumed that a coordinate transformation is applied on the basis and testing triangles so that the basis triangle lies on the x - y plane and one of its edges lies on the x axis, as shown in Figure 3.2. Such an operation is essential in order to simplify the numerical operations and singularity extractions for the inner integrals. Then,

$$\begin{aligned} Z_{ik,jl}^M &= \delta_{ij} \frac{l_{ik}l_{jl}}{4A_iA_j} \int_{S_i} d\mathbf{r} (\mathbf{r} - \mathbf{r}_{ik}) \cdot (\mathbf{r} - \boldsymbol{\rho}_{jl}) \\ &- \frac{l_{ik}l_{jl}}{4A_iA_j} \int_{S_i} d\mathbf{r} (\mathbf{r} - \mathbf{r}_{ik}) \cdot \hat{\mathbf{n}} \times \int_{S_j} d\mathbf{r}' (\boldsymbol{\rho}' - \boldsymbol{\rho}_{jl}) \times (\mathbf{r} - \boldsymbol{\rho}') \frac{(1 - ikR)e^{ikR}}{4\pi R^3}, \end{aligned} \quad (4.7)$$

where

$$R = |\mathbf{r} - \mathbf{r}'| \quad (4.8)$$

is the distance between the observation point and the source point, and the gradient of the Green's function has been written explicitly. Finally, by using the identity

$$(\boldsymbol{\rho}' - \boldsymbol{\rho}_{jl}) \times (\mathbf{r} - \boldsymbol{\rho}') = -(\mathbf{r} - \boldsymbol{\rho}_{jl}) \times (\boldsymbol{\rho}' - \mathbf{r}), \quad (4.9)$$

the interaction between a pair of basis and testing triangles can be written as

$$\begin{aligned} Z_{ik,jl}^M &= \delta_{ij} \frac{l_{ik}l_{jl}}{4A_iA_j} \int_{S_i} d\mathbf{r} (\mathbf{r} - \mathbf{r}_{ik}) \cdot (\mathbf{r} - \boldsymbol{\rho}_{jl}) \\ &+ \frac{l_{ik}l_{jl}}{16\pi A_iA_j} \int_{S_i} d\mathbf{r} (\mathbf{r} - \mathbf{r}_{ik}) \cdot \hat{\mathbf{n}} \times (\mathbf{r} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' (\boldsymbol{\rho}' - \mathbf{r}) \frac{(1 - ikR)e^{ikR}}{R^3}. \end{aligned} \quad (4.10)$$

4.1.2 Singularity Extraction

When the observation and source points approach each other and R goes to zero, the integrand of the inner integral in Equation (4.10) diverges. This singularity can be extracted by dividing the integral into three smaller parts as

$$\int_{S_j} d\mathbf{r}'(\boldsymbol{\rho}' - \mathbf{r})f_m(R) = \int_{S_j} d\mathbf{r}'(\boldsymbol{\rho}' - \mathbf{r})f_{m1}(R) + \frac{k^2}{2} \int_{S_j} d\mathbf{r}'(\boldsymbol{\rho}' - \mathbf{r})f_{m2}(R) + \int_{S_j} d\mathbf{r}'(\boldsymbol{\rho}' - \mathbf{r})f_{m3}(R), \quad (4.11)$$

where

$$\begin{aligned} f_{m1}(R) &= \frac{(1 - ikR)e^{ikR} - (1 + k^2R^2/2)}{R^3} \\ f_{m2}(R) &= \frac{1}{R} \\ f_{m3}(R) &= \frac{1}{R^3} \\ f_m(R) &= f_{m1}(R) + f_{m2}(R) + f_{m3}(R) = \frac{(1 - ikR)e^{ikR}}{R^3}. \end{aligned} \quad (4.12)$$

As R goes to zero, the function $f_{m1}(R)$ has a finite limit value

$$\lim_{R \rightarrow 0} f_{m1}(R) = \lim_{R \rightarrow 0} \left[\frac{(1 - ikR)e^{ikR} - (1 + k^2R^2/2)}{R^3} \right] = \frac{ik^3}{3} \quad (4.13)$$

and the first integral in Equation (4.11) can be evaluated numerically without any singularity problem, while the second and the third integrals will be evaluated analytically. The latter integrals can be further divided as

$$\begin{aligned} \int_{S_j} d\mathbf{r}'(\boldsymbol{\rho}' - \mathbf{r})f_{m2}(R) &= \int_{S_j} d\mathbf{r}'(\boldsymbol{\rho}' - \boldsymbol{\rho})f_{m2}(R) - \int_{S_j} d\mathbf{r}'\hat{\mathbf{z}}zf_{m2}(R) \\ \int_{S_j} d\mathbf{r}'(\boldsymbol{\rho}' - \mathbf{r})f_{m3}(R) &= \int_{S_j} d\mathbf{r}'(\boldsymbol{\rho}' - \boldsymbol{\rho})f_{m3}(R) - \int_{S_j} d\mathbf{r}'\hat{\mathbf{z}}zf_{m3}(R). \end{aligned}$$

The integrals involving the $1/R$ singularity were evaluated in the context of the EFIE formulation in Section 3.2. The other two integrals will be evaluated here in a very similar manner.

Evaluation of $\int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z f_{m3}(R)$

Considering Figure 3.3, the integral can be partitioned as

$$\int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z \frac{1}{R^3} = \hat{\mathbf{z}} z \left(\int_{S_j - S_\varepsilon} d\mathbf{r}' \frac{1}{R^3} + \int_{S_\varepsilon} d\mathbf{r}' \frac{1}{R^3} \right), \quad (4.14)$$

where the region S_ε is the infinitesimal area centered at the projection of the observation point on the x - y plane. It should be recalled that this area may be a circle, part of a circle or may not exist according to the location of the projection with respect to the triangle. Then, the identity

$$\nabla'_S \cdot \left(\frac{1}{RP} \hat{\mathbf{P}} \right) = -\frac{1}{R^3} \quad (4.15)$$

with

$$P = |\mathbf{P}| = |\boldsymbol{\rho} - \boldsymbol{\rho}'| \quad (4.16)$$

can be used, so that

$$\begin{aligned} \int_{S_j} d\mathbf{r}' \frac{1}{R^3} &= -\lim_{\varepsilon \rightarrow 0} \int_{S_j - S_\varepsilon} d\mathbf{r}' \nabla'_S \cdot \left(\frac{1}{RP} \hat{\mathbf{P}} \right) + \lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} d\mathbf{r}' \frac{1}{R^3} \\ &= -\lim_{\varepsilon \rightarrow 0} \int_{\partial(S_j - S_\varepsilon)} \frac{1}{RP} \hat{\mathbf{P}} \cdot \hat{\mathbf{u}} dl' + \lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} \frac{1}{(z^2 + P^2)^{3/2}} P dP d\phi \\ &= -\lim_{\varepsilon \rightarrow 0} \int_{\partial(S_j - S_\varepsilon)} \frac{1}{RP} \hat{\mathbf{P}} \cdot \hat{\mathbf{u}} dl' + \lim_{\varepsilon \rightarrow 0} \left[\frac{\alpha(\rho)}{|z|} - \frac{\alpha(\rho)}{(z^2 + \varepsilon^2)^{1/2}} \right] \\ &= -\sum_{i=1}^3 \hat{\mathbf{P}}_i^0 \cdot \hat{\mathbf{u}}_i P_i^0 \int_{\partial_i S} \frac{1}{P^2 R} dl' - \lim_{\varepsilon \rightarrow 0} \int_{\partial S_\varepsilon} \frac{1}{RP} \hat{\mathbf{P}} \cdot \hat{\mathbf{u}} dl' \\ &\quad + \lim_{\varepsilon \rightarrow 0} \left[\frac{\alpha(\rho)}{|z|} - \frac{\alpha(\rho)}{(z^2 + \varepsilon^2)^{1/2}} \right] \\ &= -\sum_{i=1}^3 \hat{\mathbf{P}}_i^0 \cdot \hat{\mathbf{u}}_i P_i^0 \int_{\partial_i S} \frac{1}{P^2 R} dl' + \lim_{\varepsilon \rightarrow 0} \int_{\partial S_\varepsilon} \frac{1}{R} d\phi \\ &\quad + \lim_{\varepsilon \rightarrow 0} \left[\frac{\alpha(\rho)}{|z|} - \frac{\alpha(\rho)}{(z^2 + \varepsilon^2)^{1/2}} \right] \\ &= -\sum_{i=1}^3 \hat{\mathbf{P}}_i^0 \cdot \hat{\mathbf{u}}_i P_i^0 \int_{\partial_i S} \frac{1}{P^2 R} dl' \\ &\quad + \lim_{\varepsilon \rightarrow 0} \left[\frac{\alpha(\rho)}{|z|} - \frac{\alpha(\rho)}{(z^2 + \varepsilon^2)^{1/2}} + \frac{\alpha(\rho)}{(z^2 + \varepsilon^2)^{1/2}} \right] \\ &= -\sum_{i=1}^3 \hat{\mathbf{P}}_i^0 \cdot \hat{\mathbf{u}}_i P_i^0 \int_{\partial_i S} \frac{1}{P^2 R} dl' + \frac{\alpha(\rho)}{|z|}. \end{aligned}$$

The final result can be written in terms of the geometric variables shown in Figure 3.4 and the integral becomes

$$\int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z \frac{1}{R^3} = \hat{\mathbf{z}} \frac{|z|}{z} \alpha(\rho) - \hat{\mathbf{z}} \frac{|z|}{z} \sum_{i=1}^3 \hat{\mathbf{P}}_i^0 \cdot \hat{\mathbf{u}}_i \left(\tan^{-1} \frac{|z| l_i^+}{P_i^0 R_i^+} - \tan^{-1} \frac{|z| l_i^-}{P_i^0 R_i^-} \right). \quad (4.17)$$

It should be noted that $\alpha(\rho)$ is equal to the angle of the S_ϵ lying within S_j , so that it is equal to 2π if the projection of the observation point is inside the triangle, and is equal to zero if it is outside the triangle. It can also take values between zero and 2π if the projection is on an edge or a vertex. The expression for $\alpha(\rho)$ given in Equation (3.29) can be used to rewrite Equation (4.17) as

$$\int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z \frac{1}{R^3} = \hat{\mathbf{z}} \frac{|z|}{z} \sum_{i=1}^3 \hat{\mathbf{P}}_i^0 \cdot \hat{\mathbf{u}}_i \beta_i, \quad (4.18)$$

where

$$\beta_i = \tan^{-1} \frac{P_i^0 l_i^+}{(R_i^0)^2 + |z| R_i^+} - \tan^{-1} \frac{P_i^0 l_i^-}{(R_i^0)^2 + |z| R_i^+} \quad (4.19)$$

with the aid of the identity given in Equation (3.31).

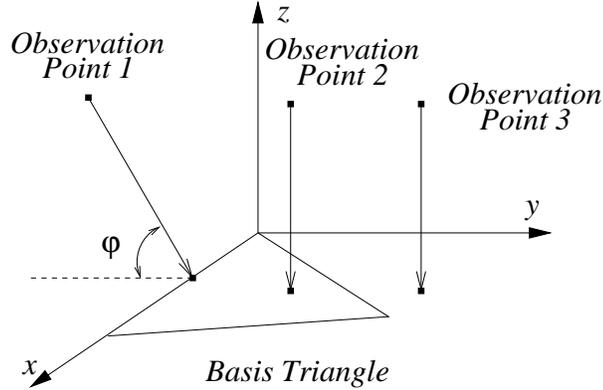


Figure 4.1: Limit cases for observation points approaching the basis triangle.

It is essential to check the analytic evaluation in the limit case, when the observation point approaches the x - y plane as shown in Figure 4.1. If the limit point

is outside or inside the triangle, but not on any edge, then

$$\lim_{z \rightarrow 0} \int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z \frac{1}{R^3} = \hat{\mathbf{z}} \sum_{i=1}^3 \hat{P}_i^0 \cdot \hat{u}_i \left(\tan^{-1} \frac{l_i^+}{P_i^0} - \tan^{-1} \frac{l_i^-}{P_i^0} \right) = \hat{\mathbf{z}} \alpha(\rho). \quad (4.20)$$

However, if the observation point approaches an edge, the contribution from that edge to the summation in Equation (4.18) becomes

$$\begin{aligned} & \lim_{R_1^0 \rightarrow 0} \left[\hat{P}_1^0 \cdot \hat{u}_1 \left(\tan^{-1} \frac{P_1^0 l_1^+}{(R_1^0)^2 + |z| R_1^+} - \tan^{-1} \frac{P_1^0 l_1^-}{(R_1^0)^2 + |z| R_1^+} \right) \right] \\ &= - \lim_{R_1^0 \rightarrow 0} \left[\tan^{-1} \frac{R_1^0 \cos(\varphi) l_1^+}{(R_1^0)^2 + |R_1^0 \sin(\varphi)| [(R_1^0)^2 + (l_1^+)^2]^{1/2}} \right. \\ & \quad \left. - \tan^{-1} \frac{R_1^0 \cos(\varphi) l_1^-}{(R_1^0)^2 + |R_1^0 \sin(\varphi)| [(R_1^0)^2 + (l_1^-)^2]^{1/2}} \right] \\ &= - \lim_{R_1^0 \rightarrow 0} \left[\tan^{-1} \frac{\cos(\varphi) l_1^+}{\sin(\varphi) |l_1^+|} - \tan^{-1} \frac{\cos(\varphi) l_1^-}{\sin(\varphi) |l_1^-|} \right] \\ &= 2\varphi - \pi, \end{aligned} \quad (4.21)$$

where φ is the angle between the approach path and the x - y plane. Contribution from the other two edges can also be evaluated as

$$\begin{aligned} & \lim_{R_1^0 \rightarrow 0} \left[\sum_{i=2}^3 \hat{P}_i^0 \cdot \hat{u}_i \left(\tan^{-1} \frac{P_i^0 l_i^+}{(R_i^0)^2 + |z| R_i^+} - \tan^{-1} \frac{P_i^0 l_i^-}{(R_i^0)^2 + |z| R_i^+} \right) \right] \\ &= \sum_{i=2}^3 \hat{P}_i^0 \cdot \hat{u}_i \left(\tan^{-1} \frac{l_i^+}{P_i^0} - \tan^{-1} \frac{l_i^-}{P_i^0} \right) = \pi \end{aligned} \quad (4.22)$$

and the value of the integral becomes

$$\lim_{R_1^0 \rightarrow 0} \int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z \frac{1}{R^3} = \hat{\mathbf{z}} (2\varphi - \pi + \pi) = \hat{\mathbf{z}} 2\varphi. \quad (4.23)$$

This result is interesting because the limit value of the integral appears to depend on the angle between the approach path and the plane of the basis triangle. Figure 4.3 shows the calculated values for the magnitude of the integral when the observation point approaches the edge with various angles as illustrated in Figure 4.2. The result in Equation (4.23) is validated by demonstrating that the calculated values converge to the correct limit values as $R_1^0 \rightarrow 0$.

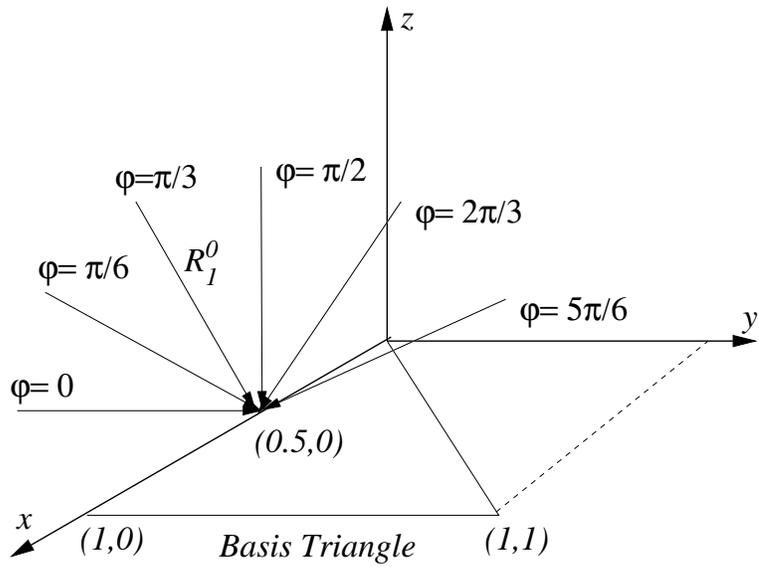


Figure 4.2: Observation points approaching the edge at different angles.

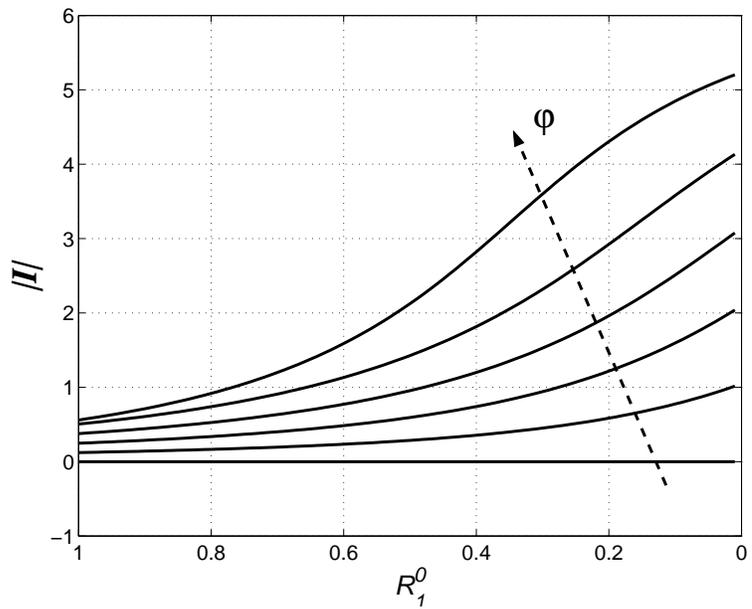


Figure 4.3: Magnitude of the integral $\mathbf{I} = \int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z f_{m3}(R)$ for the scenario in Figure 4.2.

Evaluation of $\int_{S_j} d\mathbf{r}'(\boldsymbol{\rho}' - \boldsymbol{\rho})f_{m3}(R)$

Considering Figure 3.3 again,

$$\int_{S_j} d\mathbf{r}' \frac{\boldsymbol{\rho}' - \boldsymbol{\rho}}{R^3} = \left[\int_{S_j - S_\varepsilon} d\mathbf{r}' \frac{\boldsymbol{\rho}' - \boldsymbol{\rho}}{R^3} + \int_{S_\varepsilon} d\mathbf{r}' \frac{\boldsymbol{\rho}' - \boldsymbol{\rho}}{R^3} \right] \quad (4.24)$$

and the identity

$$\nabla'_S \left(\frac{1}{R} \right) = -\frac{\boldsymbol{\rho}' - \boldsymbol{\rho}}{R^3} \quad (4.25)$$

can be used to continue as

$$\begin{aligned} \int_{S_j} d\mathbf{r}' \frac{\boldsymbol{\rho}' - \boldsymbol{\rho}}{R^3} &= -\lim_{\varepsilon \rightarrow 0} \int_{S_j - S_\varepsilon} d\mathbf{r}' \nabla'_S \left(\frac{1}{R} \right) + \lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} d\mathbf{r}' \frac{(\boldsymbol{\rho}' - \boldsymbol{\rho})}{R^3} \\ &= -\lim_{\varepsilon \rightarrow 0} \int_{\partial(S_j - S_\varepsilon)} \frac{1}{R} \hat{u} dl' + \lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} d\mathbf{r}' \frac{\hat{P}P}{R^3} \\ &= -\sum_{i=1}^3 \hat{u}_i \int_{\partial_i S} \frac{1}{R} dl' - \lim_{\varepsilon \rightarrow 0} \int_{\partial S_\varepsilon} \frac{1}{R} \hat{u} dl' \\ &\quad + \lim_{\varepsilon \rightarrow 0} \int_0^{2\pi} \int_0^\varepsilon \hat{P} \frac{P^2}{(d^2 + P^2)^{3/2}} dP d\phi \\ &= -\sum_{i=1}^3 \hat{u}_i \ln \frac{R_i^+ + l_i^+}{R_i^- + l_i^-}. \end{aligned} \quad (4.26)$$

It should be noted that the logarithmic function in Equation (4.26) diverges when the observation point is on one of the edges of the triangle. In other words, the analytic evaluation of an inner integral results in unbounded values for some locations of the observation point. If the integration rule that is applied on the testing triangle requires taking points on the edges, then an additional singularity extraction has to be applied to some of the outer integrals for the interaction of neighboring triangles, in order to avoid numerical problems.

4.1.3 Evaluation of the Impedance-Matrix Elements

Any interaction between a basis and a testing triangle can be evaluated by using Equation (4.10) with the aid of singularity extraction introduced in the previous subsection. The first integral in this equation is nonzero only for the self-interaction of the triangles, while the second integral contributes to all interactions. However, there exists a shorter way of evaluating the self-interactions. In addition, the interactions are not calculated as shown in Equation (4.10), but the integrals are divided into basic integrals for an efficient implementation. This subsection goes deeper into the evaluation of the integrals from the point of view of the implementation side.

Evaluation of the Self-Interactions of the Triangles

In the self-interaction of the triangles, all of the observation points are in the basis triangle. Therefore, Figure 4.1 can be considered again for the observation point that approaches a point inside the triangle on the x - y plane. Then, the integrand of the second integral in Equation (4.10) can be written in the limit form as

$$\begin{aligned}
& \lim_{z \rightarrow 0} \left\{ (\mathbf{r} - \boldsymbol{\rho}_{ik}) \cdot \hat{\mathbf{n}} \times (\mathbf{r} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' (\boldsymbol{\rho}' - \mathbf{r}) \frac{(1 - ikR)e^{ikR}}{R^3} \right\} \\
& = (\boldsymbol{\rho} - \boldsymbol{\rho}_{ik}) \cdot \hat{\mathbf{n}} \times (\boldsymbol{\rho} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' (\boldsymbol{\rho}' - \boldsymbol{\rho}) \frac{(1 - ikP)e^{ikP}}{P^3} \\
& \quad - \lim_{z \rightarrow 0} \left\{ (\mathbf{r} - \boldsymbol{\rho}_{ik}) \cdot \hat{\mathbf{n}} \times (\mathbf{r} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z \frac{(1 - ikR)e^{ikR}}{R^3} \right\}.
\end{aligned} \tag{4.27}$$

The first part of the integrand evaluates to zero due to the cross product between the two vectors in the x - y plane. This cross product gives a vector in the z direction, and the second cross product with $\hat{\mathbf{n}}$ vanishes since $\hat{\mathbf{n}} = \hat{\mathbf{z}}$ in the case of self-interaction. Although there exists a logarithmic singularity due to the $(\boldsymbol{\rho} - \boldsymbol{\rho}')/P^3$ term in Equation (4.26), this causes no problem since it is multiplied

by the vanishing cross product. Then,

$$\begin{aligned}
& \lim_{z \rightarrow 0} \left\{ (\mathbf{r} - \boldsymbol{\rho}_{ik}) \cdot \hat{\mathbf{n}} \times (\mathbf{r} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' (\boldsymbol{\rho}' - \mathbf{r}) \frac{(1 - ikR)e^{ikR}}{R^3} \right\} \\
&= - \lim_{z \rightarrow 0} \left\{ (\mathbf{r} - \boldsymbol{\rho}_{ik}) \cdot \hat{\mathbf{n}} \times (\mathbf{r} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z \frac{(1 - ikR)e^{ikR}}{R^3} \right\} \\
&= - \lim_{z \rightarrow 0} \left\{ (\mathbf{r} - \boldsymbol{\rho}_{ik}) \cdot \hat{\mathbf{n}} \times (\mathbf{r} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z f_{m1}(R) \right\} \\
&\quad - \lim_{z \rightarrow 0} \left\{ (\mathbf{r} - \boldsymbol{\rho}_{ik}) \cdot \hat{\mathbf{n}} \times (\mathbf{r} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z f_{m2}(R) \right\} \\
&\quad - \lim_{z \rightarrow 0} \left\{ (\mathbf{r} - \boldsymbol{\rho}_{ik}) \cdot \hat{\mathbf{n}} \times (\mathbf{r} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z f_{m3}(R) \right\}, \tag{4.28}
\end{aligned}$$

where $f_{m1}(R)$, $f_{m2}(R)$, and $f_{m3}(R)$ are given in Equation (4.12). $f_{m1}(R)$ has been shown to have finite values in all cases so that the integral of $z f_{m1}(R)$ goes to zero as z goes to zero. Furthermore, Equation (3.26) shows that the integral of $z f_{m2}(R)$ is also zero in the limit case, while Equation (4.20) shows that the integral of $z f_{m3}(R)$ converges to $\alpha(\rho)$, which is equal to 2π in the basis triangle. Therefore,

$$\begin{aligned}
& \lim_{z \rightarrow 0} \left\{ (\mathbf{r} - \boldsymbol{\rho}_{ik}) \cdot \hat{\mathbf{n}} \times (\mathbf{r} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' (\boldsymbol{\rho}' - \mathbf{r}) \frac{(1 - ikR)e^{ikR}}{R^3} \right\} \\
&= - \lim_{z \rightarrow 0} \left\{ (\boldsymbol{\rho} - \boldsymbol{\rho}_{ik}) \cdot \hat{\mathbf{n}} \times (\boldsymbol{\rho} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z f_{m3}(R) \right\} \\
&= - (\boldsymbol{\rho} - \boldsymbol{\rho}_{ik}) \cdot \hat{\mathbf{n}} \times (\boldsymbol{\rho} - \boldsymbol{\rho}_{jl}) \times \hat{\mathbf{z}} \alpha(\rho) \\
&= -2\pi (\boldsymbol{\rho} - \boldsymbol{\rho}_{ik}) \cdot (\boldsymbol{\rho} - \boldsymbol{\rho}_{jl}). \tag{4.29}
\end{aligned}$$

This final result can be substituted in Equation (4.10) and the expression for the self-interaction of a triangle becomes

$$\begin{aligned}
Z_{ik,il} &= \frac{l_{ik}l_{il}}{4A_i^2} \left[\int_{S_i} d\mathbf{r} (\boldsymbol{\rho} - \boldsymbol{\rho}_{ik}) \cdot (\boldsymbol{\rho} - \boldsymbol{\rho}_{il}) - \frac{1}{2} \int_{S_i} d\mathbf{r} (\boldsymbol{\rho} - \boldsymbol{\rho}_{ik}) \cdot (\boldsymbol{\rho} - \boldsymbol{\rho}_{jl}) \right] \\
&= \frac{l_{ik}l_{il}}{8A_i^2} \int_{S_i} d\mathbf{r} (\boldsymbol{\rho} - \boldsymbol{\rho}_{ik}) \cdot (\boldsymbol{\rho} - \boldsymbol{\rho}_{il}). \tag{4.30}
\end{aligned}$$

As a result, the evaluation of self-interactions reduces to a very simple form, which does not require the explicit implementation of singularity extraction. This result is expected from the discussion in Chapter 2. Since the current density is on a

planar surface, magnetic field produced by an infinitesimal portion of the current is perpendicular to the surface, except at the source location, where its magnitude is equal to half of the current density and its direction is tangential to the plane and perpendicular to the current flow. Then, testing of the radiation integral reduces to testing of the limit value in Equation (2.10), while the principle value is omitted.

For efficiency, Equation (4.30) is partitioned as

$$Z_{ik,il} = \frac{l_{ik}l_{il}}{4A_i^2} \left\{ \int_{S_i} d\mathbf{r}x^2 + \int_{S_i} d\mathbf{r}y^2 - (x_{il} + x_{ik}) \int_{S_i} d\mathbf{r}x \right. \\ \left. - (y_{il} + y_{ik}) \int_{S_i} d\mathbf{r}y + (x_{il}x_{ik} + y_{il}y_{ik})A_i \right\}. \quad (4.31)$$

This new expression indicates that the integrals related to self-interaction of a triangle can be evaluated without any alignment information about the testing and basis functions. Then, the alignment information can be used to evaluate nine different values to be added to different locations in the impedance matrix.

Evaluation of the Interactions Between Separate Triangles

For the interaction of separate triangles, Equation (4.10) becomes

$$Z_{ik,jl} = \frac{l_{ik}l_{jl}}{16\pi A_i A_j} \int_{S_i} d\mathbf{r}(\mathbf{r} - \mathbf{r}_{ik}) \cdot \hat{\mathbf{n}} \times (\mathbf{r} - \boldsymbol{\rho}_{jl}) \\ \times \int_{S_j} d\mathbf{r}'(\boldsymbol{\rho}' - \mathbf{r}) \frac{(1 - ikR)e^{ikR}}{R^3}. \quad (4.32)$$

The above is partitioned for an efficient implementation as

$$Z_{ik,jl} = \frac{l_{ik}l_{jl}}{16\pi A_i A_j} \left\{ \left(n_z x_{ik} x_{jl} + n_z y_{ik} y_{jl} - n_x x_{jl} z_{ik} - n_y y_{jl} z_{ik} \right) I_{m1} \right. \\ \left. + \left(n_y x_{ik} y_{jl} - n_x y_{jl} y_{ik} \right) I_{m2} + \left(n_x y_{ik} x_{jl} - n_y x_{jl} x_{ik} \right) I_{m3} \right. \\ \left. + \left(n_x z_{ik} - n_z x_{jl} - n_z x_{ik} \right) I_{m4} + \left(n_y z_{ik} - n_z y_{jl} - n_z y_{ik} \right) I_{m5} \right\} \quad (4.33)$$

$$\begin{aligned}
& + \left[n_x(y_{jl} + y_{ik}) - n_y x_{ik} \right] I_{m8} + \left[n_y(x_{jl} + x_{ik}) - n_x y_{ik} \right] I_{m10} \\
& + \left(n_x z_{ik} - n_z x_{ik} \right) I_{m9} + \left(n_y z_{ik} - n_z y_{ik} \right) I_{m12} \\
& + \left(n_x x_{jl} + n_y y_{jl} \right) I_{m6} - n_y y_{jl} I_{m7} - n_x x_{jl} I_{m11} \\
& + n_z I_{m13} - n_x I_{m14} + n_z I_{m15} - n_y I_{m16} + n_y I_{m17} + n_z I_{m18} \\
& - n_x I_{m19} - n_x I_{m20} - n_y I_{m21} + n_x I_{m22} + n_z I_{m23} - n_y I_{m24} \left. \vphantom{\int} \right\}, \tag{4.34}
\end{aligned}$$

where n_x , n_y and n_z are the x , y and z components of $\hat{\mathbf{n}}$. The basic integrals in Equation (4.33) can be listed as

$$\begin{aligned}
I_{m1} &= \int_{S_i} d\mathbf{r} I_{m1}^{in} & I_{m2} &= \int_{S_i} d\mathbf{r} I_{m2}^{in} & I_{m3} &= \int_{S_i} d\mathbf{r} I_{m3}^{in} \\
I_{m4} &= \int_{S_i} d\mathbf{r} x I_{m1}^{in} & I_{m5} &= \int_{S_i} d\mathbf{r} y I_{m1}^{in} & I_{m6} &= \int_{S_i} d\mathbf{r} z I_{m1}^{in} \\
I_{m7} &= \int_{S_i} d\mathbf{r} x I_{m2}^{in} & I_{m8} &= \int_{S_i} d\mathbf{r} y I_{m2}^{in} & I_{m9} &= \int_{S_i} d\mathbf{r} z I_{m2}^{in} \\
I_{m10} &= \int_{S_i} d\mathbf{r} x I_{m3}^{in} & I_{m11} &= \int_{S_i} d\mathbf{r} y I_{m3}^{in} & I_{m12} &= \int_{S_i} d\mathbf{r} z I_{m3}^{in} \\
I_{m13} &= \int_{S_i} d\mathbf{r} x^2 I_{m1}^{in} & I_{m14} &= \int_{S_i} d\mathbf{r} x z I_{m1}^{in} & I_{m15} &= \int_{S_i} d\mathbf{r} y^2 I_{m1}^{in} \\
I_{m16} &= \int_{S_i} d\mathbf{r} y z I_{m1}^{in} & I_{m17} &= \int_{S_i} d\mathbf{r} x y I_{m2}^{in} & I_{m18} &= \int_{S_i} d\mathbf{r} x z I_{m2}^{in} \\
I_{m19} &= \int_{S_i} d\mathbf{r} y^2 I_{m2}^{in} & I_{m20} &= \int_{S_i} d\mathbf{r} z^2 I_{m2}^{in} & I_{m21} &= \int_{S_i} d\mathbf{r} x^2 I_{m3}^{in} \\
I_{m22} &= \int_{S_i} d\mathbf{r} x y I_{m3}^{in} & I_{m23} &= \int_{S_i} d\mathbf{r} y z I_{m3}^{in} & I_{m24} &= \int_{S_i} d\mathbf{r} z^2 I_{m3}^{in}
\end{aligned} \tag{4.35}$$

and the inner integrals can be defined as

$$\begin{aligned}
I_{m1}^{in} &= \int_{S_j} d\mathbf{r}' z f_m(R) \\
I_{m2}^{in} &= \int_{S_j} d\mathbf{r}' (x' - x) f_m(R) \\
I_{m3}^{in} &= \int_{S_j} d\mathbf{r}' (y' - y) f_m(R), \tag{4.36}
\end{aligned}$$

where $f_m(R)$ is given in Equation (4.12). The twenty-four basic integrals in Equation (4.35) can be evaluated without alignment information about the testing and basis functions. Then, Equation (4.33) can be used for nine times to calculate the contributions to the impedance-matrix elements. It should be also noted that the basic integrals contain only three different inner integrals that can be evaluated only once.

The interaction between the separate triangles that are on the same plane can be entirely omitted in the MFIE implementations. This is because of the fact that the current source on the basis triangle produces only perpendicular magnetic field on the testing triangle. Therefore, these interactions can be neglected without any attempt of evaluation.

Evaluation of the Interactions Between the Neighboring Triangles

Calculation of the interactions between the testing and basis triangles, which share common edges, can be considered as the interactions of separate triangles. However, analytical evaluation of the $(\rho - \rho')/R^3$ integral given in Equation (4.26) results in unbounded values as the observation point approaches the common edge, and this singularity is not suppressed by a vanishing factor as in the case of self-interactions. Since the singularity is logarithmic, it does not cause a serious problem in the numerical integration on the testing triangle as long as the integration points are taken inside the triangle. However, it is also possible to apply a singularity-extraction technique for the outer integrals as will be shown in the next subsection.

Figure 4.4 shows the testing and basis triangles that are sharing a common edge. The observation point sampled on the testing triangle approaches the common edge at an angle φ with the x - y plane. The integrand of the integral in Equation

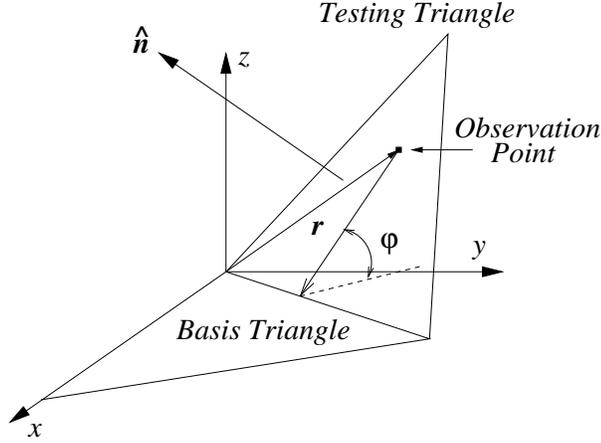


Figure 4.4: Interaction between the neighboring basis and testing triangles.

(4.10) can be considered in the limit case as

$$\begin{aligned}
& \lim_{R_1^0 \rightarrow 0} \left\{ (\mathbf{r} - \mathbf{r}_{ik}) \cdot \hat{\mathbf{n}} \times (\mathbf{r} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' (\boldsymbol{\rho}' - \mathbf{r}) \frac{(1 - ikR)e^{ikR}}{R^3} \right\} \\
&= (\boldsymbol{\rho} - \mathbf{r}_{ik}) \cdot \hat{\mathbf{n}} \times (\boldsymbol{\rho} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' (\boldsymbol{\rho}' - \boldsymbol{\rho}) \frac{(1 - ikP)e^{ikP}}{P^3} \\
&- \lim_{R_1^0 \rightarrow 0} \left\{ (\mathbf{r} - \mathbf{r}_{ik}) \cdot \hat{\mathbf{n}} \times (\mathbf{r} - \boldsymbol{\rho}_{jl}) \times \left[\int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z f_{m1}(R) \right. \right. \\
&+ \left. \left. \frac{k^2}{2} \int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z f_{m2}(R) + \int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z f_{m3}(R) \right] \right\} \\
&= (\boldsymbol{\rho} - \mathbf{r}_{ik}) \cdot \hat{\mathbf{n}} \times (\boldsymbol{\rho} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' (\boldsymbol{\rho}' - \boldsymbol{\rho}) \frac{(1 - ikR)e^{ikR}}{R^3} \\
&- \lim_{R_1^0 \rightarrow 0} \left\{ (\mathbf{r} - \mathbf{r}_{ik}) \cdot \hat{\mathbf{n}} \times (\mathbf{r} - \boldsymbol{\rho}_{jl}) \times \left[\int_{S_j} d\mathbf{r}' \hat{\mathbf{z}} z f_{m3}(R) \right] \right\}. \quad (4.37)
\end{aligned}$$

It should be noted that $\hat{\mathbf{n}}$ is not in the z direction as in the case of self-interaction. Therefore, the cross products in the first term does not evaluate to zero and the logarithmic singularity is not suppressed. On the other hand, integrals of $z f_{m1}(R)$ and $z f_{m2}(R)$ become zero again, while integral of $z f_{m3}(R)$ has the limit

value given in Equation (4.23). Using this value, the integrand becomes

$$\begin{aligned}
& \lim_{R_1^0 \rightarrow 0} \left\{ (\mathbf{r} - \mathbf{r}_{ik}) \cdot \hat{\mathbf{n}} \times (\mathbf{r} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' (\boldsymbol{\rho}' - \mathbf{r}) \frac{(1 - ikR)e^{ikR}}{R^3} \right\} \\
&= (\boldsymbol{\rho} - \mathbf{r}_{ik}) \cdot \hat{\mathbf{n}} \times (\boldsymbol{\rho} - \boldsymbol{\rho}_{jl}) \times \int_{S_j} d\mathbf{r}' (\boldsymbol{\rho}' - \boldsymbol{\rho}) \frac{(1 - ikR)e^{ikR}}{R^3} \\
&\quad - (\boldsymbol{\rho} - \mathbf{r}_{ik}) \cdot \hat{\mathbf{n}} \times (\boldsymbol{\rho} - \boldsymbol{\rho}_{jl}) \times \hat{\mathbf{z}} 2\varphi.
\end{aligned} \tag{4.38}$$

The second term of the right-hand side of Equation (4.38) represents the testing operation of the magnetic field produced by the infinitesimal portion of the basis triangle around the limit location of the observation point, while the first term represents the testing operation of the magnetic field produced by the rest of the basis triangle.

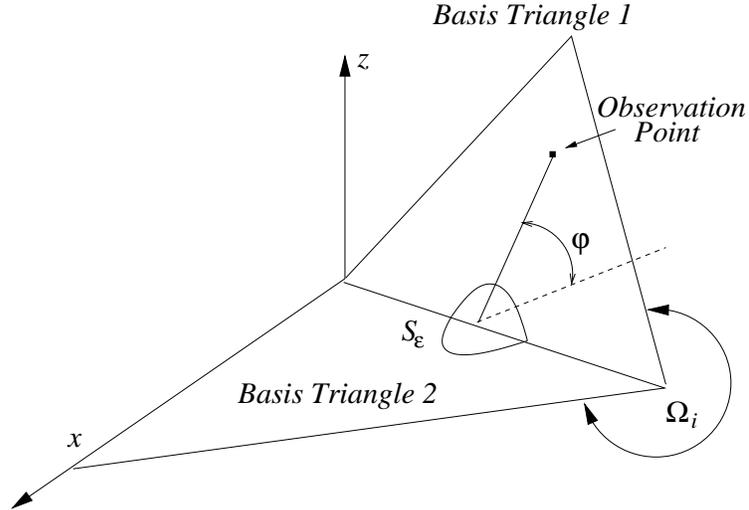


Figure 4.5: Limit case as the observation point approaches the edge of the basis function.

Finally, it is possible to determine the magnetic field produced by an infinitesimal source on the same source location residing on an edge. Figure 4.5 shows a basis function located on an edge, where the model geometry has an internal solid angle of Ω_i . The observation point is in one of the triangles and approaches the edge. The infinitesimal portion of the basis function located around the limit

point is non planar and has two sides on the both triangles. The second term in Equation (4.38) can be used to write the contribution from the side in the second triangle as

$$H_{\varepsilon \rightarrow 0}^{sca,2}(\mathbf{r}) = \frac{2\varphi}{4\pi}(\boldsymbol{\rho} - \boldsymbol{\rho}_{j_2 l_2}) \frac{l_{j_2 l_2}}{A_{j_2}} \times \hat{\mathbf{n}}_2 = \frac{\varphi}{2\pi} \mathbf{J}_2(r) \times \hat{\mathbf{n}}_2 \quad (4.39)$$

with the aid of the constants in Equation (4.10), while omitting the constants related to the testing function. Similarly, the contribution from the side in the first triangle can be written with the aid of Equation (4.29) as

$$H_{\varepsilon \rightarrow 0}^{sca,1}(\mathbf{r}) = \frac{2\pi}{4\pi}(\boldsymbol{\rho} - \boldsymbol{\rho}_{j_1 l_1}) \frac{l_{j_1 l_1}}{A_{j_1}} \times \hat{\mathbf{n}}_1 = \frac{\pi}{2\pi} \mathbf{J}_1(r) \times \hat{\mathbf{n}}_1. \quad (4.40)$$

Then, the total field becomes

$$H_{\varepsilon \rightarrow 0}^{sca}(\mathbf{r}) = \frac{\pi}{2\pi} \mathbf{J}_1(r) \times \hat{\mathbf{n}}_1 + \frac{\varphi}{2\pi} \mathbf{J}_2(r) \times \hat{\mathbf{n}}_2. \quad (4.41)$$

This final expression is different from the one in Equation (2.19) since the current density at the limit location of the observation point is not continuous in this case.

4.1.4 Singularity Extraction for Outer Integrals

Among the integrands of the basic integrals shown in Equation (4.35), those that include $I_{m_2}^{in}$ and $I_{m_3}^{in}$ may have logarithmic singularities related to their f_{m_3} term given in Equation (4.26) if they are calculated for the interactions of neighboring triangles. Some of these integrands also include z factor, which goes to zero as the observation point approaches the common edge, and the singularity is suppressed by this vanishing factor. One way to annihilate the singularity of the other integrands is to apply a second coordinate transformation so that the common edge always lies on the x axis, as shown in Figure 4.6.

With this rotation, the integrands including the $I_{m_2}^{in}$ term become bounded and the singularity is suppressed since the contribution of the common edge to the

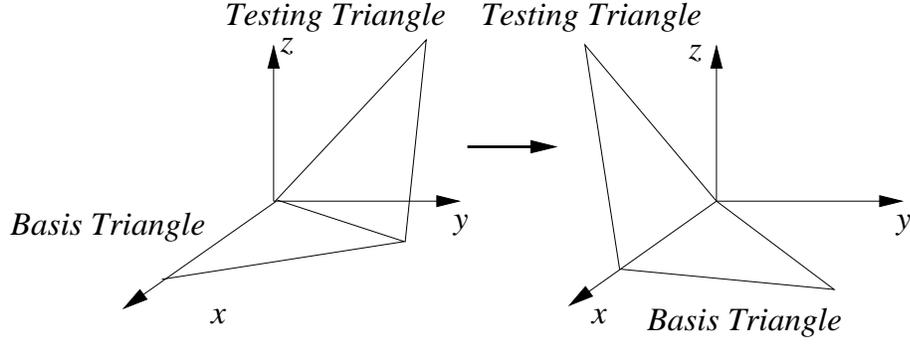


Figure 4.6: Second coordinate transformation used to annihilate the singularity of the outer integral.

summation in Equation (4.26) becomes zero. It should be recalled that only this contribution leads to unbounded values, while the contributions from the other two edges are bounded. Considering the integrands including the I_{m3}^{in} term, it can be observed that some of them include a y factor, which also becomes zero as the observation point approaches the common edge in the rotated configuration. Then, the singularities in these integrands also disappear.

Finally, there are only three basic integrals

$$\begin{aligned}
 I_{m3} &= \int_{S_i} d\mathbf{r} \int_{S_j} d\mathbf{r}' (y' - y) f_m(R) \\
 I_{m10} &= \int_{S_i} d\mathbf{r} x \int_{S_j} d\mathbf{r}' (y' - y) f_m(R) \\
 I_{m21} &= \int_{S_i} d\mathbf{r} x^2 \int_{S_j} d\mathbf{r}' (y' - y) f_m(R),
 \end{aligned} \tag{4.42}$$

which still have logarithmic singularities in their integrands. Since the logarithmic singularity is weak, numerical evaluation of these integrals can be achieved by dense sampling in the testing domain and avoiding approaching the common edge. Alternatively, singularity extraction for the outer integrals can also be employed, as will be shown here.

Using Equations (4.12) and (4.26),

$$\begin{aligned}
I_{m3} &= \int_{S_i} d\mathbf{r} \int_{S_j} d\mathbf{r}' (y' - y) \left[f_{m1}(R) + \frac{k^2}{2} f_{m2}(R) \right] \\
&\quad + \int_{S_i} d\mathbf{r} \int_{S_j} d\mathbf{r}' (y' - y) f_{m3}(R) \\
&= \int_{S_i} d\mathbf{r} \int_{S_j} d\mathbf{r}' (y' - y) \left[f_{m1}(R) + \frac{k^2}{2} f_{m2}(R) \right] \\
&\quad - \int_{S_i} d\mathbf{r} \sum_{i=1}^3 \hat{u}_i \ln \frac{R_i^+ + l_i^+}{R_i^- + l_i^-}
\end{aligned} \tag{4.43}$$

and the singularity is only due to the contribution from the common edge, so that

$$\begin{aligned}
I_{m3} &= \int_{S_i} d\mathbf{r} \int_{S_j} d\mathbf{r}' (y' - y) \left[f_{m1}(R) + \frac{k^2}{2} f_{m2}(R) \right] \\
&\quad - \int_{S_i} d\mathbf{r} \sum_{i=2}^3 \hat{u}_i \ln \frac{R_i^+ + l_i^+}{R_i^- + l_i^-} + \int_{S_i} d\mathbf{r} \left[\ln \frac{R_1^+ + l_1^+}{R_1^- + l_1^-} + 2 \ln(R_1^0) \right] \\
&\quad - \int_{S_i} d\mathbf{r} 2 \ln(R_1^0).
\end{aligned} \tag{4.44}$$

As the observation point approaches the common edge,

$$\begin{aligned}
&\lim_{R_1^0 \rightarrow 0} \left\{ \ln \frac{R_1^+ + l_1^+}{R_1^- + l_1^-} + 2 \ln(R_1^0) \right\} \\
&= \lim_{R_1^0 \rightarrow 0} \left\{ \ln \left(\frac{[(R_1^0)^2 + (l_1^+)^2]^{1/2} + l_1^+}{[(R_1^0)^2 + (l_1^-)^2]^{1/2} + l_1^-} (R_1^0)^2 \right) \right\} \\
&= \ln \left[4l_1^+ |l_1^+| \right],
\end{aligned} \tag{4.45}$$

which is finite. Then,

$$\begin{aligned}
I_{m3} &= \int_{S_i} d\mathbf{r} \left[\int_{S_j} d\mathbf{r}' (y' - y) f_m(R) + 2 \ln(R_1^0) \right] \\
&\quad - \int_{S_i} d\mathbf{r} 2 \ln(R_1^0)
\end{aligned} \tag{4.46}$$

and the first integral can be evaluated numerically without any singularity problem, while the second integral can be evaluated analytically. Similarly,

$$I_{m10} = \int_{S_i} d\mathbf{r}x \left[\int_{S_j} d\mathbf{r}'(y' - y)f_m(R) + 2 \ln(R_1^0) \right] - \int_{S_i} d\mathbf{r}2x \ln(R_1^0) \quad (4.47)$$

$$I_{m21} = \int_{S_i} d\mathbf{r}x^2 \left[\int_{S_j} d\mathbf{r}'(y' - y)f_m(R) + 2 \ln(R_1^0) \right] - \int_{S_i} d\mathbf{r}2x^2 \ln(R_1^0). \quad (4.48)$$

Finally, the analytical integrals can be evaluated as

$$\begin{aligned} \int_{S_i} d\mathbf{r}2 \ln(R_1^0) &= A_i \left[\ln(h_i) - \frac{3}{2} \right] \\ \int_{S_i} d\mathbf{r}2x \ln(R_1^0) &= l_1 A_i \left[\frac{2}{3} \ln(h_i) - \frac{8}{9} \right] + l_2 A_i \left[\frac{1}{3} \ln(h_i) - \frac{11}{18} \right] \\ \int_{S_i} d\mathbf{r}2x^2 \ln(R_1^0) &= l_1^2 A_i \left[\frac{1}{2} \ln(h_i) - \frac{5}{8} \right] + l_2^2 A_i \left[\frac{1}{6} \ln(h_i) - \frac{25}{72} \right] \\ &\quad + l_1 l_2 A_i \left[-\frac{1}{2} \ln(h_i) + \frac{7}{8} \right]. \end{aligned} \quad (4.49)$$

4.1.5 Numerical Integration of Inner and Outer Integrals

The numerical evaluation of the inner integrals in the MFIE is very similar to the evaluation in the EFIE formulation. The inner integrals in (4.36) can be calculated in a parallel manner by using an adaptive algorithm. Then, these integrals are used in forming the outer integrals given in (4.35). It should be noted that the number of outer integrals is twenty-four, which is much larger than the number of outer integrals in the EFIE. These integrals are also evaluated simultaneously with high-ordered Gaussian quadrature rules or adaptive algorithms. On the other hand, three of the outer integrals require singularity extraction in the case of interactions between neighboring triangles. Thus, only the numerical parts of these integrals are evaluated in the same numerical routine with the other integrals, while their analytical parts are calculated separately.

4.1.6 Excitation

Plane-wave excitation can be used with the MFIE formulation as in the EFIE.

In the excitation formula

$$v_m^M = \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \hat{\mathbf{n}} \times \mathbf{H}^{inc}(\mathbf{r}), \quad (4.50)$$

the incident magnetic field can be written as

$$\mathbf{H}^{inc}(\mathbf{r}) = \frac{1}{\eta} \hat{\mathbf{k}} \times \mathbf{E}_0 e^{i\mathbf{k} \cdot \mathbf{r}}, \quad (4.51)$$

where \mathbf{E}_0 is a complex vector, which satisfies

$$\mathbf{E}_0 \cdot \mathbf{k} = 0. \quad (4.52)$$

Then,

$$v_m^M = \frac{1}{\eta} \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \hat{\mathbf{n}} \times \hat{\mathbf{k}} \times \mathbf{E}_0 e^{i\mathbf{k} \cdot \mathbf{r}} \quad (4.53)$$

can be used to fill the right-hand-side vector of the MOM system.

4.2 MFIE-MOM Implementation with Point Testing Functions

This section introduces the relatively simpler implementation of the MOM using the MFIE formulation. This implementation uses point testing functions defined at the center of each edge and in the vector direction perpendicular to the edges as shown in Figure 4.7. These functions are in the planes of the triangles so that, if the two triangles are not on the same plane, then the testing functions also split into two unaligned parts. Using these functions, the self-interactions in the MOM matrix reduces to a very simple form and it becomes possible to omit the singularity extraction entirely [19].

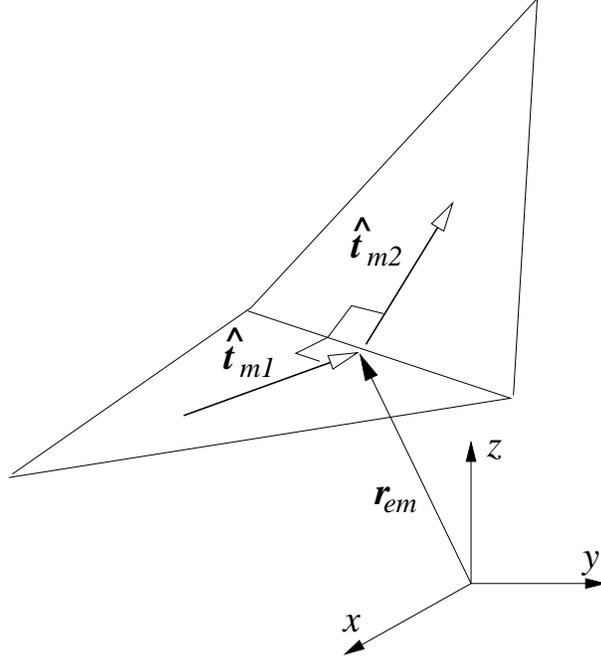


Figure 4.7: Point testing functions.

4.2.1 Formulation

Equation (4.3) shows the MOM system as

$$\sum_{n=1}^N a_n Z_{mn}^M = v_m, \quad (4.54)$$

where

$$Z_{mn}^M = \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \mathbf{b}_n(\mathbf{r}) - \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \hat{\mathbf{n}} \times \int_{S_n} d\mathbf{r}' \mathbf{b}_n(\mathbf{r}') \times \nabla' g(\mathbf{r}, \mathbf{r}') \quad (4.55)$$

and

$$v_m^M = \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \hat{\mathbf{n}} \times \mathbf{H}^{inc}(\mathbf{r}) \quad (4.56)$$

if the MFIE is used. The testing function defined on the m^{th} edge can be written as

$$\mathbf{t}_m(\mathbf{r}) = \frac{1}{2} \delta(\mathbf{r} - \mathbf{r}_{em}) \hat{\mathbf{t}}_{m1} + \frac{1}{2} \delta(\mathbf{r} - \mathbf{r}_{em}) \hat{\mathbf{t}}_{m2}, \quad (4.57)$$

where \mathbf{r}_{em} represents the center location of the m^{th} edge, while $\hat{\mathbf{t}}_{m1}$ and $\hat{\mathbf{t}}_{m2}$ represents the unit vectors, which are perpendicular to the m^{th} edge and lying in the planes of the triangles. Using this expression,

$$Z_{mn}^M = \left[\frac{1}{2}\hat{\mathbf{t}}_{m1} + \frac{1}{2}\hat{\mathbf{t}}_{m2} \right] \cdot \mathbf{b}_n(\mathbf{r}_{em}) - \hat{\mathbf{e}}_m \cdot \int_{S_n} d\mathbf{r}' \mathbf{b}_n(\mathbf{r}') \times \nabla' g(\mathbf{r}_{em}, \mathbf{r}') \quad (4.58)$$

and

$$v_m^M = \hat{\mathbf{e}}_m \cdot \mathbf{H}^{inc}(\mathbf{r}_{em}), \quad (4.59)$$

where $\hat{\mathbf{e}}_m$ is the unit vector in the direction of the m^{th} edge. Finally, if the RWG basis functions are used, the interaction between a basis triangle and a testing function (part of an impedance-matrix element) becomes

$$\begin{aligned} Z_{m,jl}^M &= \left[\frac{1}{2}\hat{\mathbf{t}}_{m1} + \frac{1}{2}\hat{\mathbf{t}}_{m2} \right] \cdot \delta_j(\mathbf{r}_{em}) \frac{l_{jl}(\mathbf{r}_{em} - \mathbf{r}_{jl})}{2A_j} \\ &\quad - \hat{\mathbf{e}}_m \cdot \frac{l_{jl}}{2A_j} \int_{S_n} d\mathbf{r}' (\mathbf{r}' - \mathbf{r}_{jl}) \times \nabla' g(\mathbf{r}_{em}, \mathbf{r}'), \end{aligned} \quad (4.60)$$

where the index j is used to indicate that the interaction is calculated for the j^{th} basis triangle and the index l represents the alignment of the basis function on this triangle. As it was done before, a rotation can be applied at this stage, so that the basis triangle resides on the x - y plane. Then,

$$\begin{aligned} Z_{m,jl}^M &= \left[\frac{1}{2}\hat{\mathbf{t}}_{m1} + \frac{1}{2}\hat{\mathbf{t}}_{m2} \right] \cdot \delta_j(\mathbf{r}_{em}) \frac{l_{jl}(\mathbf{r}_{em} - \boldsymbol{\rho}_{jl})}{2A_j} \\ &\quad - \hat{\mathbf{e}}_m \cdot \frac{l_{jl}}{2A_j} \int_{S_n} d\mathbf{r}' (\boldsymbol{\rho}' - \boldsymbol{\rho}_{jl}) \times (\mathbf{r}_{em} - \boldsymbol{\rho}') \frac{(1 - ikR_{em})e^{ikR_{em}}}{4\pi R_{em}^3} \\ &= \left[\frac{1}{2}\hat{\mathbf{t}}_{m1} + \frac{1}{2}\hat{\mathbf{t}}_{m2} \right] \cdot \delta_j(\mathbf{r}_{em}) \frac{l_{jl}(\mathbf{r}_{em} - \boldsymbol{\rho}_{jl})}{2A_j} \\ &\quad + \hat{\mathbf{e}}_m \cdot \frac{l_{jl}}{2A_j} (\mathbf{r}_{em} - \boldsymbol{\rho}_{jl}) \times \int_{S_n} d\mathbf{r}' (\boldsymbol{\rho}' - \mathbf{r}_{em}) \frac{(1 - ikR_{em})e^{ikR_{em}}}{4\pi R_{em}^3}, \end{aligned} \quad (4.61)$$

where

$$R_{em} = |\mathbf{r}_{em} - \mathbf{r}'| \quad (4.62)$$

and with the aid of the identity given in Equation (4.9).

4.2.2 Evaluation of the Impedance-Matrix Elements

Equation (4.61) can be used easily to calculate the interaction between a testing function and a basis triangle that are not coincident in space. For those interactions, the first term evaluates to zero. Because, $\delta_j(\mathbf{r}_{em})$ is zero since \mathbf{r}_{em} is located out of the basis triangle.

On the other hand, when the testing function is defined on one of the edges of the basis triangle, it becomes possible for the observation point to be at the same location with the source point. However, this causes numerical problems since the Green's function tends to diverge as R goes to zero. As a result, singularity extraction has to be applied to the integrand when the testing function resides on one of the edges of the basis triangle.

Evaluation of Self-Interactions

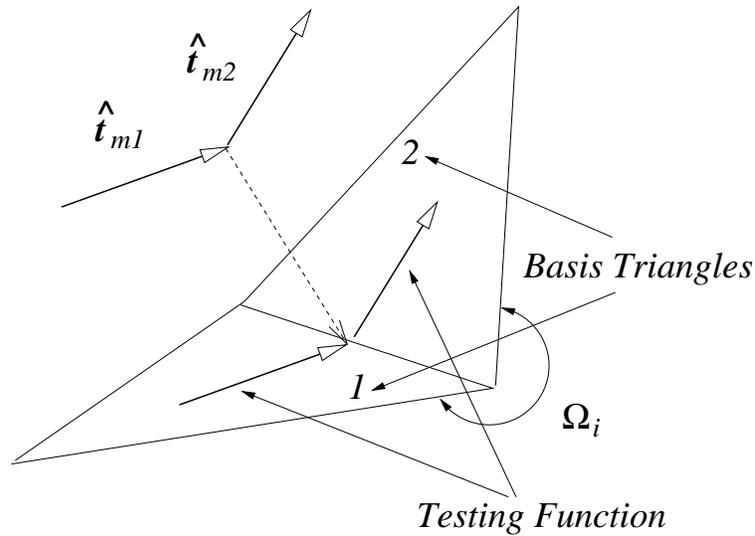


Figure 4.8: Point testing function approaching the center of the edge of the basis function from outside.

Figure 4.8 shows the testing function defined on the same edge as the basis function, and it is desired to calculate the interaction between these functions. It

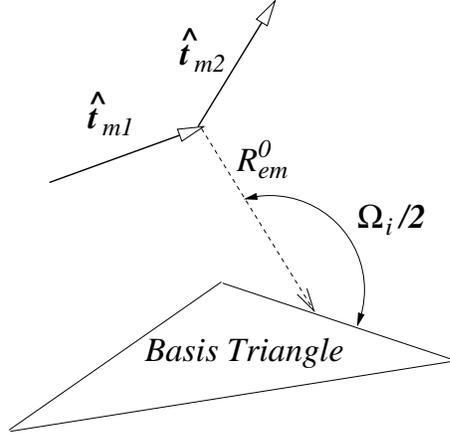


Figure 4.9: Point testing function approaching the edge of the basis triangle.

is assumed that the testing function approaches from the outside of the geometry. Considering only the first triangle of the basis function, the scenario simplifies to the one shown in Figure 4.9. The observation point approaches the edge on a line making an angle φ with the x - y plane and

$$\varphi = \frac{\Omega_i}{2}, \quad (4.63)$$

where Ω_i is the internal solid angle at the edge. Then,

$$\begin{aligned} Z_{m,j_1l_1}^M &= \frac{1}{2} \hat{\mathbf{t}}_{m1} \cdot \frac{l_{j_1l_1}(\boldsymbol{\rho}_{em} - \boldsymbol{\rho}_{j_1l_1})}{2A_{j_1}} \\ &+ \hat{\mathbf{e}}_m \cdot \frac{l_{j_1l_1}}{2A_{j_1}} (\boldsymbol{\rho}_{em} - \boldsymbol{\rho}_{j_1l_1}) \times \int_{S_n} d\mathbf{r}' (\boldsymbol{\rho}' - \boldsymbol{\rho}_{em}) \frac{(1 - ikR_{em})e^{ikR_{em}}}{4\pi R_{em}^3} \\ &- \lim_{R_{em}^0 \rightarrow 0} \left\{ \hat{\mathbf{e}}_m \cdot \frac{l_{j_1l_1}}{2A_{j_1}} (\mathbf{r}_{em} - \boldsymbol{\rho}_{j_1l_1}) \times \hat{\mathbf{z}} \int_{S_n} d\mathbf{r}' z_{em} \frac{(1 - ikR_{em})e^{ikR_{em}}}{4\pi R_{em}^3} \right\}. \end{aligned} \quad (4.64)$$

It should be noted that half of the testing function resides on the other side of the edge so that $\hat{\mathbf{t}}_{m2}$ has no contribution in the first term. It can be also observed that the second term in (4.64) evaluates to zero since the cross product of two vectors on the x - y plane gives a vector in the z direction, which is perpendicular to $\hat{\mathbf{e}}_m$. Although the integral contains a logarithmic singularity, it is suppressed by the

vanishing dot product. For the third term, only the integral of $z_{em}f_{m3}(R_{em})$ is nonzero as z_{em} goes to zero and

$$Z_{m,j_1l_1}^M = \frac{1}{2}\hat{\mathbf{t}}_{m1} \cdot \frac{l_{j_1l_1}(\boldsymbol{\rho}_{em} - \boldsymbol{\rho}_{j_1l_1})}{2A_{j_1}} - \hat{\mathbf{e}}_m \cdot \frac{l_{j_1l_1}(\boldsymbol{\rho}_{em} - \boldsymbol{\rho}_{j_1l_1})}{2A_{j_1}} \times \hat{\mathbf{z}} \frac{2\varphi}{4\pi} \quad (4.65)$$

with the aid of the limit value given in Equation (4.23). Finally, these two terms can be combined as

$$\begin{aligned} Z_{m,j_1l_1}^M &= \frac{1}{2}\hat{\mathbf{t}}_{m1} \cdot \frac{l_{j_1l_1}(\boldsymbol{\rho}_{em} - \boldsymbol{\rho}_{j_1l_1})}{2A_{j_1}} - \frac{\varphi}{2\pi}\hat{\mathbf{t}}_{m1} \cdot \frac{l_{j_1l_1}(\boldsymbol{\rho}_{em} - \boldsymbol{\rho}_{j_1l_1})}{2A_{j_1}} \\ &= \frac{1}{2}\left[1 - \frac{\Omega_i}{2\pi}\right]\hat{\mathbf{t}}_{m1} \cdot \frac{l_{j_1l_1}(\boldsymbol{\rho}_{em} - \boldsymbol{\rho}_{j_1l_1})}{2A_{j_1}} \\ &= \frac{1}{2}\left[1 - \frac{\Omega_i}{2\pi}\right]\hat{\mathbf{t}}_{m1} \cdot \mathbf{J}_1(\boldsymbol{\rho}_{em}) \\ &= \frac{1}{2}\left[1 - \frac{\Omega_i}{2\pi}\right] \end{aligned} \quad (4.66)$$

since the normal component of the RWG function is unity at the edge.

It can be observed that the interaction between the second basis triangle and the testing function is the same:

$$Z_{m,j_2l_2}^M = \frac{1}{2}\left[1 - \frac{\Omega_i}{2\pi}\right]\hat{\mathbf{t}}_{m2} \cdot \mathbf{J}_2(\boldsymbol{\rho}_{em}) = \frac{1}{2}\left[1 - \frac{\Omega_i}{2\pi}\right]. \quad (4.67)$$

Therefore, self-interactions in the impedance matrix become

$$Z_{mn}^M = \left[1 - \frac{\Omega_i}{2\pi}\right], \quad (4.68)$$

which is a very simple expression to be evaluated when $m = n$.

Evaluation of Neighboring Interactions

As it is shown in Figure 4.10, the testing function may reside on one of the edges of the triangle, but it may not be defined on the same edge as the basis function.

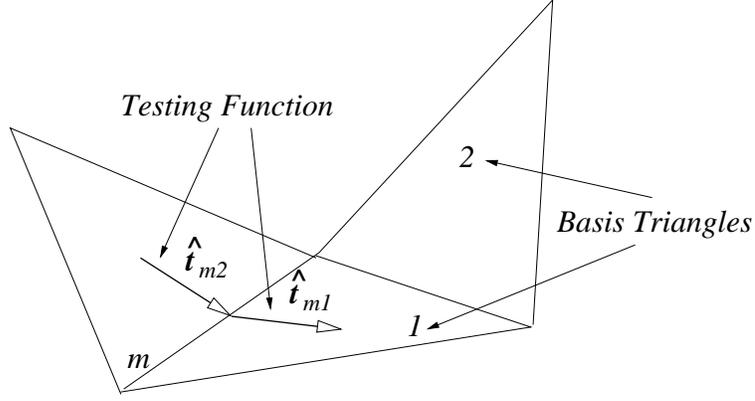


Figure 4.10: The testing function residing on one of the edges of the basis triangle, on which the basis function is not defined.

Then, using the final result in Equation (4.65), the interaction becomes

$$\begin{aligned}
Z_{m,j_1l_1}^M &= \frac{1}{2} \hat{\mathbf{t}}_{m1} \cdot \frac{l_{j_1l_1}(\boldsymbol{\rho}_{em} - \boldsymbol{\rho}_{j_1l_1})}{2A_{j_1}} - \frac{\varphi}{2\pi} \hat{\mathbf{t}}_{m1} \cdot \frac{l_{j_1l_1}(\boldsymbol{\rho}_{em} - \boldsymbol{\rho}_{j_1l_1})}{2A_{j_1}} \\
&= \frac{1}{2} \left[1 - \frac{\Omega_i}{2\pi} \right] \hat{\mathbf{t}}_{m1} \cdot \mathbf{J}_1(\boldsymbol{\rho}_{em}) \\
&= \frac{1}{2} \left[1 - \frac{\Omega_i}{2\pi} \right] \\
&= 0
\end{aligned} \tag{4.69}$$

since the current flow is parallel to the edge and perpendicular to the testing function. Consequently, the interactions between the neighboring testing functions and basis triangles can simply be omitted.

Evaluation of Interactions Between Nontouching Functions

When point testing functions are used in the direction perpendicular to the edges, evaluation of self-interactions in the MFIE does not require any singularity extraction. In addition, neighboring interactions evaluate to zero. Therefore, to simplify the implementation, singularity extraction can be totally omitted even for the near-distance interactions between nontouching functions.

The interactions between nontouching basis triangles and testing functions can be written as

$$\begin{aligned}
Z_{m,jl}^M &= \hat{\mathbf{e}}_m \cdot \frac{l_{jl}}{2A_j} (\mathbf{r}_{em} - \boldsymbol{\rho}_{jl}) \times \int_{S_n} d\mathbf{r}' (\boldsymbol{\rho}' - \mathbf{r}_{em}) \frac{(1 - ikR_{em})e^{ikR_{em}}}{4\pi R_{em}^3} \\
&= \hat{\mathbf{e}}_m \cdot \frac{l_{jl}}{2A_j} (\mathbf{r}_{em} - \boldsymbol{\rho}_{jl}) \times \int_{S_n} d\mathbf{r}' (\boldsymbol{\rho}' - \mathbf{r}_{em}) f_m(R_{em}). \tag{4.70}
\end{aligned}$$

For an efficient implementation, this expression is divided into smaller integrals as

$$\begin{aligned}
Z_{m,jl}^M &= \frac{l_{jl}}{2A_j} e_{mx} \left(y_{jl} z_{em} I_{mp1}^{in} - z_{em} I_{mp3}^{in} \right) \\
&\quad - \frac{l_{jl}}{2A_j} e_{my} \left(x_{jl} z_{em} I_{mp1}^{in} - z_{em} I_{mp2}^{in} \right) \\
&\quad + \frac{l_{jl}}{2A_j} e_{mz} \left[(x_{em} - x_{jl}) I_{mp3}^{in} - (y_{em} - y_{jl}) I_{mp2}^{in} \right. \\
&\quad \quad \left. + (x_{jl} y_{em} - y_{jl} x_{em}) I_{mp1}^{in} \right], \tag{4.71}
\end{aligned}$$

where e_{mx} , e_{my} , and e_{mz} are the x , y , and z components of $\hat{\mathbf{e}}_m$ respectively. The integrals are slightly different from the ones given in Equation (4.36) as

$$\begin{aligned}
I_{mp1}^{in} &= \int_{S_n} d\mathbf{r}' f_m(R_{em}) \\
I_{mp2}^{in} &= \int_{S_n} d\mathbf{r}' x' f_m(R_{em}) \\
I_{mp3}^{in} &= \int_{S_n} d\mathbf{r}' y' f_m(R_{em}) \tag{4.72}
\end{aligned}$$

and singularity extraction is not applied to the integrands.

4.3 CFIE-MOM Implementation with RWG Basis and Testing Functions

Combining Equations (3.15) and (4.6) with the aid of Equation (2.25),

$$\begin{aligned}
Z_{ik,jl}^C = & \alpha \frac{l_{ik}l_{jl}}{4A_iA_j} \left\{ \int_{S_i} d\mathbf{r}(\mathbf{r} - \mathbf{r}_{ik}) \cdot \int_{S_j} d\mathbf{r}'g(\mathbf{r}, \mathbf{r}')(\mathbf{r}' - \mathbf{r}_{jl}) \right. \\
& \left. - \frac{4}{k^2} \int_{S_m} d\mathbf{r} \int_{S_n} d\mathbf{r}'g(\mathbf{r}, \mathbf{r}') \right\} \\
& + (1 - \alpha) \frac{i}{k} \frac{l_{ik}l_{jl}}{4A_iA_j} \left\{ \delta_{ij} \int_{S_i} d\mathbf{r}(\mathbf{r} - \mathbf{r}_{ik}) \cdot (\mathbf{r} - \mathbf{r}_{jl}) \right. \\
& \left. - \int_{S_i} d\mathbf{r}(\mathbf{r} - \mathbf{r}_{ik}) \cdot \hat{\mathbf{n}} \times \int_{S_j} d\mathbf{r}'(\mathbf{r}' - \mathbf{r}_{jl}) \times \nabla'g(\mathbf{r}, \mathbf{r}') \right\} \quad (4.73)
\end{aligned}$$

can be used to calculate the interactions in the CFIE formulation employing RWG basis and testing functions. The MOM implementation on the CFIE formulation should not be constructed as the sequential combination of the EFIE and the MFIE parts. Although these formulations are generally different, their detailed implementations share some of the inner and outer integrals. In addition, some basis operations, such as the coordinate transformation, can be done in common to avoid duplications. Consequently, the MFIE and the EFIE implementations should be mixed to form an efficient CFIE implementation. For each triangle-triangle pair, the EFIE and the MFIE formulations should be applied simultaneously to calculate interaction in the CFIE.

Equations (4.53) and (3.47) can also be combined as

$$v_m = \alpha \frac{i}{k\eta} \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \mathbf{E}^0 e^{i\mathbf{k} \cdot \mathbf{r}} + (1 - \alpha) \frac{i}{k\eta} \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \hat{\mathbf{n}} \times \hat{\mathbf{k}} \times \mathbf{E}_0 e^{i\mathbf{k} \cdot \mathbf{r}} \quad (4.74)$$

to form the expression for plane-wave excitation in the CFIE.

4.4 CFIE-MOM Implementation with Point Testing Function for the MFIE

The MFIE and the EFIE implementations can also be combined when they use different types of testing functions. However, the contribution of the equations into the linear sum has to be adjusted appropriately. When the MFIE with point testing functions is combined with the EFIE using RWG testing functions, Equations (3.5) and (4.60) can be used as

$$\begin{aligned}
Z_{ik,jl}^C = & \alpha \frac{l_{ik}l_{jl}}{4A_iA_j} \left\{ \int_{S_m} d\mathbf{r}(\mathbf{r} - \mathbf{r}_{eik}) \cdot \int_{S_n} d\mathbf{r}'g(\mathbf{r}, \mathbf{r}')(\mathbf{r}' - \mathbf{r}_{jl}) \right. \\
& \left. - \frac{4}{k^2} \int_{S_i} d\mathbf{r} \int_{S_j} d\mathbf{r}'g(\mathbf{r}, \mathbf{r}') \right\} \\
& + (1 - \alpha) \frac{i}{k} A_i \left\{ \frac{1}{2} \hat{\mathbf{t}}_{ik} \cdot \delta_j(\mathbf{r}_{eik}) \frac{l_{jl}(\mathbf{r}_{eik} - \mathbf{r}_{jl})}{2A_j} \right. \\
& \left. - \frac{\hat{\mathbf{e}}_{ik}}{2} \cdot \frac{l_{jl}}{2A_j} \int_{S_n} d\mathbf{r}'(\mathbf{r}' - \mathbf{r}_{jl}) \times \nabla' g(\mathbf{r}_{eik}, \mathbf{r}') \right\}, \quad (4.75)
\end{aligned}$$

where the MFIE part is multiplied by the area of the testing triangle for the correct weighting. The corresponding expression for the plane-wave excitation becomes

$$\begin{aligned}
v_m = & \alpha \frac{i}{k\eta} \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \mathbf{E}^0 e^{i\mathbf{k} \cdot \mathbf{r}} + (1 - \alpha) \frac{i}{k} A_i \hat{\mathbf{e}}_m \cdot \mathbf{H}^{inc}(\mathbf{r}_{em}) \\
= & \alpha \frac{i}{k\eta} \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \mathbf{E}^0 e^{i\mathbf{k} \cdot \mathbf{r}} + (1 - \alpha) \frac{i}{k\eta} A_i \hat{\mathbf{e}}_m \cdot \hat{\mathbf{k}} \times \mathbf{E}^0 e^{i\mathbf{k} \cdot \mathbf{r}_{em}} \quad (4.76)
\end{aligned}$$

to form the excitation expression for the CFIE. In both (4.75) and (4.76), the MFIE expressions are multiplied by A_i in order to balance the weight of the outer integral in the EFIE, which is missing in the MFIE.

4.5 Results

Figure 4.11 shows eight different triangularly meshed models of sphere having a radius of 0.3 m. The information about these models are given in Table 4.1. Only

Table 4.1: Sphere problems.

Mesh-size	Corresponding Frequency	Number of Unknowns	Radius of the Sphere
6 cm	500 MHz	930	0.5 λ
5 cm	600 MHz	1302	0.6 λ
4 cm	750 MHz	2076	0.75 λ
3 cm	1000 MHz	3723	1.0 λ
2 cm	1500 MHz	8364	1.5 λ
1.5 cm	2000 MHz	14871	2.0 λ
1.0 cm	3000 MHz	32862	3.0 λ
0.5 cm	6000 MHz	132003	6.0 λ

four of these models will be considered at this step due to the limited efficiency of the MOM. Each of these models are excited by plane waves propagating in the $-x$ direction and with electric fields polarized in y direction. Following MOM implementations are used to solve the problems at the corresponding frequencies.

1. *EFIE-MOM*: This implementation uses the EFIE formulation introduced in Chapter 3. RWG functions are used for both testing and basis functions. Singularity extraction is applied for the inner integrals. Numerical parts of the inner integrals as well as the outer integrals are evaluated by adaptive algorithms with 1% error criteria. Figure 4.12 shows the plots of calculated RCS results on the x - y plane. Reference values obtained with an analytical technique, i.e., Mie scattering, are also plotted to make comparisons.
2. *MFIE-MOM with Point Testing*: This implementation uses the MFIE formulation with RWG basis functions and point testing functions. Singularity extraction is applied for the inner integrals and numerical parts are evaluated by adaptive algorithms with 1% error criteria. Figure 4.13 compares the computed RCS values to the analytical results.
3. *MFIE-MOM with RWG Functions and Galerkin Method*: This implementation also uses the MFIE formulation, but Galerkin method is applied with

RWG functions. Singularity extraction is applied for the inner integrals and numerical parts are evaluated by adaptive algorithms with 1% error criteria. For the outer integrals, Gaussian quadrature rules are used. It has been observed that 16-point rule and 7-point rule with singularity extraction for the outer integrals result in convergence of the matrix elements with less than 1% error. Figure 4.14 compares the computed RCS values to the analytical results.

4. *CFIE-MOM with RWG Functions and Galerkin Method*: This implementation is the combination of the first and the third implementations. CFIE parameter α is chosen to be 0.2. Figure 4.15 compares the computed RCS values to the analytical results.

It can be observed that the MFIE implementations produce RCS values that are less accurate than those of the EFIE implementation. This is more obvious in the second implementation, where point testing functions are used. Better testing of the equation with the RWG functions leads to improvement, but some inaccuracy still exists in the third implementation. It should be indicated that this error is not due to the numerical evaluation of the integrals since the evaluation and the convergence of the integrals have been carefully controlled. The reason behind the relative inaccuracy of the MFIE as compared to the EFIE is not well understood. However, the use of basis functions that may be more suitable than the RWG functions is currently under investigation.

If the MFIE is used, the RCS values can be significantly incorrect due to the internal resonance problem [15, 20]. For example, this situation can be observed when the third model in Figure 4.11 is solved at 800 MHz. Figure 4.16 shows the plots of the RCS values obtained by the EFIE, the MFIE, and the CFIE formulations. It can be seen that the values obtained by the MFIE are significantly different (and incorrect) as compared to the the EFIE and the CFIE values.

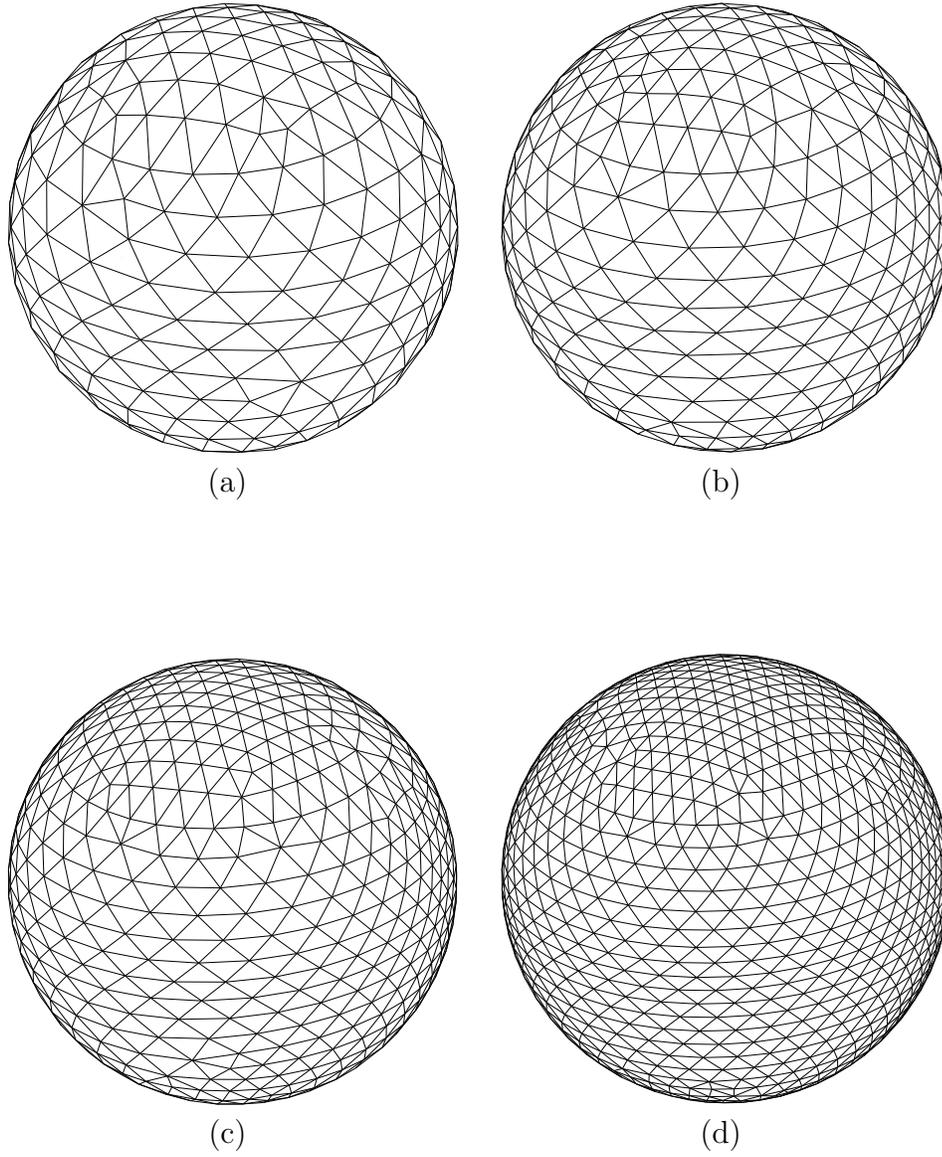
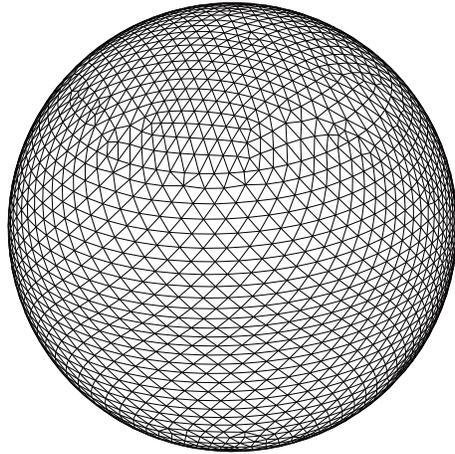
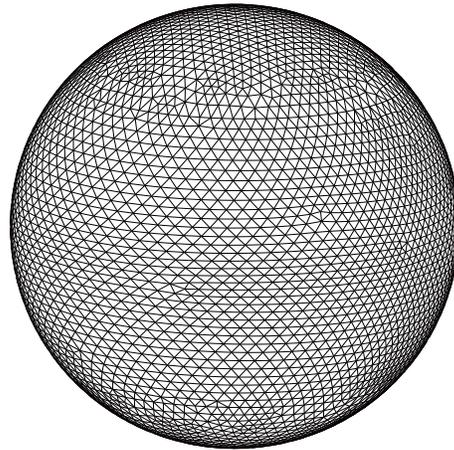


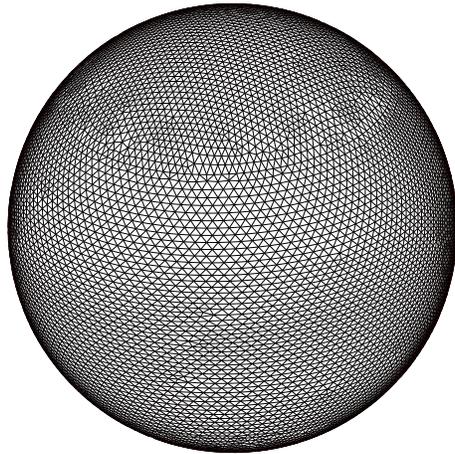
Figure 4.11: Models of a sphere having a radius of 0.3 m with mesh sizes (a) 6 cm, (b) 5 cm, (c) 4 cm, and (d) 3 cm.



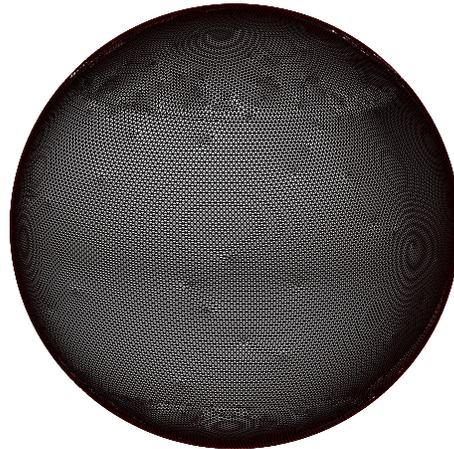
(e)



(f)

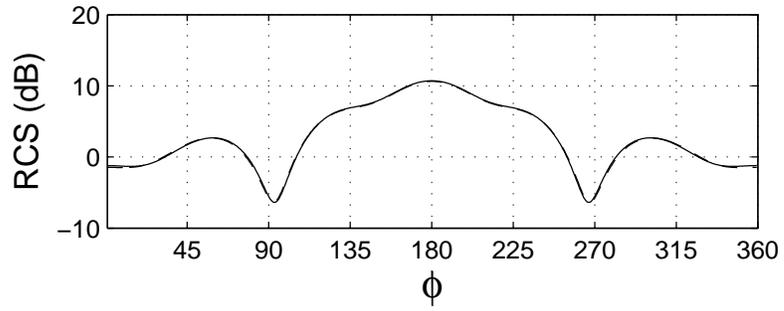


(g)

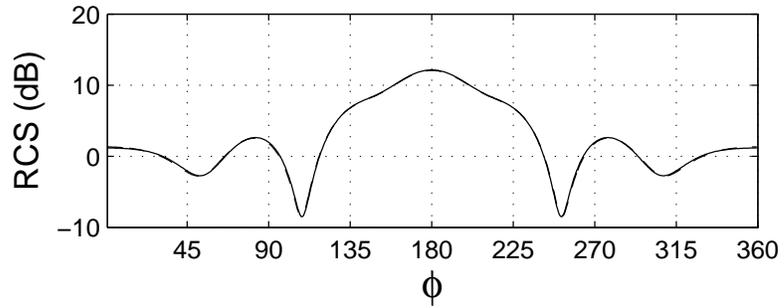


(h)

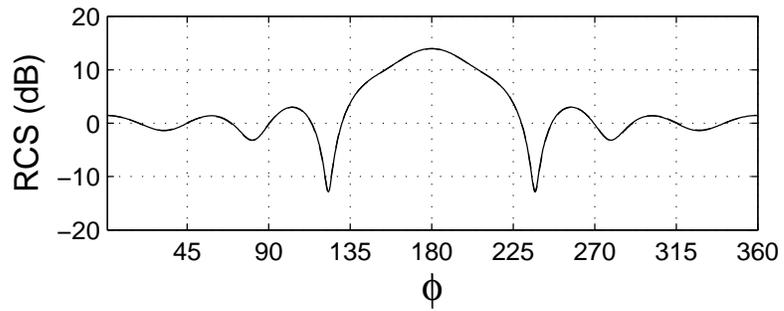
Figure 4.11: (continued) Models of a sphere having a radius of 0.3 m with mesh sizes (e) 2 cm, (f) 1.5 cm, (g) 1.0 cm, and (h) 0.5 cm.



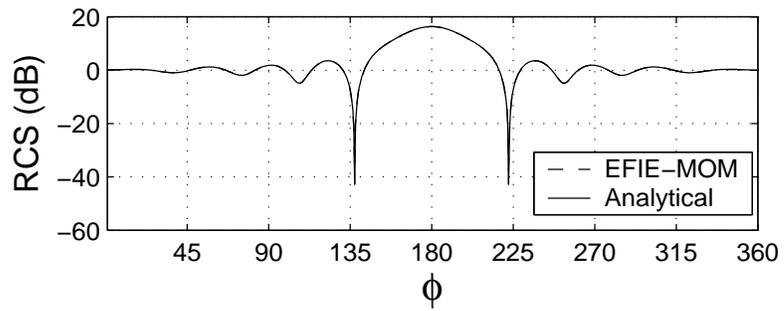
(a)



(b)

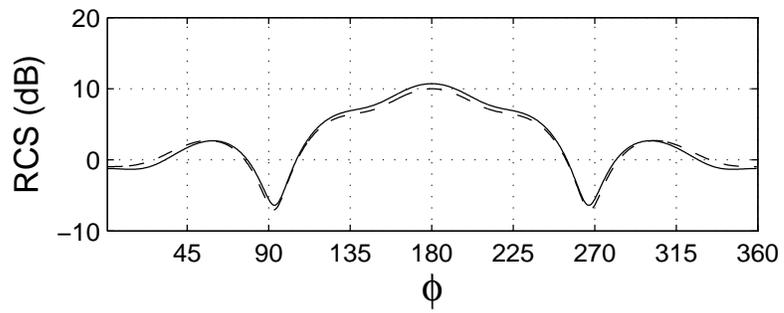


(c)

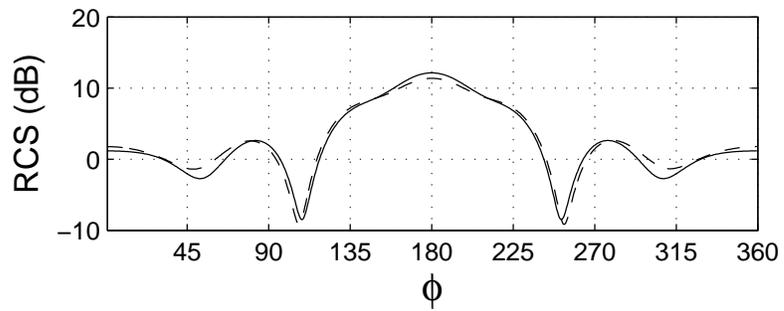


(d)

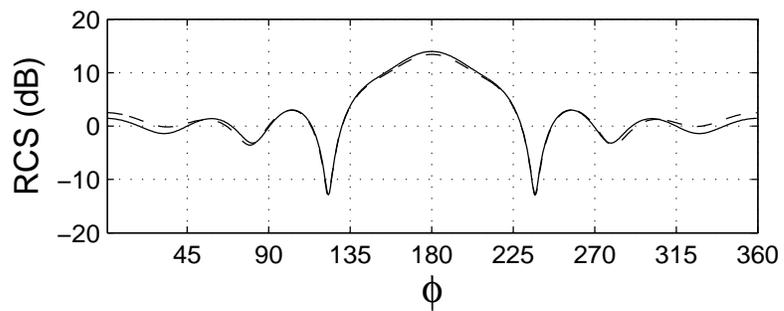
Figure 4.12: RCS values obtained by the EFIE-MOM implementation for a sphere with a radius of (a) 0.5λ , (b) 0.6λ , (c) 0.75λ , and (d) 1.0λ .



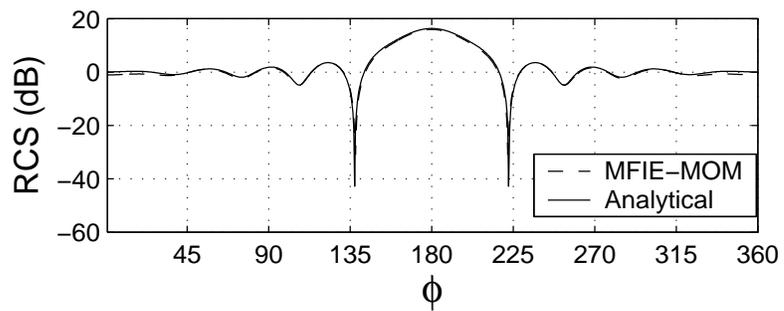
(a)



(b)

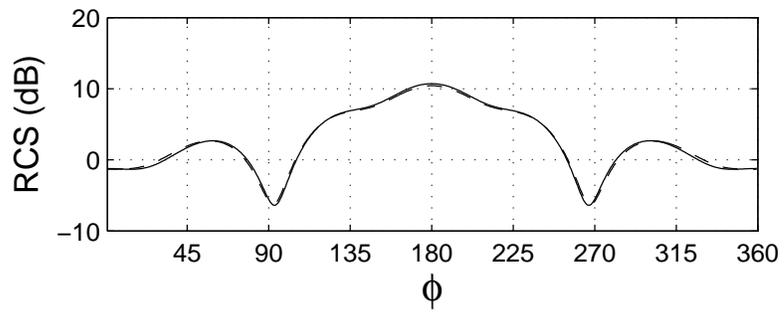


(c)

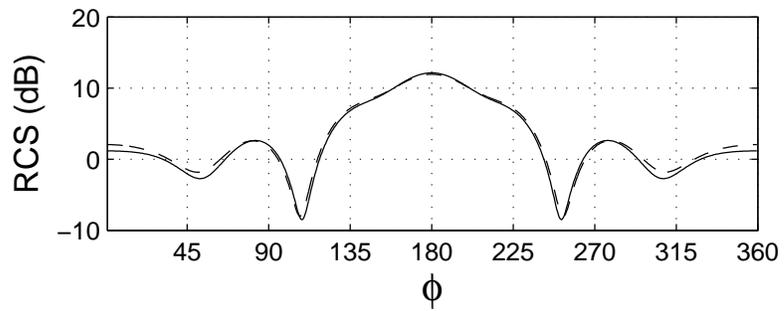


(d)

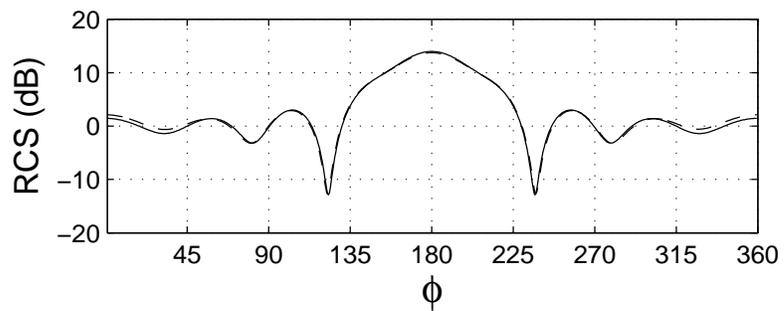
Figure 4.13: RCS values obtained by the MFIE-MOM implementation using point testing functions for a sphere with a radius of (a) 0.5λ , (b) 0.6λ , (c) 0.75λ , and (d) 1.0λ .



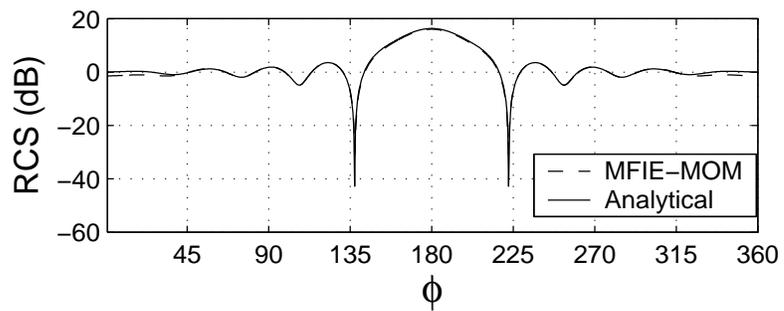
(a)



(b)

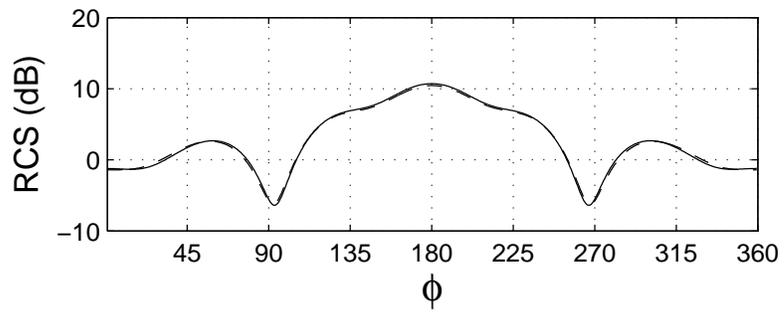


(c)

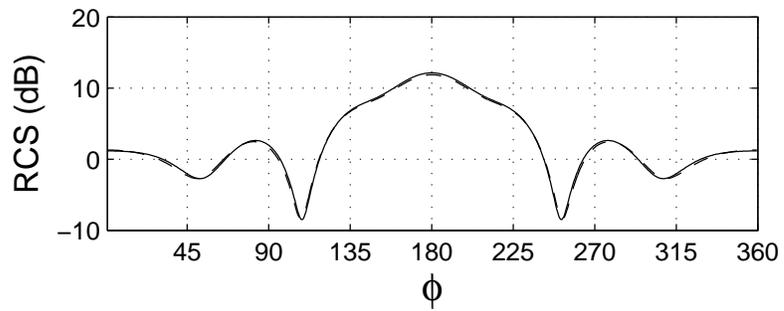


(d)

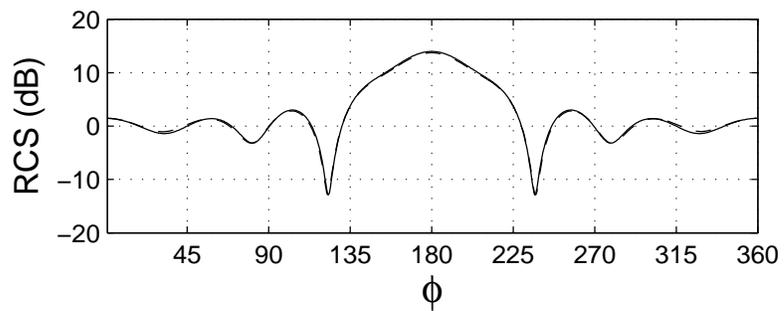
Figure 4.14: RCS values obtained by the MFIE-MOM implementation using the RWG functions and Galerkin method for a sphere with a radius of (a) 0.5λ , (b) 0.6λ , (c) 0.75λ , and (d) 1.0λ .



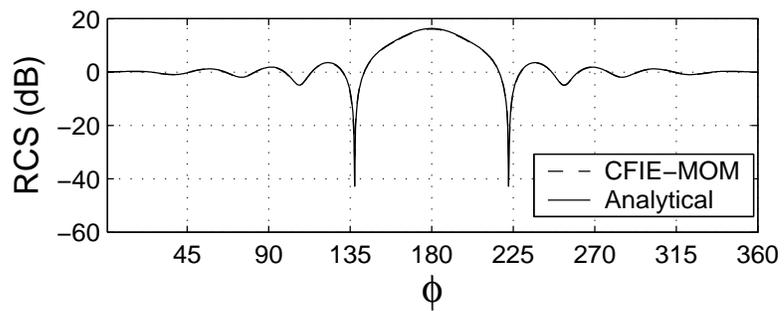
(a)



(b)



(c)



(d)

Figure 4.15: RCS values obtained by the CFIE-MOM implementation using the RWG functions and Galerkin method for a sphere with a radius of (a) 0.5λ , (b) 0.6λ , (c) 0.75λ , and (d) 1.0λ .

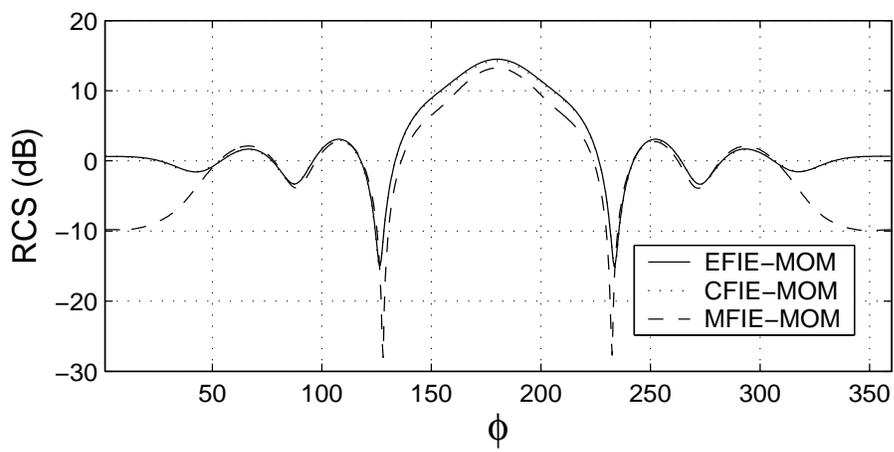


Figure 4.16: RCS values obtained by the EFIE, the MFIE, and the CFIE formulations for a sphere with a radius of 0.8λ . The MFIE suffers from the internal resonance problem.

Chapter 5

FMM Implementations

As it was stated in Chapter 3, the MOM requires $O(N^2)$ memory and $O(N^3)$ processing time for a problem of N unknowns. Therefore, as the number of unknowns increases, this method becomes inefficient in terms of the computational resources. In this chapter, another method called the fast multipole method (FMM) will be introduced, which has lower complexity and can be used to solve larger problems with less memory and processing time. This method was first introduced by Rokhlin [2, 3] and used in many studies to efficiently solve electromagnetic scattering problems, as an alternative to the MOM. A two-level FMM implementation reduces both the complexity of matrix-vector multiplication and memory requirement to $O(N^{3/2})$. Multi-level fast multipole algorithm (MLFMA), later developed by Song and Chew [4]–[6], further reduces the complexity and memory requirement to $O(N \log N)$, making it possible to solve much larger problems with limited hardware in a shorter time.

The FMM is based on the MOM, but calculates the interactions in a different way, using the new expression developed by the factorization of the Green's function. The system is solved iteratively, and the FMM provides a quick way

of matrix-vector multiplication. The efficiency heavily depends on the used integral equation, the iterative solver, and the preconditioning technique. Chapter 6 investigates these components in detail.

5.1 Theory of the FMM

Expressions for the impedance-matrix elements in the MOM implementations using the EFIE and the MFIE formulations are given in (3.6) and (4.4) as

$$Z_{mn}^E = \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \int_{S_n} d\mathbf{r}' \overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{b}_n(\mathbf{r}') \quad (5.1)$$

and

$$\begin{aligned} Z_{mn}^M &= \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \mathbf{b}_n(\mathbf{r}) \\ &\quad - \int_{S_m} d\mathbf{r} \mathbf{t}_m(\mathbf{r}) \cdot \hat{\mathbf{n}} \times \int_{S_n} d\mathbf{r}' \mathbf{b}_n(\mathbf{r}') \times \nabla' g(\mathbf{r}, \mathbf{r}'), \end{aligned} \quad (5.2)$$

respectively. Due to the long-range effect of the Green's function in both formulas, these expressions have to be calculated separately for all basis-testing function pairs in order to form the impedance matrix. The FMM formulation starts with a factorization of the Green's function to make it possible to calculate the interactions in a group-by-group manner.

5.1.1 Factorization of the Green's Function

Rewriting the Green's function as

$$g(\mathbf{r}, \mathbf{r}') = \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} = \frac{e^{ik|\mathbf{D}+\mathbf{d}|}}{|\mathbf{D}+\mathbf{d}|}, \quad (5.3)$$

the addition theorem can be used to factorize the Green's function as

$$\frac{e^{ik|\mathbf{D}+\mathbf{d}|}}{4\pi|\mathbf{D}+\mathbf{d}|} = \frac{ik}{4\pi} \sum_{l=0}^{\infty} (-1)^l (2l+1) j_l(kd) h_l^{(1)}(kD) P_l(\hat{\mathbf{d}} \cdot \hat{\mathbf{D}}), \quad (5.4)$$

where $j_l(x)$ is the spherical Bessel function, $h_l^{(1)}(x)$ is the spherical Hankel function of the first kind, P_l is the Legendre polynomial, and $d < D$. Inserting the identity

$$4\pi i^l j_l(kd) P_l(\hat{\mathbf{d}} \cdot \hat{\mathbf{D}}) = \int d^2 \hat{\mathbf{k}} e^{i\mathbf{k} \cdot \mathbf{d}} P_l(\hat{\mathbf{k}} \cdot \hat{\mathbf{D}}), \quad (5.5)$$

where the integration is over the unit sphere, (5.4) can be rewritten as

$$\frac{e^{ik|\mathbf{D}+\mathbf{d}|}}{|\mathbf{D}+\mathbf{d}|} \approx \frac{ik}{16\pi^2} \int d^2 \hat{\mathbf{k}} \sum_{l=0}^L i^l (2l+1) h_l^{(1)}(kD) P_l(\hat{\mathbf{k}} \cdot \hat{\mathbf{D}}), \quad (5.6)$$

where the summation is truncated at L and the equality turns into an approximation. At this stage, the summation part can be represented by a function, called the translation function, as

$$T_L(k, D, \psi) = \frac{ik}{4\pi} \sum_{l=0}^L i^l (2l+1) h_l^{(1)}(kD) P_l(\cos\psi), \quad (5.7)$$

where ψ is the angle between the unit vectors $\hat{\mathbf{k}}$ and $\hat{\mathbf{D}}$. Then, the Green's function becomes

$$\frac{e^{ik|\mathbf{D}+\mathbf{d}|}}{4\pi|\mathbf{D}+\mathbf{d}|} \approx \frac{1}{4\pi} \int d^2 \hat{\mathbf{k}} e^{i\mathbf{k} \cdot \mathbf{d}} T_L(k, D, \psi). \quad (5.8)$$

5.1.2 Calculation of an Interaction

It is desired to calculate the interaction between a basis function and a testing function, as shown in Figure 5.1, by using (5.1) or (5.2). To insert Equation (5.8) into the formulation, the distance between the basis and testing functions is divided into segments as

$$\begin{aligned} \mathbf{r} - \mathbf{r}' &= (\mathbf{r} - \mathbf{r}_{fm}) + (\mathbf{r}_{fm} - \mathbf{r}_c) + (\mathbf{r}_c - \mathbf{r}_{c'}) + (\mathbf{r}_{c'} - \mathbf{r}_{fn}) + (\mathbf{r}_{fn} - \mathbf{r}') \\ &= \mathbf{r}_{rfm} + \mathbf{r}_{fmc} + \mathbf{r}_{cc'} + \mathbf{r}_{c'fn} + \mathbf{r}_{fnr'}, \end{aligned} \quad (5.9)$$

where \mathbf{r}_{fm} and \mathbf{r}_{fn} represent the reference points in the testing and basis functions, respectively, while \mathbf{r}_c stands for a point near the testing function and $\mathbf{r}_{c'}$

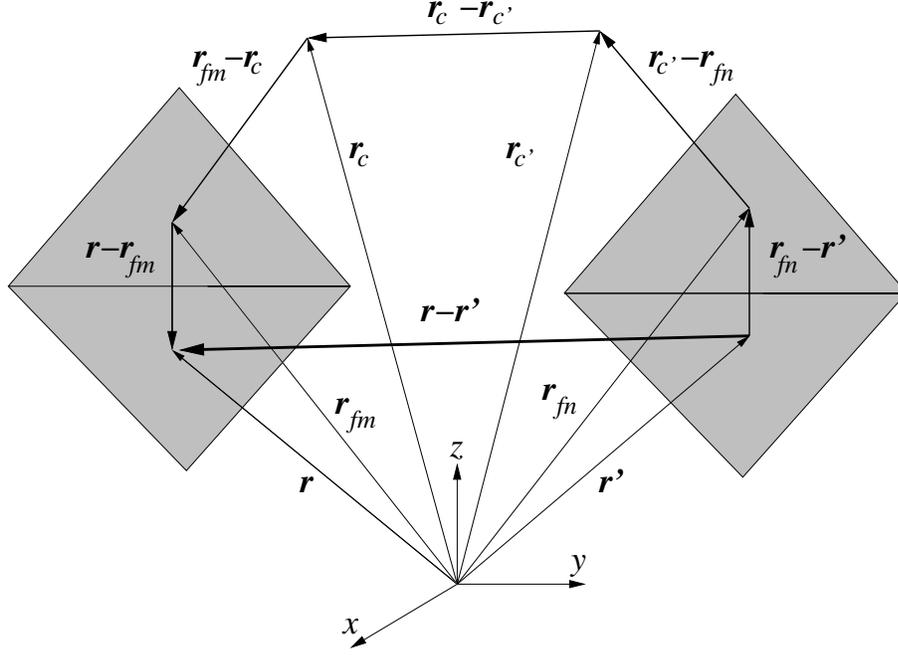


Figure 5.1: Dividing $\mathbf{r} - \mathbf{r}'$ into segments.

stands for another one near the basis function. Then, the segments can be arranged as

$$\begin{aligned}
 \mathbf{d}_1 &= \mathbf{r}_{rfm} + \mathbf{r}_{fmc} \\
 \mathbf{d}_2 &= \mathbf{r}'_{c'fn} + \mathbf{r}'_{fnr'} \\
 \mathbf{d} &= \mathbf{d}_1 + \mathbf{d}_2 \\
 \mathbf{D} &= \mathbf{r}_{cc'}
 \end{aligned} \tag{5.10}$$

and (5.8) becomes

$$\frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} \approx \frac{1}{4\pi} \int d^2\hat{\mathbf{k}} e^{i\mathbf{k}\cdot(\mathbf{r}_{rfm}+\mathbf{r}_{fmc}+\mathbf{r}'_{c'fn}+\mathbf{r}'_{fnr'})} T_L(k, |\mathbf{r}_{cc'}|, \hat{\mathbf{r}}_{cc'} \cdot \hat{\mathbf{k}}). \tag{5.11}$$

In addition, using the above, the dyadic Green's function can be expressed as

$$\overline{\mathbf{G}}(\mathbf{r}, \mathbf{r}') \approx \frac{1}{4\pi} \int d^2\hat{\mathbf{k}} (\overline{\mathbf{I}} - \hat{\mathbf{k}}\hat{\mathbf{k}}) e^{i\mathbf{k}\cdot(\mathbf{r}_{rfm}+\mathbf{r}_{fmc}+\mathbf{r}'_{c'fn}+\mathbf{r}'_{fnr'})} T_L(k, |\mathbf{r}_{cc'}|, \hat{\mathbf{r}}_{cc'} \cdot \hat{\mathbf{k}}). \tag{5.12}$$

Finally, the interaction between the basis and testing functions can be written as

$$Z_{mn}^E = \frac{1}{4\pi} \int d^2\hat{\mathbf{k}} \mathbf{F}_{fmc}^E(\hat{\mathbf{k}}) T_L(k, |\mathbf{r}_{cc'}|, \hat{\mathbf{r}}_{cc'} \cdot \hat{\mathbf{k}}) \cdot \mathbf{F}_{fnc'}^E(\hat{\mathbf{k}}) \tag{5.13}$$

for the EFIE formulation and

$$Z_{mn}^M = \frac{k}{4\pi i} \int d^2\hat{\mathbf{k}} \mathbf{F}_{fmc}^M(\hat{\mathbf{k}}) T_L(k, |\mathbf{r}_{cc'}|, \hat{\mathbf{r}}_{cc'} \cdot \hat{\mathbf{k}}) \cdot \mathbf{F}_{fnc'}^M(\hat{\mathbf{k}}) \quad (5.14)$$

for the MFIE formulation, where

$$\mathbf{F}_{fmc}^E(\hat{\mathbf{k}}) = e^{i\mathbf{k}\cdot(\mathbf{r}_{fm}-\mathbf{r}_c)} \int_{S_m} d\mathbf{r} e^{i\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}_{fm})} (\bar{\mathbf{I}} - \hat{\mathbf{k}}\hat{\mathbf{k}}) \cdot \mathbf{t}_m(r) \quad (5.15)$$

$$\mathbf{F}_{fnc'}^E(\hat{\mathbf{k}}) = e^{i\mathbf{k}\cdot(\mathbf{r}_{c'}-\mathbf{r}_{fn})} \int_{S_n} d\mathbf{r}' e^{-i\mathbf{k}\cdot(\mathbf{r}'-\mathbf{r}_{fn})} (\bar{\mathbf{I}} - \hat{\mathbf{k}}\hat{\mathbf{k}}) \cdot \mathbf{b}_n(\mathbf{r}') \quad (5.16)$$

$$\mathbf{F}_{fmc}^M(\hat{\mathbf{k}}) = -\hat{\mathbf{k}} \times e^{i\mathbf{k}\cdot(\mathbf{r}_{fm}-\mathbf{r}_c)} \int_{S_m} d\mathbf{r} e^{i\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}_{fm})} \mathbf{t}_m(r) \times \hat{\mathbf{n}} \quad (5.17)$$

$$\mathbf{F}_{fnc'}^M(\hat{\mathbf{k}}) = e^{i\mathbf{k}\cdot(\mathbf{r}_{c'}-\mathbf{r}_{fn})} \int_{S_n} d\mathbf{r}' e^{-i\mathbf{k}\cdot(\mathbf{r}'-\mathbf{r}_{fn})} \mathbf{b}_n(\mathbf{r}'). \quad (5.18)$$

It should be noted that the self-interaction term in the MFIE is omitted, since the FMM can only be applied to distant interactions, as will be explained later. Since the CFIE is simply the linear combination of the EFIE and the MFIE, it is also possible to write the interaction in the CFIE formulation as

$$\begin{aligned} Z_{mn}^C &= \alpha Z_{mn}^E + (1 - \alpha) \frac{i}{k} Z_{mn}^M \\ &= \frac{1}{4\pi} \int d^2\hat{\mathbf{k}} \mathbf{F}_{fmc}^C(\hat{\mathbf{k}}) T_L(k, |\mathbf{r}_{cc'}|, \hat{\mathbf{r}}_{cc'} \cdot \hat{\mathbf{k}}) \cdot \mathbf{F}_{fnc'}^C(\hat{\mathbf{k}}), \end{aligned} \quad (5.19)$$

where

$$\begin{aligned} \mathbf{F}_{fmc}^C(\hat{\mathbf{k}}) &= \alpha e^{i\mathbf{k}\cdot(\mathbf{r}_{fm}-\mathbf{r}_c)} \int_{S_m} d\mathbf{r} e^{i\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}_{fm})} (\bar{\mathbf{I}} - \hat{\mathbf{k}}\hat{\mathbf{k}}) \cdot \mathbf{t}_m(r) \\ &\quad - (1 - \alpha) \hat{\mathbf{k}} \times e^{i\mathbf{k}\cdot(\mathbf{r}_{fm}-\mathbf{r}_c)} \int_{S_m} d\mathbf{r} e^{i\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}_{fm})} \mathbf{t}_m(r) \times \hat{\mathbf{n}} \end{aligned} \quad (5.20)$$

and

$$\mathbf{F}_{fnc'}^C(\hat{\mathbf{k}}) = e^{i\mathbf{k}\cdot(\mathbf{r}_{c'}-\mathbf{r}_{fn})} \int_{S_n} d\mathbf{r}' e^{-i\mathbf{k}\cdot(\mathbf{r}'-\mathbf{r}_{fn})} (\bar{\mathbf{I}} - \hat{\mathbf{k}}\hat{\mathbf{k}}) \cdot \mathbf{b}_n(\mathbf{r}'). \quad (5.21)$$

It should be noted that \mathbf{F}_{fmc}^M has only θ and ϕ components so that $\mathbf{F}_{fnc'}^M$ can be reduced to $\mathbf{F}_{fnc'}^E$ without any complication and $\mathbf{F}_{fnc'}^C$ becomes the same as $\mathbf{F}_{fnc'}^E$.

5.1.3 Clustering and Evaluation of the Interactions

The addition theorem in (5.4) is valid only if $d < D$. Therefore, the new expressions can be used to calculate the interactions between testing and basis functions if and only if

$$(\mathbf{r} - \mathbf{r}_c) + (\mathbf{r}_{c'} - \mathbf{r}') < (\mathbf{r}_c - \mathbf{r}_{c'}), \quad (5.22)$$

which means that the points c and c' should be chosen close to the testing and basis functions, respectively, compared to the distance between these points.

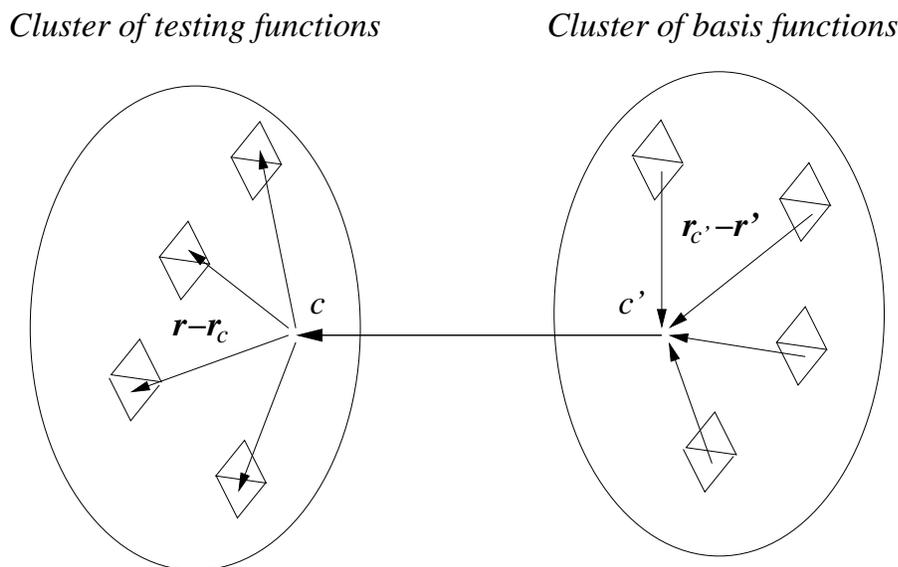


Figure 5.2: Clustering of basis and testing functions provides a single translation between the clusters.

Figure 5.2 shows basis and testing functions, which are grouped (clustered) according to their locations in space. If the points c and c' are chosen to be common points for all functions in the testing and basis clusters, respectively, the translation function, $T_L(k, |\mathbf{r}_{cc'}|, \hat{\mathbf{r}}_{cc'} \cdot \hat{\mathbf{k}})$, becomes the same for all interactions between pairs of functions in these clusters. Then, instead of calculating the interactions separately, a single translation between the clusters can be used for efficiency.

In the FMM, basis and testing functions are grouped to calculate the interactions in a group-by-group manner. The procedure can be outlined as follows:

1. In each basis group, the expressions in (5.16) and (5.18) are calculated for all basis functions, using the common point (cluster center) chosen for the group. For each basis function, the expression is evaluated for K points (directions) on the unit sphere and the value at each direction can be interpreted as the radiation of the basis function in that direction from the cluster center.

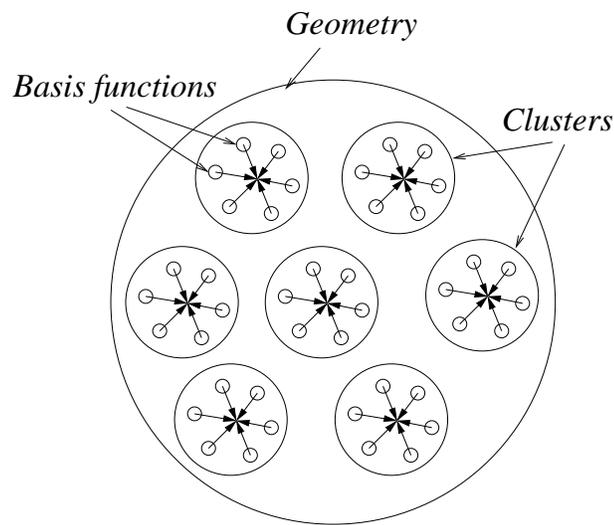


Figure 5.3: Aggregation process in basis clusters.

2. In each basis group, the values of the radiations of the basis functions are added at each direction. Therefore, the overall radiation of the group is found with respect to the cluster center. This aggregation process is illustrated in Figure 5.3.
3. The translation function is defined for each pair of basis and testing groups, instead of all pairs of the functions. For each group-group interaction, the translation function multiplies the overall radiation of the basis group and translates it to the center of the testing cluster. This operation can be

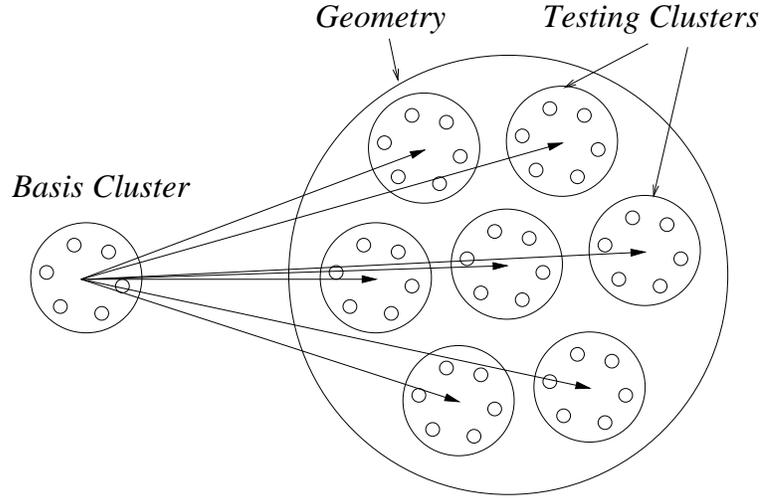


Figure 5.4: Translation from a basis cluster to testing clusters.

interpreted as the translation of the radiation into incoming waves for the testing functions. Figure 5.4 sketches this step for a basis cluster.

4. To test the incoming wave, the expressions in (5.15) and (5.17) are calculated for each testing function. Following the translations, the radiations from all basis groups are added at the common point of the testing group. Then, the total radiation from all basis functions can be tested by each testing function at K points. This disaggregation process is illustrated in Figure 5.5.
5. A group-group interaction is completed when the integration in (5.13), (5.14), or (5.19) is evaluated on the unit sphere.

The FMM is based on the iterative solution of the system. Therefore, the impedance matrix is not stored in the memory. Instead, the procedure above is used in every iteration to perform the matrix-vector products required by the iterative solver. The coefficient vector is included in the first step of the procedure, as the radiated fields of the basis functions are aggregated. Radiation of each basis function is multiplied by the corresponding coefficient to weight its

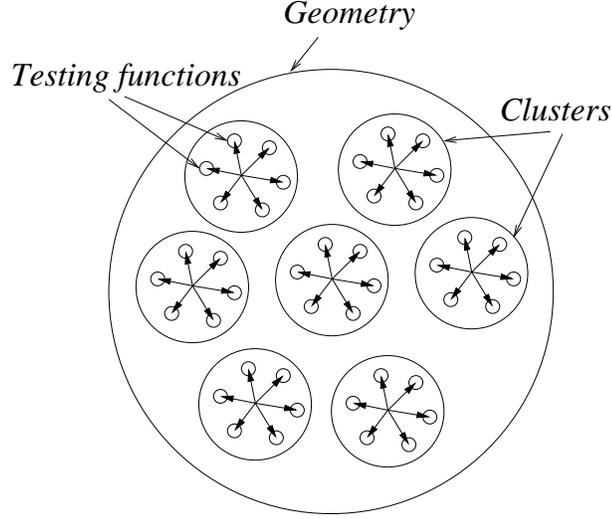


Figure 5.5: Disaggregation process in testing clusters.

contribution to the group radiation. Then, the other steps are applied to find the resulting vector of the multiplication.

Due to the restriction in the addition theorem, not all interactions can be calculated by the FMM. When the basis and testing clusters are close to each other, the distance between the cluster centers becomes small and (5.22) is not satisfied. These near-field interactions are calculated as in the MOM and combined with the FMM interactions in the iterative procedure. The matrix-vector multiplication in the FMM can be written as

$$\begin{aligned}
 \sum_{n=1}^N Z_{mn} a_n &= \sum_{C' \in N(C)} \sum_{n \in C'} Z_{mn} a_n \\
 &+ \frac{1}{4\pi} \int d^2 \hat{\mathbf{k}} \mathbf{F}_{fmc}(\hat{\mathbf{k}}) \cdot \sum_{C' \in F(C)} T_L(k, |\mathbf{r}_{cc'}|, \hat{\mathbf{r}}_{cc'} \cdot \hat{\mathbf{k}}) \sum_{n \in C'} \mathbf{F}_{fnc'}(\hat{\mathbf{k}}) a_n,
 \end{aligned} \tag{5.23}$$

where m stands for the testing function, which belongs to cluster C . The clusters close to this cluster are represented as $N(C)$, while the ones in the far-field zone are represented as $F(C)$. On the right-hand side of (5.23), the first term includes the near-field interactions between the m^{th} testing function, and those

basis functions, which belong to clusters in the near-field of C . Then, the second term adds the contributions from the other basis functions, belonging to clusters far from C . The efficiency of an FMM implementation relies on keeping the number of near-field interactions small as compared to the far-field interactions.

5.1.4 Truncation Number

The summation in (5.7) is truncated at L , which is an important parameter requiring special attention. One of the error sources of the FMM is this truncation since the addition theorem is originally based on an infinite sum. To reduce the error in the factorization of Green's function, one may attempt to choose L large and expect the error to drop as L gets larger. However, this is not the case due to the finite precision used in the computations.

As l exceeds the variable x and gets larger, the Hankel function, $h_l^{(1)}(x)$ in (5.4) tends to grow rapidly. Figure 5.6 shows the magnitude of the imaginary part of the Hankel function with respect to l , for different values of x as 1, 2, ..., 10. It can be observed that the plots tend to blow up whenever l exceeds x .

The growing values of the Hankel function are suppressed by the Bessel function so that the contribution to the summation in (5.4) becomes smaller for larger harmonics. On the other hand, the equality holds better as these small contributions are included.

In principle, the equality in (5.6) holds better as L gets larger. However, this requires the perfect evaluation of the Hankel function so that the angular integration can be performed correctly. Due to the Hankel function, the integrand takes very large values with a large variation introduced by the Legendre polynomial. The perfect evaluation would result in a moderate value for the integration.

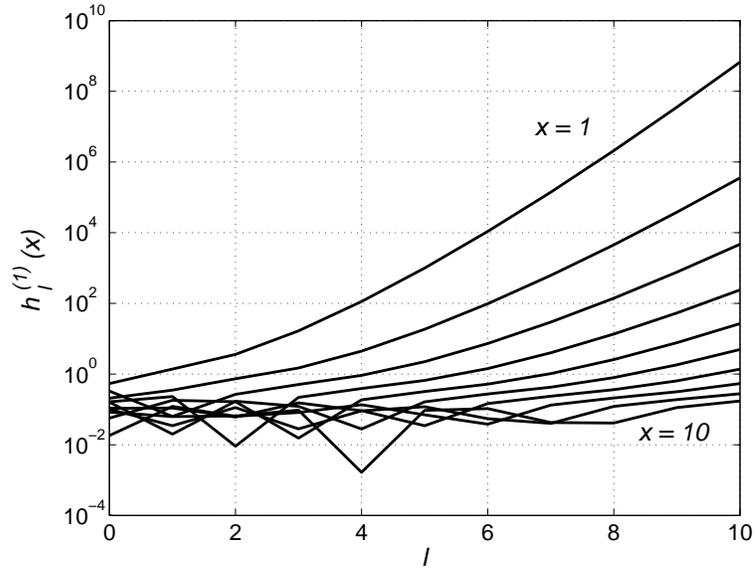


Figure 5.6: Magnitude of the imaginary part of the Hankel function, $h_l^{(1)}(x)$ with respect to l for different values of x .

However, this becomes impossible when the used precision in the computation is insufficient to represent the value of the integrand correctly. Then, the angular integration fails and the equality does not hold.

Consequently, the truncation number, L , should be chosen small enough to avoid divergence of the summation, while it should be large enough for the convergence. Various ways of choosing the truncation number have been developed. A refined formula is given as [6]

$$L \approx kd + 1.8d_0^{2/3}(kd)^{1/3}, \quad (5.24)$$

where d_0 is the desired correct number of digits. Considering (5.10), it should be noted that value of d ranges from 0 to the sum of the two cluster diameters, according to the location of testing and basis functions with respect to the cluster centers. However, for a cluster-cluster pair, a single value of L have to be used. As a result, the value of d in (5.24) is generally determined by considering the worst case when d has its largest value.

To control L and avoid the deviation of the sum, a simple method is to compare L to $kr_{cc'}$, which is the variable for the Hankel function. If L exceeds this value, the clusters are said to be in the near-field of each other and the FMM should not be used between these clusters. Then, the interactions between the basis and testing functions in these clusters have to be calculated as in the MOM.

Using formulas in the determination of L is well suited for static clustering. A usual way of clustering the unknowns is to put the geometry in a fictitious box and to divide the box into sub boxes [6]. Chapter 7 introduces multi-level FMM implementations, which use this type of clustering.

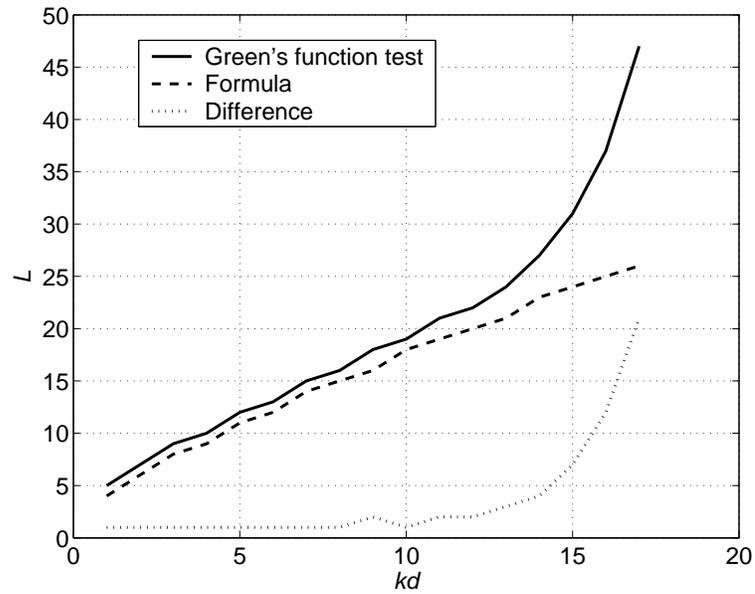


Figure 5.7: Values of L obtained with the formula given in Equation (5.24) and direct test of Equation (5.8) for an accuracy of three digits.

The formula given in Equation (5.24) can be tested by comparing its values for L with the ones obtained by a direct evaluation of Equation (5.8). Figure 5.7 shows the L values when $kr_{cc'} = 20$ and $d_0 = 3$ for an accuracy of three digits.

For L values up to 20, the formula shows very good performance with small error. After that, its values begin to deviate from the correct curve and the error increases drastically. This result shows that the formula in (5.24) may not be reliable for every possible case.

Another way of determining the truncation number is to form a look-up table, filled with values for L according to variables $d, r_{cc'}$ and d_0 . This table can be formed easily by using Equation (5.8) and checking L for the given values of variables.

The method using the look-up table works well with dynamic clustering, in which the clusters have various sizes. In this method, after determining the value for L , it should be still checked against $kr_{cc'}$. Although these values satisfy Equation (5.8), their usage in the calculation of impedance-matrix elements may result in failure due to large values of the Hankel function. Another strategy is to restrict the value of L to be under a predetermined value in the look-up table.

5.1.5 Integration Points

The angular integration points used in Equations (5.13), (5.14), and (5.19) should be chosen carefully and they must be sufficient to perform a quadrature that is exact for all spherical harmonics of orders less than $2L$. A simple method is presented in [3] as choosing the values of θ such that they are the zeros of the Legendre polynomial, $P_L(\cos\theta)$. Then, ϕ can be sampled uniformly at $2L$ points and the number of samples, K , becomes $2L^2$. Figure 5.8 shows the distribution of the integration points when $L = 5$.

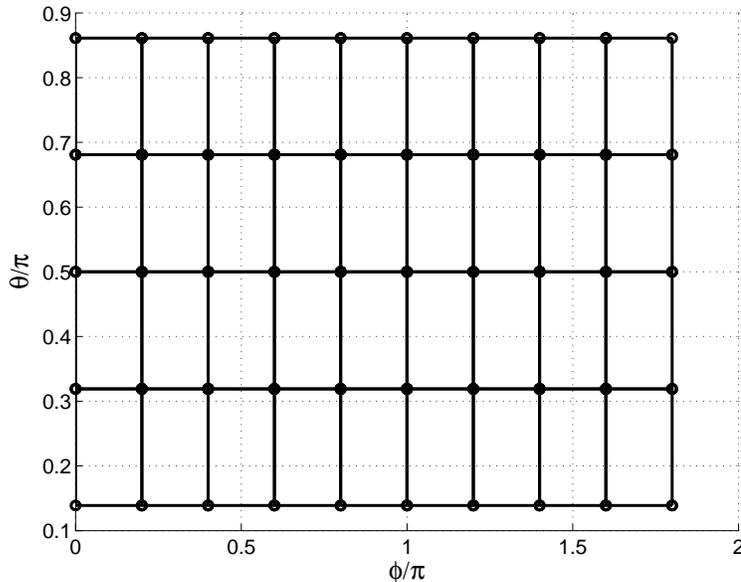


Figure 5.8: Angular integration points for $L = 5$.

5.1.6 Computational Analysis of the FMM

The FMM does not fill the impedance matrix completely as MOM does. Only the near-field interactions are calculated and stored in the memory and this leads to a reduction in the memory usage. In addition, the rest of the interactions are calculated in a group-by-group manner, which reduces the number of operations significantly. In this subsection, individual steps of an FMM implementation will be investigated to determine the requirements for the processing time and memory usage for the solution of a problem with N unknowns.

An FMM implementation has three main parts:

1. *Aggregation*: Equations (5.16) and (5.18) have to be calculated for all basis functions and added to the overall radiation of the clusters. The calculation part is done once for a problem, but the addition has to be done at each iteration. For a basis function, $K = 2L^2 \propto (kd)^2$ sample points are needed on the unit sphere. If M is the number of clusters, there will be about

N/M unknowns in each cluster and $kd \propto (N/M)^{1/2}$. As a result,

$$K \propto N/M \tag{5.25}$$

and $O(NK) = O(N^2/M)$ FLOPs are required to calculate the radiations. The aggregation process can also be performed with $O(N^2/M)$ FLOPs for each iteration. The memory requirement appears to be $O(N^2/M)$ at this step to store the radiations of the basis functions.

2. *Translation:* For each pair of clusters that are far from each other, a translation has to be computed in K directions. This requires $O(KM^2) = O(NM)$ FLOPs to be done for each iteration. On the other hand, calculation of the translation function at K directions for each cluster-cluster pair requires $O(LKM^2) = O(N^{3/2}M^{1/2})$ FLOPs, while the memory requirement to store the translation function is $O(NM)$.
3. *Disaggregation:* At each cluster, the sum of the translation results has to be multiplied by the receiving pattern of the testing functions. This requires $O(NK) = O(N^2/M)$ FLOPs, which should be done at each iteration. Calculation of the receiving patterns of the testing functions has to be done once for a problem and requires $O(N^2/M)$ operations, while $O(N^2/M)$ memory is required to store them.

In addition to these three steps, an FMM implementation also includes some near-field interactions to be evaluated as in the MOM. For a single cluster, there exist $O(1)$ near clusters, so that the number of near-field interactions for this single cluster becomes $O(N^2/M^2)$. Considering all of the clusters, the near-field interactions require $O(N^2/M)$ FLOPs. The memory requirement to keep these interactions is also $O(N^2/M)$.

As a result, an FMM implementation requires

$$\begin{aligned}
& C_{agg1}N^2/M + C_{agg2}IN^2/M + C_{trans1}INM + C_{trans2}N^{3/2}M^{1/2} \\
& + C_{dis1}N^2/M + C_{dis2}IN^2/M + C_{near}N^2/M \\
& = O(IN^2/M) + O(INM) + O(N^{3/2}M^{1/2})
\end{aligned} \tag{5.26}$$

FLOPs, where I is the number of iterations, while the memory requirement appears to be

$$C_{agg}N^2/M + C_{trans}NM + C_{dis}N^2/M + C_{near}N^2/M = O(N^2/M) + O(NM). \tag{5.27}$$

The third term in the processing time is usually neglected, since the constant C_{trans2} is small compared to other constants, and the number of iterations, I , usually increases with the number of unknowns. Then, choosing $M = N^{1/2}$ leads to an optimization in both the processing time and the memory requirement. With this choice, the memory requirement becomes $O(N^{3/2})$ and the processing time becomes $O(N^{3/2})$ per iteration for the FMM.

In Chapter 3, the MOM was shown to require $O(N^2)$ memory and $O(IN^2)$ processing time. They are reduced to $O(N^{3/2})$ and $O(IN^{3/2})$, respectively, by the FMM. Therefore, the FMM makes it possible to solve larger problems with lower increase in the need of computational sources. Results proving this efficiency will be presented at the end of this chapter.

5.2 FMM Implementation with RWG Basis Functions

If the RWG functions are used to expand the current, the expression in (5.16) becomes

$$\mathbf{F}_{fjl}^E(\hat{\mathbf{k}}) = \frac{l_{jl}}{A_j} e^{i\mathbf{k}\cdot(\mathbf{r}_{c'}-\mathbf{r}_{fjl})} \int_{S_j} d\mathbf{r}' e^{-i\mathbf{k}\cdot(\mathbf{r}'-\mathbf{r}_{fjl})} (\bar{\mathbf{I}} - \hat{\mathbf{k}}\hat{\mathbf{k}}) \cdot (\mathbf{r}' - \mathbf{r}_{jl}) \quad (5.28)$$

for the basis triangle j and the function alignment l . It should be noted that this expression can also be used in the MFIE formulation since \mathbf{F}_{fmc}^M has only θ and ϕ components and the r component in \mathbf{F}_{fnc}^M can be eliminated. The evaluation of (5.28) for each basis function involves integrations in the form of

$$\int_{S_j} d\mathbf{r}' e^{-i\mathbf{k}\cdot(\mathbf{r}'-\mathbf{r}_{fjl})} (\mathbf{r}' - \mathbf{r}_{jl}). \quad (5.29)$$

For each triangle, the reference point of the basis function can be chosen as the vertex, at which the current starts or ends. Then, the integrals to be evaluated become

$$\int_{S_j} d\mathbf{r}' e^{-i\mathbf{k}\cdot(\mathbf{r}'-\mathbf{r}_{jl})} (\mathbf{r}' - \mathbf{r}_{jl}). \quad (5.30)$$

These integrals can be calculated analytically, instead of a numerical treatment.

If the Galerkin method is used and testing functions are also chosen as RWG functions, (5.15) becomes

$$\mathbf{F}_{fik}^E(\hat{\mathbf{k}}) = \frac{l_{ik}}{A_i} e^{i\mathbf{k}\cdot(\mathbf{r}_{fik}-\mathbf{r}_c)} \int_{S_i} d\mathbf{r} e^{i\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}_{fik})} (\bar{\mathbf{I}} - \hat{\mathbf{k}}\hat{\mathbf{k}}) \cdot (\mathbf{r} - \mathbf{r}_{ik}), \quad (5.31)$$

where i represents the testing triangle, while k is the alignment of the function on this triangle. Comparison of (5.31) to (5.28) for a RWG function shows that the receiving pattern of the function is simply the complex conjugate of the radiation pattern. Therefore, for each function, it is sufficient to calculate only (5.28). When the function is used for testing, the same data can be used after a simple

conjugation operation.

The complex-conjugate relation holds only for the EFIE formulation. It should be noted that the testing functions require a different treatment in the MFIE formulation. With the RWG functions, Equation (5.17) becomes

$$\mathbf{F}_{fikc}^M(\hat{\mathbf{k}}) = -\frac{l_{ik}}{A_i} \hat{\mathbf{k}} \times e^{i\mathbf{k}\cdot(\mathbf{r}_{fik}-\mathbf{r}_c)} \int_{S_i} d\mathbf{r} e^{i\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}_{fik})} (\mathbf{r} - \mathbf{r}_{ik}) \times \hat{\mathbf{n}}, \quad (5.32)$$

which is not simply the complex conjugate of the expression in (5.31).

If point testing functions are used as testing functions, Equation (4.57) can be inserted in (5.17) to obtain

$$\begin{aligned} \mathbf{F}_{fmc}^M(\hat{\mathbf{k}}) &= -\hat{\mathbf{k}} \times e^{i\mathbf{k}\cdot(\mathbf{r}_{fm}-\mathbf{r}_c)} \int_{S_m} d\mathbf{r} e^{i\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}_{fm})} \mathbf{t}_m(r) \times \hat{\mathbf{n}} \\ &= -\hat{\mathbf{k}} \times e^{-i\mathbf{k}\cdot\mathbf{r}_c} \int_{S_m} d\mathbf{r} e^{i\mathbf{k}\cdot\mathbf{r}} \frac{1}{2} \left\{ \delta(\mathbf{r} - \mathbf{r}_{em}) \hat{t}_{m1} + \delta(\mathbf{r} - \mathbf{r}_c) \hat{t}_{m2} \right\} \times \hat{\mathbf{n}} \\ &= -(\hat{\mathbf{k}} \times \hat{\mathbf{e}}) e^{i\mathbf{k}\cdot(\mathbf{r}_{em}-\mathbf{r}_c)}, \end{aligned} \quad (5.33)$$

where $\hat{\mathbf{e}}$ is the unit vector parallel to the edge and \mathbf{r}_{em} represents the mid-point of the edge. This expression is relatively simpler to evaluate, but leads to less accuracy as in the MOM case shown in the previous chapter.

5.3 Results

To check the accuracy and efficiency of the FMM implementations, the sphere models shown in Figure 4.11 can be considered again. It should be recalled that these models are excited by a y -polarized plane wave propagating in the $-x$ direction. Two FMM implementations are used to solve the problems at the corresponding frequencies:

1. *EFIE-FMM*: This implementation uses the EFIE formulation with the RWG functions for both testing and basis functions. For near-field interactions,

the integrals are evaluated by adaptive algorithms with 1% error criteria, and singularity extraction is applied to inner integrals.

2. *CFIE-FMM*: This implementation combines the EFIE and the MFIE implementations with $\alpha = 0.2$. RWG functions are used for both basis and testing functions. For near-field interactions, the inner integrals are evaluated by adaptive algorithms with 1% error criteria and with the aid of singularity extraction. The outer integrals are evaluated by 7-point Gaussian quadrature rule, while the singularity extraction applied to outer integrals in the MFIE.

5.3.1 Accuracy of the FMM Implementations

Different from the MOM, the FMM carries two extra error sources. One of them is the truncation of the summation in the translation function, and the other is the integration error in the angular integration. Both of these errors are directly controlled by single parameter, d_0 .

Figure 5.9 shows the RCS values in the x - y plane, calculated by the EFIE-FMM implementation with $d_0 = 3$. Corresponding plots for the CFIE-FMM are shown in Figure 5.10. Both figures show that the FMM implementations have good accuracies as compared to the MOM implementations, although they are much more efficient.

5.3.2 Efficiency of the FMM Implementations

In the next chapter, it will be shown that the FMM implementations with the CFIE formulation are more efficient than those with the EFIE formulation. At this stage, only the EFIE-MOM and the EFIE-FMM implementations will be compared to observe the superiority of the FMM over the MOM.

Tables 5.1 and 5.2 show the processing times and required memory, respectively, to solve various sphere problems. As the problem size gets larger, increases in the memory requirement and processing time are slower for the FMM implementation. Thus, the FMM makes it possible to solve larger problems, which would be very difficult to solve by using the MOM. This result was expected from the discussions on the theoretical complexity of these implementations. The memory requirements for the MOM and the FMM are expected to be $O(N^3)$ and $O(N^{3/2})$ respectively, while the processing times are $O(N^3)$ and $O(N^{5/2})$, if $I = O(N)$. The superiority of the FMM over the MOM is more evident as N gets larger.

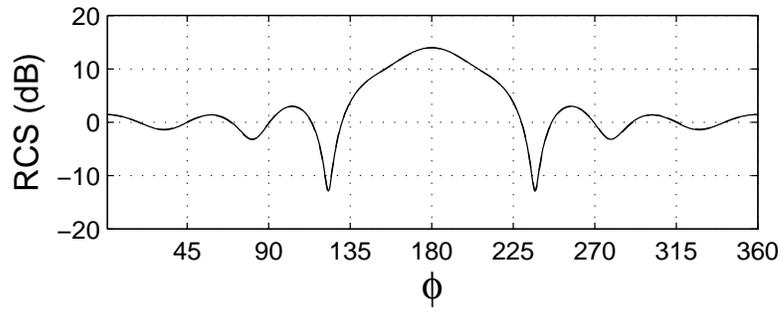
Table 5.1: Comparison of the processing times of the EFIE-MOM and the EFIE-FMM implementations.

Radius of the Sphere	Processing Time for the MOM (sec)	Processing Time for the FMM (sec)
0.5 λ	103	37
0.6 λ	210	82
0.75 λ	567	183
1.0 λ	2211	471
1.5 λ	20511	2676
2.0 λ	NA	10846

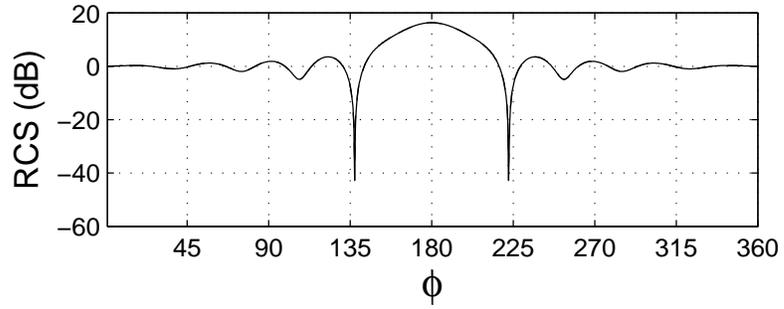
Table 5.2: Comparison of the memory requirements of the EFIE-MOM and the EFIE-FMM implementations.

Radius of the Sphere	Memory Requirement for the MOM (MB)	Memory Requirement for the FMM (MB)
0.5 λ	14	25
0.6 λ	27	37
0.75 λ	68	71
1.0 λ	217	155
1.5 λ	1018	552
2.0 λ	NA	1420

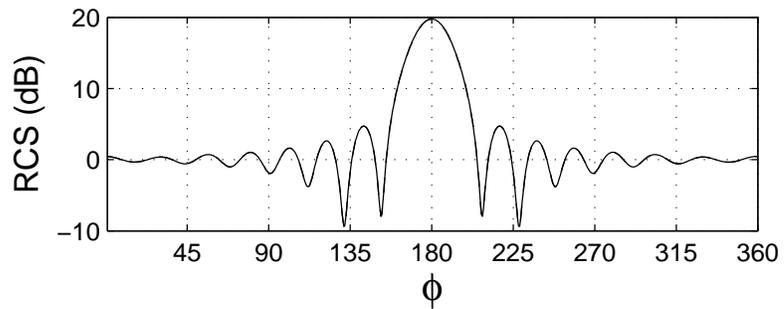
On the other hand, a careful observation reveals that the memory requirement of the EFIE-FMM is not $O(N^{3/2})$, but it is somewhat higher. This problem will be investigated and solved in the next chapter. In general, the EFIE systems are ill-conditioned, especially when they are used for closed surfaces, so that their iterative solutions become difficult. In the solution of the sphere problems, very large preconditioners have to be used in order to reach the result in a reasonable number of iterations. Otherwise, the solution may take very long times to converge or may not converge at all. Attempting to develop and use an optimal preconditioner, which balances the trade-off between the speed of the convergence and memory requirement, may be useful, however, modifying the integral equation to fundamentally change and improve the conditioning of the matrix equation is usually more effective in the pursuit to reach at more efficient solvers.



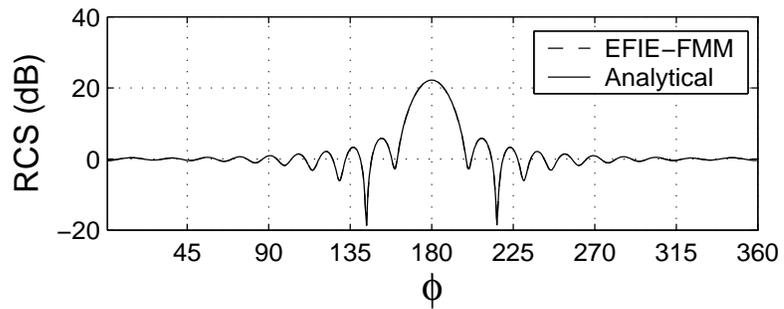
(a)



(b)

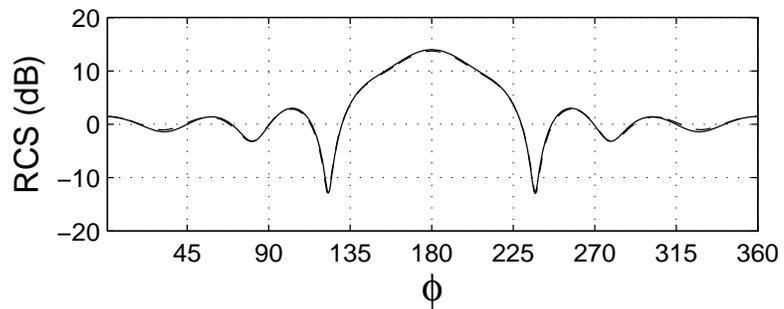


(c)

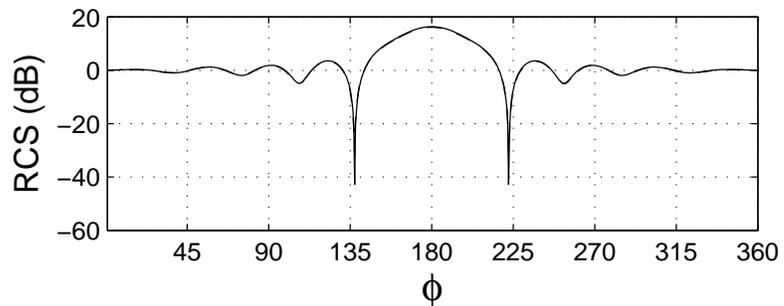


(d)

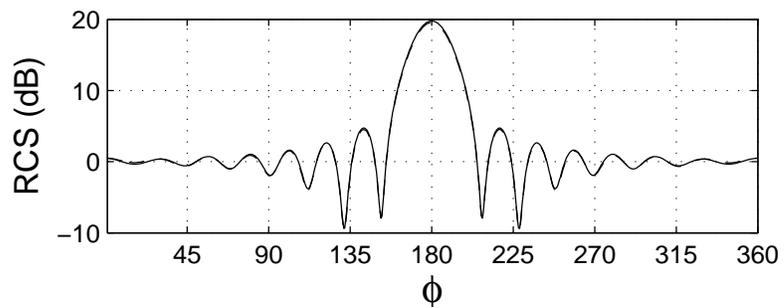
Figure 5.9: RCS values obtained by the EFIE-FMM implementation using the RWG functions and the Galerkin method for a sphere having a radius of (a) 0.75λ , (b) 1.0λ , (c) 1.5λ , and (d) 2.0λ .



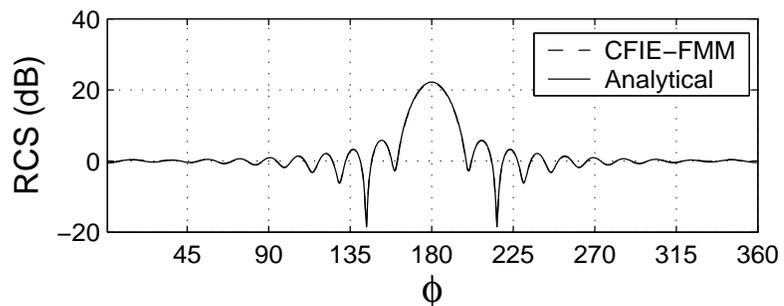
(a)



(b)



(c)



(d)

Figure 5.10: RCS values obtained by the CFIE-FMM implementation using the RWG functions and the Galerkin method for a sphere having a radius of (a) 0.75λ , (b) 1.0λ , (c) 1.5λ , and (d) 2.0λ .

Chapter 6

Iterative Methods and Preconditioning Techniques

The FMM, introduced in the previous chapter, is based on the iterative solution of electromagnetics problems. This chapter introduces the iterative methods as the backbone of the implementations, which use the FMM as a routine to perform the matrix-vector multiplications. These methods can also be used to solve the systems obtained by the application of the MOM and stored in the memory. However, the main discussion in this chapter focuses on the use of the iterative solvers within the framework of the FMM.

The efficiency of the iterative solution affects directly the efficiency of the overall implementation. In this chapter, two types of iterative solvers and various preconditioning techniques will be investigated in detail. In addition, the integral equation used to formulate the problem will be shown to affect the convergence of the solver directly. This will give an idea for the choice of the integral equation whenever more than one of them is available for a problem.

6.1 Iterative Solution of the Linear Systems

To solve the linear systems

$$\sum_{n=1}^N a_n Z_{mn}^{E,M,C} = v_m^{E,M,C}, \quad (6.1)$$

which are obtained by the application of the MOM or the FMM on the integral equations, iterative methods make successive trials to obtain the solution with a given error criteria. In each trial for the solution, it becomes important to determine the new guess in a clever way, in view of the results of the previous guesses. Figure 6.1 sketches the general form of an iterative solution. The iterative algorithm starts with a guess about the solution and it requires this guess to be multiplied by the system matrix. Then, new guesses are formed, and under certain conditions, convergence is accomplished after a number of iterations.

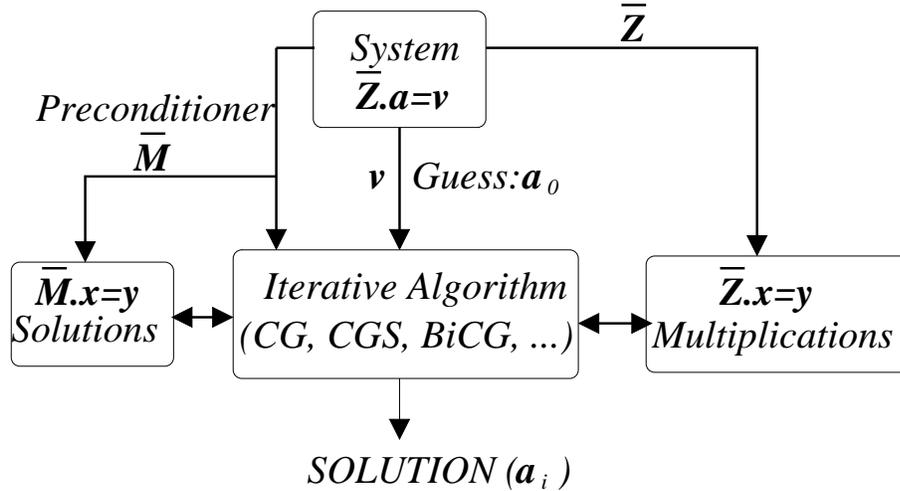


Figure 6.1: General form of an iterative solution.

For many of the iterative algorithms, preconditioners can be implemented to help the solver to reach the solution faster. This technique is based on the transformation of the system into a new system as

$$\overline{M}^{-1} \cdot \overline{Z} \cdot a = \overline{M}^{-1} \cdot v, \quad (6.2)$$

which has the same solution but may be better conditioned. To speed up the convergence of the solver, $\overline{\mathbf{M}}$ should be chosen similar to $\overline{\mathbf{Z}}$, e.g., by copying some of the elements of $\overline{\mathbf{Z}}$ directly into $\overline{\mathbf{M}}$. In the extreme case, $\overline{\mathbf{M}} = \overline{\mathbf{Z}}$ and the iterative solution turns into a direct solution in single iteration. However, $\overline{\mathbf{M}}$ is required to be sparse, having limited number of nonzero elements, in order to avoid large memory requirement and increasing the processing time. Different types of preconditioners and their usages will be investigated in Section 6.2.

There are various iterative algorithms to apply on general linear systems. Two of them will be presented here as the most appropriate ones to work with the FMM in the solution of the scattering problems.

6.1.1 Bi-Conjugate Gradient (BiCG) Algorithm

For complex systems, bi-conjugate gradient (BiCG) algorithm [21] is based on two sequences to be updated as

$$\begin{aligned}\mathbf{r}^{(i)} &= \mathbf{r}^{(i-1)} - c_{(i)} \overline{\mathbf{Z}} \cdot \mathbf{p}^{(i)} \\ \tilde{\mathbf{r}}^{(i)} &= \tilde{\mathbf{r}}^{(i-1)} - c_{(i)} \overline{\mathbf{Z}}^{*T} \cdot \tilde{\mathbf{p}}^{(i)},\end{aligned}\tag{6.3}$$

where “ T ” indicates the transpose and “ $*$ ” indicates the complex conjugate, with the search directions defined as

$$\begin{aligned}\mathbf{p}^{(i)} &= \mathbf{r}^{(i-1)} + d_{(i-1)} \mathbf{p}^{(i-1)} \\ \tilde{\mathbf{p}}^{(i)} &= \tilde{\mathbf{r}}^{(i-1)} + d_{(i-1)} \tilde{\mathbf{p}}^{(i-1)}.\end{aligned}\tag{6.4}$$

In these equations \mathbf{r} , $\tilde{\mathbf{r}}$, \mathbf{p} , and $\tilde{\mathbf{p}}$ are vectors of dimension N , while c and d are constants that can be defined as

$$c_{(i)} = \frac{\langle \tilde{\mathbf{r}}^{(i-1)}, \mathbf{r}^{(i-1)} \rangle}{\langle \tilde{\mathbf{p}}^{(i)}, \overline{\mathbf{Z}} \cdot \mathbf{p}^{(i)} \rangle}\tag{6.5}$$

and

$$d_{(i)} = \frac{\langle \tilde{\mathbf{r}}_{(i)}, \mathbf{r}_{(i)} \rangle}{\langle \tilde{\mathbf{r}}_{(i-1)}, \mathbf{r}_{(i-1)} \rangle}, \quad (6.6)$$

where the inner operator is given as

$$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^{*T} \cdot \mathbf{b}. \quad (6.7)$$

Definition of the constants as in the above ensures the orthogonality relations as

$$\langle \tilde{\mathbf{r}}_{(i)}, \mathbf{r}_{(j)} \rangle = 0 \quad \text{if } i \neq j \quad (6.8)$$

and

$$\langle \tilde{\mathbf{p}}_{(i)}, \mathbf{p}_{(j)} \rangle = 0 \quad \text{if } i \neq j. \quad (6.9)$$

The pseudocode of a general bi-conjugate algorithm for a complex system is listed in Figure 6.2.

Preconditioning in BiCG

The pseudocode in Figure 6.2 indicates the use of a preconditioner as shown on lines 04 and 05. With the preconditioner, the definitions of c_i and d_i become

$$c_{(i)} = \frac{\langle \tilde{\mathbf{r}}_{(i-1)}, \mathbf{z}_{(i-1)} \rangle}{\langle \tilde{\mathbf{p}}_{(i)}, \overline{\mathbf{Z}} \cdot \mathbf{p}_{(i)} \rangle} \quad (6.10)$$

and

$$d_{(i)} = \frac{\langle \tilde{\mathbf{r}}_{(i)}, \mathbf{z}_{(i)} \rangle}{\langle \tilde{\mathbf{r}}_{(i-1)}, \mathbf{z}_{(i-1)} \rangle}, \quad (6.11)$$

where

$$\mathbf{z}_{(i)} = \overline{\mathbf{M}}^{-1} \cdot \mathbf{r}_{(i)}. \quad (6.12)$$

This modification directly comes from the new definition of the residual for the updated system in Equation (6.2). In addition, the definitions in (6.4) are changed

```

01   $\mathbf{r}_{(0)} = \mathbf{v} - \overline{\mathbf{Z}} \cdot \mathbf{x}_{(0)}$  for some initial guess  $\mathbf{x}_{(0)}$ 
02  choose  $\tilde{\mathbf{r}}_{(0)}$ 
03  for  $i = 1, 2, \dots$ 
04      solve  $\overline{\mathbf{M}} \cdot \mathbf{z}_{(i-1)} = \mathbf{r}_{(i-1)}$ 
05      solve  $\overline{\mathbf{M}}^{*T} \tilde{\mathbf{z}}_{(i-1)} = \tilde{\mathbf{r}}_{(i-1)}$ 
06       $\rho_{(i-1)} = \tilde{\mathbf{r}}_{(i-1)}^{*T} \cdot \mathbf{z}_{(i-1)}$ 
07      if  $\rho_{(i-1)} = 0$  then method fails
08      if  $i = 1$  then
09           $\mathbf{p}_{(i)} = \mathbf{z}_{(i-1)}$ 
10           $\tilde{\mathbf{p}}_{(i)} = \tilde{\mathbf{z}}_{(i-1)}$ 
11      else
12           $d_{(i-1)} = \rho_{(i-1)} / \rho_{(i-2)}$ 
13           $\mathbf{p}_{(i)} = \mathbf{z}_{(i-1)} + d_{(i-1)} \mathbf{p}_{(i-1)}$ 
14           $\tilde{\mathbf{p}}_{(i)} = \tilde{\mathbf{z}}_{(i-1)} + d_{(i-1)} \tilde{\mathbf{p}}_{(i-1)}$ 
14      endif
15       $\mathbf{q}_{(i)} = \overline{\mathbf{Z}} \cdot \mathbf{p}_{(i)}$ 
16       $\tilde{\mathbf{q}}_{(i)} = \overline{\mathbf{Z}}^{*T} \cdot \tilde{\mathbf{p}}_{(i)}$ 
17       $c_i = \rho_{(i-1)} / \tilde{\mathbf{p}}_{(i)}^{*T} \cdot \mathbf{q}_{(i)}$ 
18       $\mathbf{x}_{(i)} = \mathbf{x}_{(i-1)} + c_i \mathbf{p}_{(i)}$ 
19       $\mathbf{r}_{(i)} = \mathbf{r}_{(i-1)} - c_i \mathbf{q}_{(i)}$ 
20       $\tilde{\mathbf{r}}_{(i)} = \tilde{\mathbf{r}}_{(i-1)} - c_i \tilde{\mathbf{q}}_{(i)}$ 
21      check the residual error  $\mathbf{r}_{(i)}$ 
22  end
23  return  $\mathbf{x}_{(i)}$ 

```

Figure 6.2: Pseudocode of a general BiCG algorithm for complex systems.

as

$$\begin{aligned}
 \mathbf{p}_{(i)} &= \mathbf{z}_{(i-1)} + d_{(i-1)} \mathbf{p}_{(i-1)} \\
 \tilde{\mathbf{p}}_{(i)} &= \tilde{\mathbf{z}}_{(i-1)} + d_{(i-1)} \tilde{\mathbf{p}}_{(i-1)}.
 \end{aligned} \tag{6.13}$$

It should be noted that the BiCG algorithm requires two solutions of the preconditioner system for each iteration.

Solution

The new guess for the solution of the problem is generated on line 18 of the pseudocode in Figure 6.2. At the end of each iteration loop, the residual error is

compared with the desired threshold for the error. If the convergence is satisfied, the final value of \mathbf{x}_i is returned as the solution of the system.

Matrix-Vector Multiplication

Lines 15 and 16 of the pseudocode show the matrix-vector multiplications used in the solution. These operations are performed by the FMM, and the BiCG algorithm requires two calls of the FMM in each iteration.

Choosing the Initial Values

The initial choice for the residual, $\mathbf{r}_{(0)}$ can be arbitrary due to the arbitrary choice of the initial guess $\mathbf{x}_{(0)}$, but it will be shown in Section 6.4 that small initial residual leads to convergence in fewer iterations even though the rate of the decay of the residual is unchanged. It is possible to choose the definition of $\tilde{\mathbf{r}}_{(0)}$, which affects directly the efficiency of the solver, in various ways. Two of these choices are

$$\tilde{\mathbf{r}}_{(0)} = \overline{\mathbf{Z}} \cdot \mathbf{r}_{(0)} \quad (6.14)$$

and

$$\tilde{\mathbf{r}}_{(0)} = \mathbf{r}_{(0)}^*. \quad (6.15)$$

The second choice results in a significant improvement of the algorithm for the solution of the complex symmetric systems [22]. If the system matrix is symmetric so that

$$\overline{\mathbf{Z}}^T = \overline{\mathbf{Z}}, \quad (6.16)$$

then this choice simplifies the algorithm in Figure 6.2 into a reduced form as shown in Figure 6.3. It should be noted that this reduced form requires only one solution of the preconditioner system and one matrix-vector multiplication for each iteration. This efficiency improvement is only applicable to systems obtained with the EFIE. The systems employing the MFIE and the CFIE are not symmetric so that they cannot use the reduced form of the BiCG algorithm.

```

01    $\mathbf{r}_{(0)} = \mathbf{v} - \overline{\mathbf{Z}} \cdot \mathbf{x}_{(0)}$  for some initial guess  $\mathbf{x}_{(0)}$ 
02   for  $i = 1, 2, \dots$ 
03       solve  $\overline{\mathbf{M}} \cdot \mathbf{z}_{(i-1)} = \mathbf{r}_{(i-1)}$ 
04        $\rho_{(i-1)} = \mathbf{r}_{(i-1)}^T \cdot \mathbf{z}_{(i-1)}$ 
05       if  $\rho_{(i-1)} = 0$  then method fails
06       if  $i = 1$  then
07            $\mathbf{p}_{(i)} = \mathbf{z}_{(i-1)}$ 
08       else
09            $d_{(i-1)} = \rho_{(i-1)} / \rho_{(i-2)}$ 
10            $\mathbf{p}_{(i)} = \mathbf{z}_{(i-1)} + d_{(i-1)}\mathbf{p}_{(i-1)}$ 
11       endif
12        $\mathbf{q}_{(i)} = \overline{\mathbf{Z}} \cdot \mathbf{p}_{(i)}$ 
13        $c_i = \rho_{(i-1)} / \mathbf{p}_{(i)}^T \cdot \mathbf{q}_{(i)}$ 
14        $\mathbf{x}_{(i)} = \mathbf{x}_{(i-1)} + c_i\mathbf{p}_{(i)}$ 
15        $\mathbf{r}_{(i)} = \mathbf{r}_{(i-1)} - c_i\mathbf{q}_{(i)}$ 
16       check the residual error  $\mathbf{r}_{(i)}$ 
17   end
18   return  $\mathbf{x}_{(i)}$ 

```

Figure 6.3: Pseudocode of the reduced form of the BiCG algorithm for complex symmetric systems.

6.1.2 Conjugate Gradient Squared (CGS) Algorithm

In the BiCG method, the residual sequence can be interpreted as

$$\mathbf{r}_{(i)} = P_i(\overline{\mathbf{Z}}) \cdot \mathbf{r}_{(0)}, \quad (6.17)$$

where P_i is an i^{th} degree polynomial, which summarizes the process from the initial residual to the residual at the i^{th} step. Conjugate gradient squared (CGS) algorithm uses this polynomial twice as

$$\mathbf{r}_{(i)} = P_i^2(\overline{\mathbf{Z}}) \cdot \mathbf{r}_{(0)}, \quad (6.18)$$

which might result in faster convergence to the solution. A general algorithm for the CGS is given in Figure 6.4. This algorithm also requires two solutions of the preconditioner system and two matrix-vector multiplications for each iteration.

Although the CGS algorithm is expected to have faster convergence than the BiCG algorithm [21, 23], this is not observed in the solution of scattering problems. The comparison on the efficiency of the algorithms will be performed in the following section.

```

01    $\mathbf{r}_{(0)} = \mathbf{v} - \overline{\mathbf{Z}} \cdot \mathbf{x}_{(0)}$  for some initial guess  $\mathbf{x}_{(0)}$ 
02   choose  $\tilde{\mathbf{r}}_{(0)}$  (usually  $\tilde{\mathbf{r}}_{(0)} = \mathbf{r}_{(0)}$ )
03   for  $i = 1, 2, \dots$ 
04        $\rho_{(i-1)} = \tilde{\mathbf{r}}_{(i-1)}^{*T} \cdot \mathbf{r}_{(i-1)}$ 
05       if  $\rho_{(i-1)} = 0$  then method fails
06       if  $i = 1$  then
07            $\mathbf{u}_{(i)} = \mathbf{r}_{(i-1)}$ 
08            $\mathbf{p}_{(i)} = \mathbf{r}_{(i-1)}$ 
09       else
10            $d_{(i)} = \rho_{(i-1)} / \rho_{(i-2)}$ 
11            $\mathbf{u}_{(i)} = \mathbf{r}_{(i-1)} + d_{(i-1)} \mathbf{q}_{(i-1)}$ 
12            $\mathbf{p}_{(i)} = \mathbf{r}_{(i-1)} + 2d_{(i-1)} \mathbf{q}_{(i-1)} + d_{(i-1)} d_{(i-1)} \mathbf{p}_{(i-1)}$ 
13       endif
14       solve  $\overline{\mathbf{M}} \cdot \tilde{\mathbf{p}} = \mathbf{p}_{(i)}$ 
15        $\tilde{\mathbf{v}} = \overline{\mathbf{Z}} \cdot \tilde{\mathbf{p}}$ 
16        $c_{(i)} = \rho_{(i-1)} / \tilde{\mathbf{r}}_{(i)}^{*T} \cdot \tilde{\mathbf{v}}$ 
17        $\mathbf{q}_{(i)} = \mathbf{u}_{(i)} - c_{(i)} \tilde{\mathbf{v}}$ 
18       solve  $\mathbf{M} \cdot \tilde{\mathbf{u}} = \mathbf{u}_{(i)} + \mathbf{q}_{(i)}$ 
19        $\mathbf{x}_{(i)} = \mathbf{x}_{(i-1)} + c_{(i)} \tilde{\mathbf{u}}$ 
20        $\tilde{\mathbf{q}} = \overline{\mathbf{Z}} \cdot \tilde{\mathbf{u}}$ 
21        $\mathbf{r}_{(i)} = \mathbf{r}_{(i-1)} - c_{(i)} \tilde{\mathbf{q}}$ 
22       check the residual error  $\mathbf{r}_{(i)}$ 
23   end
24   return  $\mathbf{x}_{(i)}$ 

```

Figure 6.4: Pseudocode of the general CGS algorithm for complex systems.

6.2 Preconditioning Techniques

In this section, various types of preconditioners will be outlined. In general, preconditioner matrices have to be similar to the impedance matrix produced by the integral equation of choice. With large preconditioners, which retain many

of the impedance-matrix elements, the modified, i.e., preconditioned, system in Equation (6.2) becomes easier to solve. However, this leads to an increase in the memory requirement due to the need to store the elements of the preconditioner. In addition, the inverse of the preconditioner has to be evaluated, which requires longer time for large preconditioners. It should be noted that the extreme case occurs when the preconditioner retains all of the elements of the impedance matrix. However, taking the inverse of this matrix to use as a preconditioner in the iterative solver is not different from the direct solution of the system. Thus, the trade-off in the use of preconditioners has to be handled carefully to avoid suppressing the efficiency of the FMM.

6.2.1 Types of Preconditioners

An iterative algorithm uses the preconditioner by the request of the operation as

$$\mathbf{y} = \overline{\mathbf{M}}^{-1} \cdot \mathbf{x}, \quad (6.19)$$

where \mathbf{x} changes for each call of the operation and \mathbf{y} is the corresponding solution of the preconditioner system. Different types of preconditioners can be implemented.

No Preconditioner

Iterative algorithms can be used without preconditioners. This can be shown as

$$\mathbf{y} = \overline{\mathbf{M}}^{-1} \cdot \mathbf{x} = \mathbf{x} \quad (6.20)$$

and it can be interpreted as $\overline{\mathbf{M}} = \overline{\mathbf{I}}$.

Diagonal Preconditioner

When the preconditioner has only the diagonal elements of the impedance matrix,

$$\begin{aligned}\overline{\mathbf{M}}^{-1}(m, m) &= 1/\overline{\mathbf{Z}}(m, m) \\ \overline{\mathbf{M}}^{-1}(m, n) &= 0, \quad \text{if } m \neq n\end{aligned}\tag{6.21}$$

and

$$\mathbf{y}(m) = \sum_{n=1}^N \overline{\mathbf{M}}^{-1}(m, n) \mathbf{x}(n) = \mathbf{x}(m) / \overline{\mathbf{Z}}(m, m).\tag{6.22}$$

The memory required of this type of preconditioner is $O(N)$, while the inverse operation requires $O(N)$ FLOPs. At each iteration, the solution of the preconditioner system also requires $O(N)$ FLOPs. Overall, the memory usage and the processing time required to implement the preconditioner appears to be $O(N)$ and $O(N) + O(IN) = O(IN)$, respectively, which do not disturb the efficiency of the FMM.

Block-Diagonal Preconditioners

This type of preconditioners are formed by choosing the interactions between the basis and testing functions in the same cluster. Since these interactions are among the functions in the near-field of each other, the chosen elements of the impedance matrix are expected to be stronger than the other interactions. On the other hand, the distribution of self-cluster elements may not originally have the block-diagonal form in the impedance matrix. This is because of the fact that the clustering of the functions are performed according to the spatial locations of the functions. Therefore, a cluster may include interactions at various locations in the impedance matrix.

Figure 6.5 shows the self-cluster interactions in an impedance matrix of a 72-unknown problem. These elements, which are distributed randomly, are desired

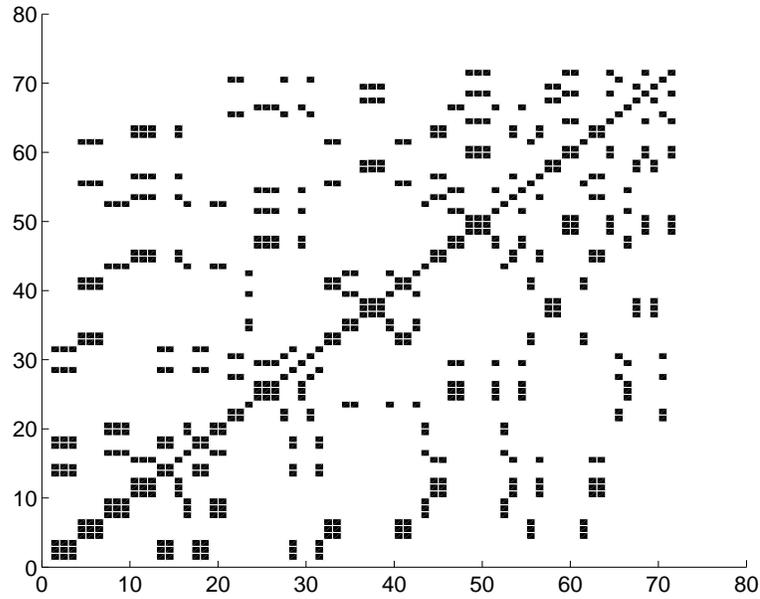


Figure 6.5: Self-cluster interactions in the impedance matrix of an 72-unknown problem.

to be used in the preconditioner. At this stage, a renumbering of the matrix can be useful so that the new preconditioner appears as in Figure 6.6, which is block diagonal. According to the renumbering in the preconditioner, the input vector \mathbf{x} and the resulting vector \mathbf{y} should also be rearranged. All of these operations can be summarized as

$$\mathbf{y} = [(\overline{\mathbf{M}}^R)^{-1} \cdot \mathbf{x}^R]^{R'}, \quad (6.23)$$

where R represents the renumbering operation and R' is the reverse of this operation. In the implementation, the renumbering may not be processed explicitly, but a mapping technique can be used in order to carry out the indexing.

The inverse of a block-diagonal matrix can be efficiently evaluated by considering the inverse of blocks separately. For each block of size of $\sqrt{N} \times \sqrt{N}$, the memory requirement is $O(N)$ and the inverse operation is done with $O(N^{3/2})$

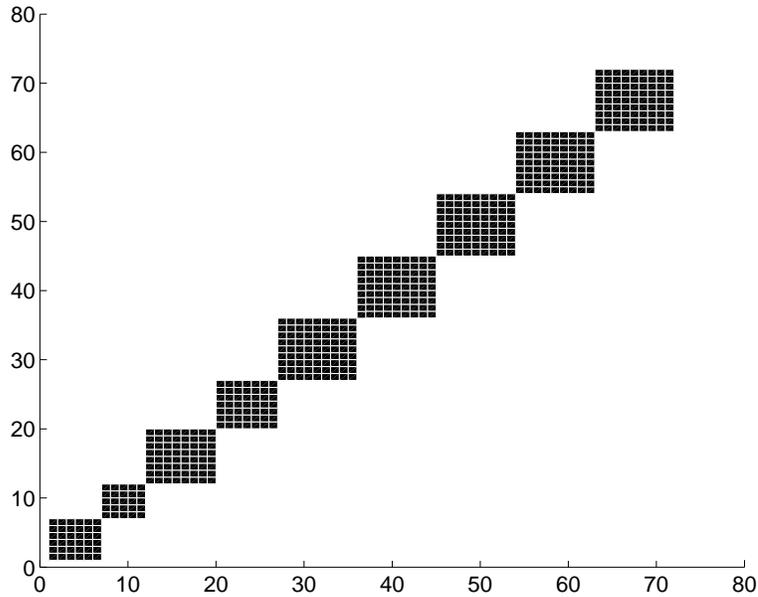


Figure 6.6: Self-cluster interactions in the impedance matrix of an 72-unknown problem after the renumbering operation.

FLOPs by direct methods. Then, for the entire factorization of the preconditioner, the number of FLOPs becomes $O(N^2)$. However, this process is performed only once before the iterative solution. The use of the preconditioner in each iteration requires $O(N^{3/2})$ FLOPs. Therefore, the overall processing time for the block-diagonal preconditioner becomes $O(N^2) + O(IN^{3/2}) = O(IN^{3/2})$, assuming $I = O(N)$. This order is the same as the complexity of matrix-vector multiplication accomplished by the FMM, while the memory requirement is again $O(N^{3/2})$.

Filtered Preconditioners

These preconditioners are formed by selectively retaining some of the elements of the impedance matrix [24]. The selection is done according to the magnitudes of the elements so that the strongest elements of the impedance matrix are chosen. By this way, the quality of the preconditioner matrix becomes high. The number of elements selected to the preconditioner can be adjusted according the memory

requirements. Therefore, the size of the filtered preconditioners varies in a wide range.

Since the distribution and the number of the elements are arbitrary, it becomes practically impossible to arrange the elements in block-diagonal or other convenient forms. This brings about the difficulty in the factorization and evaluation of the inverse matrix. Various methods on sparse matrices can be used to factorize these preconditioners to use in the iterative solver.

6.2.2 Comparisons of Preconditioners and Iterative Solvers

Figure 6.7 shows the iteration plots for different types of preconditioners used in the iterative solution of two different sphere problems. The problems are the second and fourth models shown in Figure 4.11, which correspond to 1302 and 8364 unknowns, respectively. The EFIE-FMM implementation with RWG functions and the Galerkin method are used while both CGS and BiCG are applied as the iterative solver.

Following observations can be done on the results:

1. BiCG has always faster convergence than the CGS algorithm for any type of the preconditioners, although both solvers start with the same initial guess for the solution. Especially, for the small preconditioners, such as the diagonal or the block-diagonal, the CGS algorithm does not show any convergence within a reasonable number of iterations.
2. As the number of nonzero elements is increased in the preconditioner, both solvers tend to have faster convergence. In the bottom plots for both problems, all of the near-field interactions are used in the filtered preconditioner, leading to the fastest convergence as expected.

3. Although the block-diagonal preconditioner has more nonzero elements than the diagonal preconditioner, it does not have better convergence. This is mainly due to the characteristics of the systems obtained by the EFIE. In general, it is observed that the self-cluster interactions in the impedance matrix of the EFIE are not the only strong elements in the matrix. Therefore, selecting these elements into the preconditioner while discarding some other strong ones leads to a worse performance in the convergence. For such systems, diagonal preconditioners seem to perform better.
4. In spite of the improved convergence with the improved preconditioners, the overall performance of the iterative solvers with the EFIE is not satisfactory, even for the smaller problem. Only the large preconditioners obtained by the filtering technique give rise to acceptable convergence, which makes the overall implementation inefficient due to the increased difficulty in the factorization and storage of the preconditioner.
5. It can be observed that the increase in the size of the problem gives rise to increase in the number of iterations. It is obvious that the size of the filtered preconditioner has to be increased significantly to keep the number of iterations low for larger problems.

6.3 Comparisons of the Integral Equations

As shown in the previous section, the EFIE systems are usually ill-conditioned and not suitable for the iterative solutions. As the problem gets larger, better and better preconditioners are required to obtain a convergence. Then, the use of the preconditioner becomes the most critical cost of the implementation.

Systems obtained by the MFIE formulation are generally better conditioned than the EFIE systems. The CFIE systems are even better than the MFIE systems if

the constant α in (2.25) is chosen appropriately. Then, it becomes possible to use efficient preconditioners. This is the main reason for these two formulations to be more preferable for the scattering problems involving geometries with closed surfaces.

Figure 6.8 shows the iteration plots for three formulations used in the solution of sphere problems of different sizes. For the CFIE formulation, α is chosen as 0.2, which will be shown to be an appropriate choice. The iterative solver implements the CGS algorithm employing diagonal and block-diagonal preconditioners. Following observations can be done on these results:

1. Different from the EFIE, both the MFIE and the CFIE formulations lead to good convergence even with the diagonal preconditioner. In addition, the convergence is improved as the preconditioner is changed from diagonal to block-diagonal. It seems that the use of block-diagonal preconditioner becomes applicable for the MFIE and the CFIE formulations. This is due to the fact that the strongest elements in the MFIE and the CFIE matrices are located in the self-cluster blocks. These elements, which are the interactions between the close functions, are stronger than the other elements in the impedance matrix.
2. As the problem size gets larger, the increase in the number of iterations is very limited for the CFIE. Therefore, to solve larger problems, it becomes important to use the CFIE. It can be surprising to use a linear combination of two equations and duplicating the operations, instead of using a single equation with less work. However, for large problems, the cost of the preconditioner becomes the bottleneck for the EFIE systems. The MFIE is better than the EFIE, but it also exhibits faster increase in the number of iterations compared to the CFIE, as the problem size gets larger. As a result, the CFIE seems to be better than the other two equations for the

solution of large scattering problems.

6.3.1 Efficiency of the FMM for Different Formulations

As it was shown in Figure 6.7, the EFIE systems can be forced to converge faster by using better preconditioners. However, this reduces the efficiency of the FMM implementation as shown in Tables 6.1 and 6.2, where the processing time and the memory requirements are given in detail for the solution of the sphere problems shown in Figure 4.11.

Table 6.1: Processing time of the EFIE-FMM implementation.

Radius of the Sphere	Total Processing Time (sec)	Solution Time (sec)	Number of Iterations
0.5 λ	37	9	29
0.6 λ	82	34	72
0.75 λ	183	58	61
1.0 λ	471	160	75
1.5 λ	2676	1278	198
2.0 λ	10846	6049	366

Table 6.2: Memory requirement of the EFIE-FMM implementation.

Radius of the Sphere	Total Memory Requirement (MB)	Memory Requirement for the Preconditioner (MB)
0.5 λ	25	7
0.6 λ	37	12
0.75 λ	71	28
1.0 λ	155	72
1.5 λ	552	332
2.0 λ	1420	970

To solve the problems, large preconditioners, obtained by the filtering technique, are used. As shown in Table 6.2, these preconditioners use very large portion of the total memory required by the implementation. This portion grows as the

problem size increases and the preconditioner becomes the bottleneck for the memory usage.

Table 6.1 shows that the processing time for the solution becomes more significant as the problem size gets larger. In addition, the speed of the increase is much larger than the speed of the increase in the number of iterations. This is the direct consequence of using large preconditioners since their use in the iterative algorithm becomes more costly. It is interesting to note that the increase in the number of iterations cannot be avoided despite the significant growth of the preconditioner.

It is evident from the two tables that the EFIE formulation is not convenient to solve large problems. As the problem size increases, its usage becomes more difficult in terms of the computational sources. As it was previously stated, the CFIE formulation gives better conditioned systems, which does not suffer from the problems encountered on the EFIE.

To validate the superiority of the CFIE, the Tables 6.3 and 6.4 show the processing time and memory requirements for the same problems. With the CFIE formulation, it becomes possible to use block-diagonal preconditioners. The decrease in the total memory requirement due to the reduced preconditioner is fascinating. As Table 6.3 shows, the total processing time also drops with the steep decline of the solution time. As the problem size increases, the block-diagonal preconditioner becomes smaller as compared to the impedance matrix. However, the speed of the increase in the number of iterations is still very slow. This shows the suitability of the CFIE for the solution of larger problems.

Table 6.3: Processing time of the CFIE-FMM implementation.

Radius of the Sphere	Total Processing Time (sec)	Solution Time (sec)	Number of Iterations
0.5 λ	58	5	13
0.6 λ	95	6	13
0.75 λ	226	15	15
1.0 λ	531	24	14
1.5 λ	1853	99	17
2.0 λ	3991	169	15

Table 6.4: Memory requirement of the CFIE-FMM implementation.

Radius of the Sphere	Total Memory Requirement (MB)	Memory Requirement for the Preconditioner (MB)
0.5 λ	22	0.4
0.6 λ	31	0.7
0.75 λ	52	1.4
1.0 λ	101	3.5
1.5 λ	260	11
2.0 λ	511	29

6.3.2 Choice of α in the CFIE

Figure 6.9 shows the number of iterations with respect to α in the the CFIE formulation to reach various residual errors in the solution of the sphere problems. It can be observed that the curves have their minimum for α values about 0.2-0.3. In general, the the CFIE systems are better conditioned at these optimum values of α , so that the convergence in the iterative solution is achieved with less number of iterations.

6.4 Effect of Initial Guess on the Convergence

In Figure 6.10, the iteration plots are given for 1302-unknown and 8364-unknown problems formulated by the EFIE and solved by the BiCG algorithm employing

different types of preconditioners. Improved initial guesses are used for these problems in addition to the ordinary initial guesses. For the smaller problem solved at 600 MHz, the solution obtained at 550 MHz is used as the improved initial guess, while the solution at 1450 MHz is used for the larger problem to be solved at 1500 MHz. It can be seen that the number of iterations drops with the improved initial guess for the solution. However, the speed of the convergence seems to be the same.

In many problems, the solution is required in a range of frequencies. This brings about the usage of a solution at a frequency as an initial guess for the solution in the next frequency. For small intervals of frequency, a speed-up can be obtained, leading to a significant reduction in the total time.

6.5 Multiple Excitations

With the use of efficient preconditioners, the bottleneck of the FMM implementations becomes the near-field interactions. In the next chapter, the MLFMA is introduced, which further reduces the number of near-field interactions. This algorithm also reduces the memory requirement and the processing time per iteration down to $O(N \log N)$.

If the solution of the problem is desired for multiple excitations, the near-field interactions can be computed only once, before the beginning of the iterative algorithm. Then, the iterative algorithm can be started for each excitation vector and the same near-field data stored in the memory can be used for many times. By this way, the processing time can be reduced significantly.

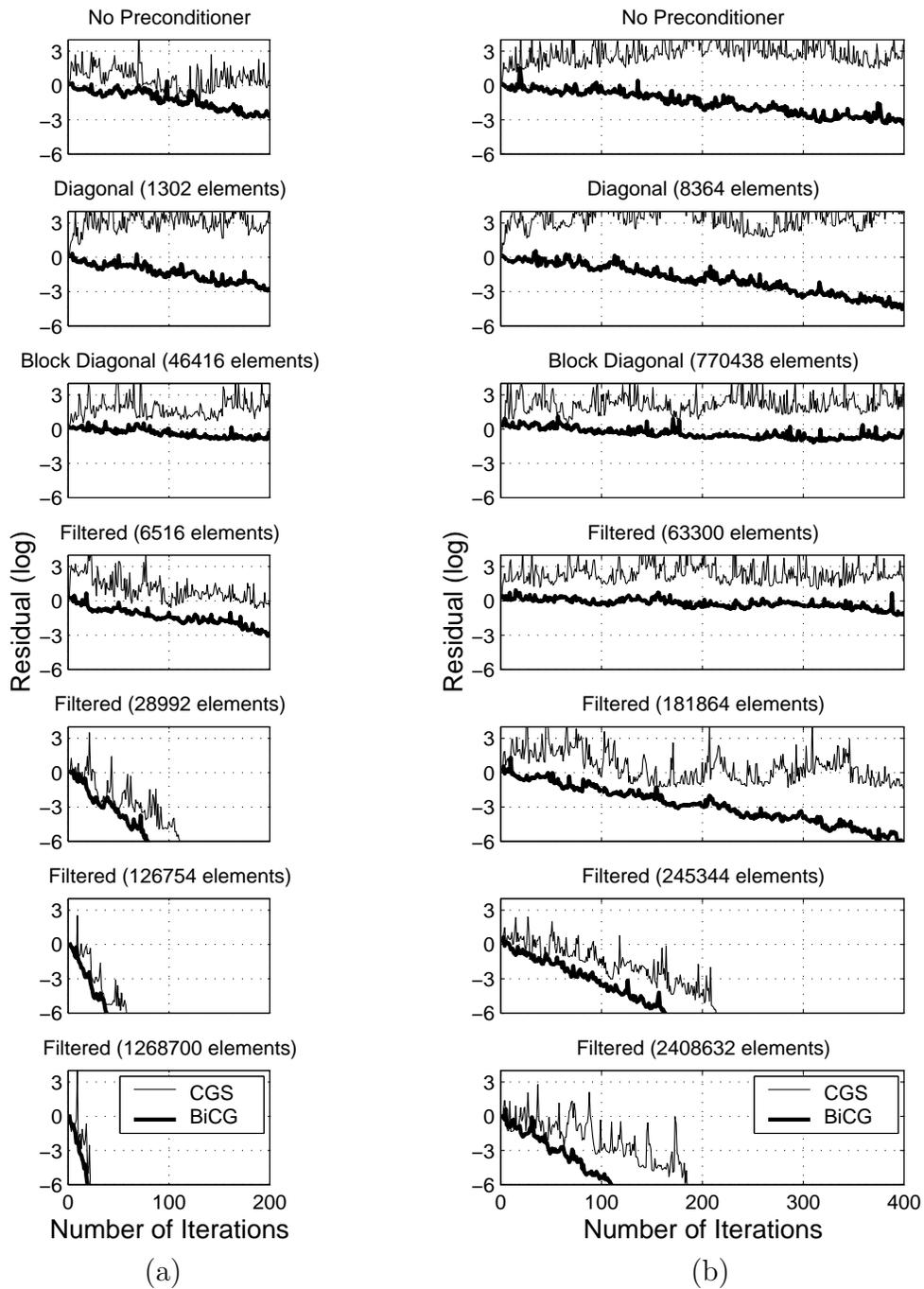


Figure 6.7: Plots of the residual versus the iteration number for different types of preconditioners implemented with CGS and BiCG for the solution of (a) 1302-unknown, and (b) 8364-unknown scattering problems of the sphere shown in Figure 4.11. The EFIE is used to formulate the problems.

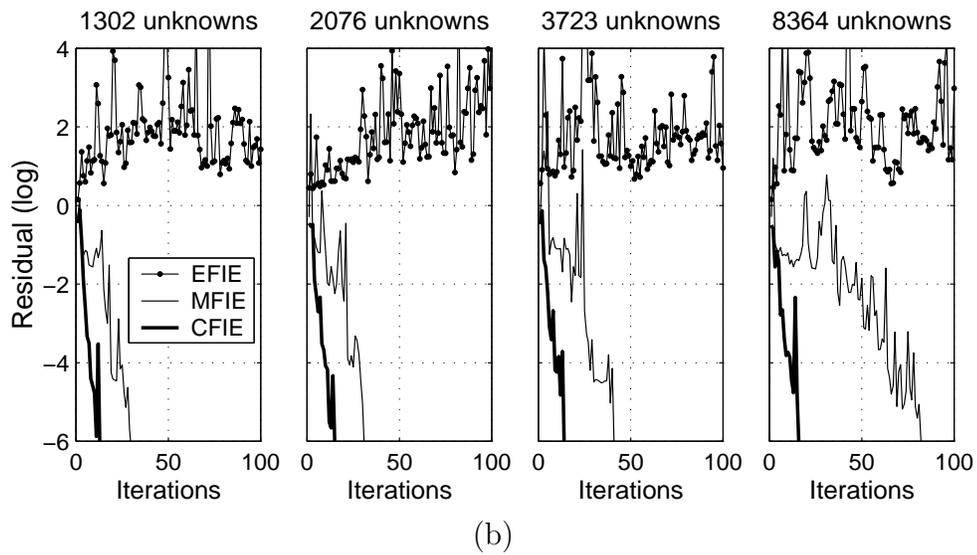
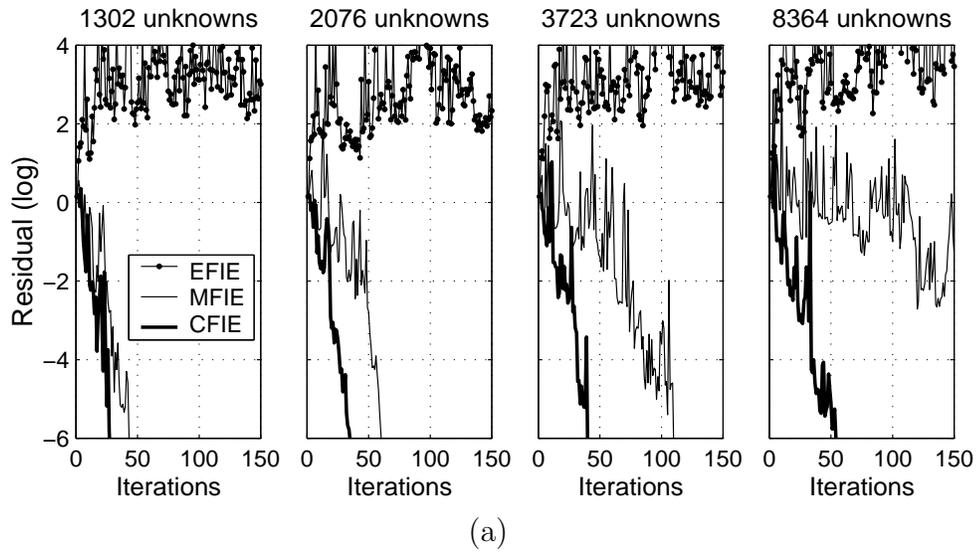
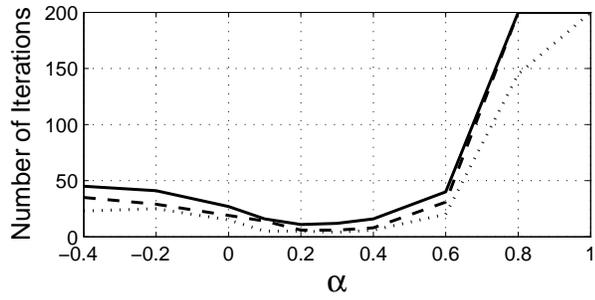
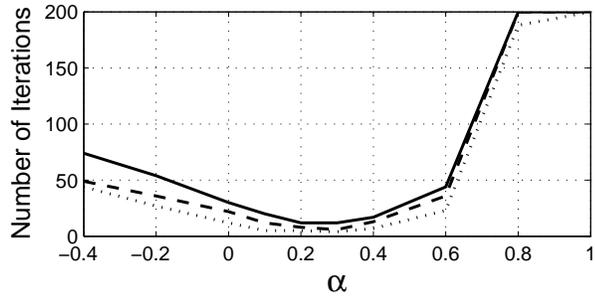


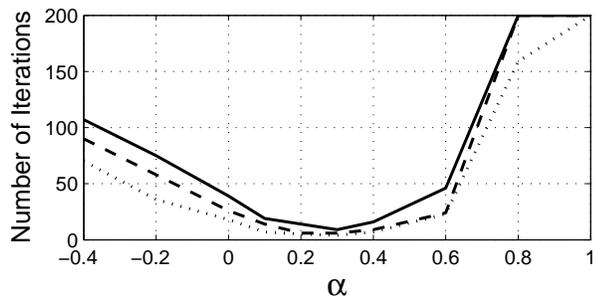
Figure 6.8: Plots of the residual versus the iteration number for the solution of sphere problems shown in Figure 4.11 using different integral-equation formulations with CGS employing (a) diagonal and (b) block-diagonal preconditioners.



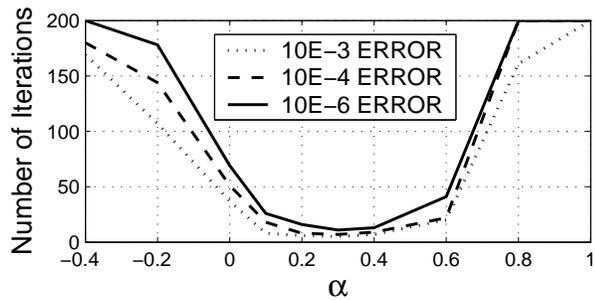
(a)



(b)



(c)



(d)

Figure 6.9: Number of iterations with respect to α to reach various residual errors for the sphere problems having (a) 1302 unknowns, (b) 2076 unknowns, (c) 3723 unknowns, and (d) 8364 unknowns.

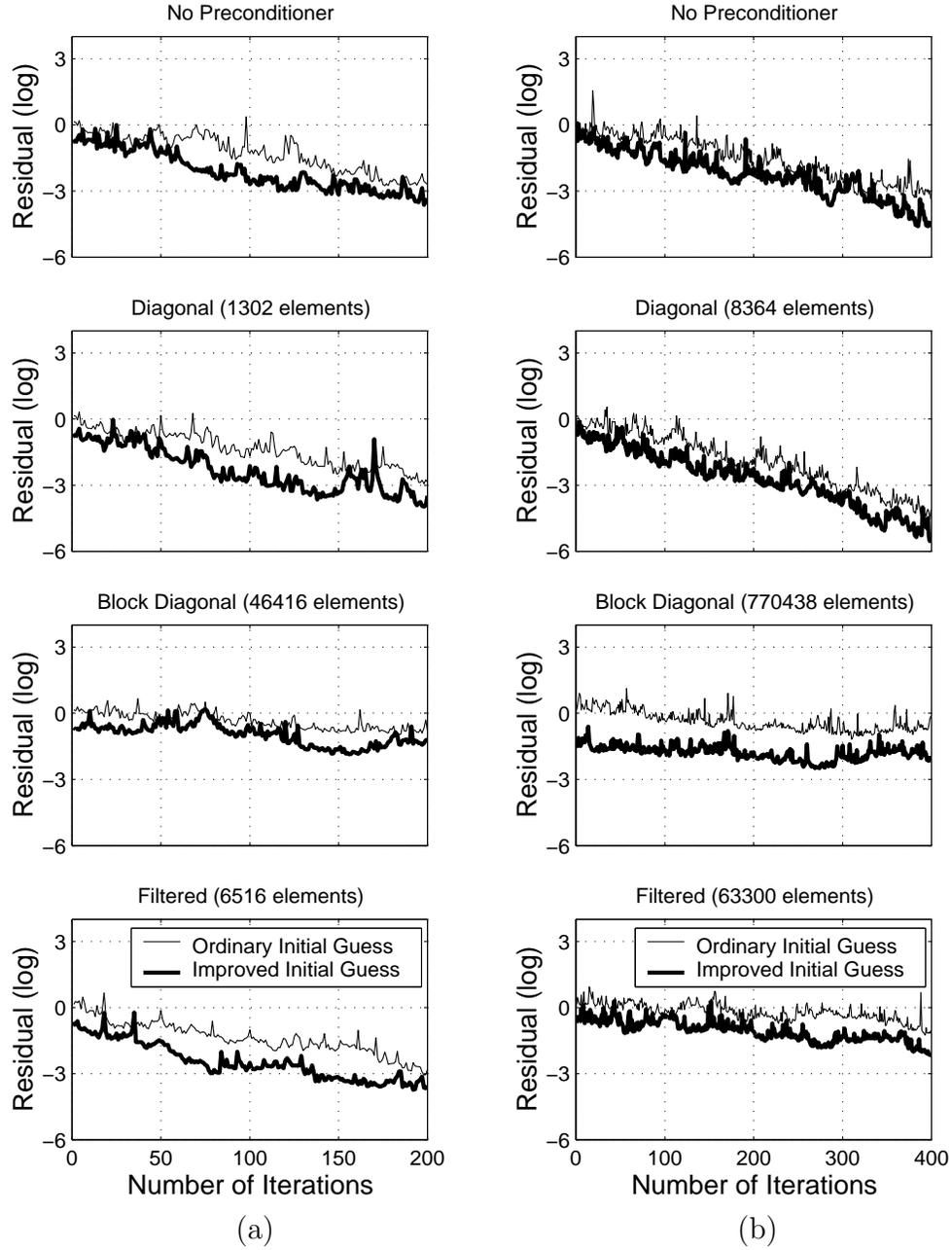


Figure 6.10: Effect of the improved initial guess on the convergence for different types of preconditioners implemented with BiCG for the solution of (a) 1302-unknown and (b) 8364-unknown scattering problems of the sphere shown in Figure 4.11.

Chapter 7

MLFMA Implementations

The generalization of the two-level FMM is the MLFMA [4]–[6], which is based on grouping the groups in the FMM. This method further reduces the memory requirement to $O(N \log N)$, while the processing time also becomes $O(N \log N)$ per iteration. Therefore, some large problems, which cannot be solved by a two-level FMM implementation with the existing computational resources, can be solved by the MLFMA. This chapter introduces this method in detail.

The aggregation, translation, and disaggregation concepts of the FMM are the same in the MLFMA. However, the reduction of the complexity to $O(N \log N)$ requires the interpolation and antinterpolation operations during the aggregation and disaggregation processes. This brings about extra design considerations on the implementation of the MLFMA. In addition, the fourth error source is introduced in these operations, which should be carefully controlled to balance the efficiency and the accuracy.

7.1 Evaluation of Interactions with the MLFMA

The usage of the MLFMA in the evaluation of the interactions appears in the matrix-vector products required by the iterative solver. The overall process can be divided into three steps.

7.1.1 Aggregation in the MLFMA

The aggregation process in the two-level FMM can be seen as the part of Equation 5.23 and the resulting function at the center of cluster C' can be written as

$$\mathbf{F}^{C'}(\hat{\mathbf{k}}) = \sum_{n \in C'} \mathbf{F}_{f_{n C'}}(\hat{\mathbf{k}}) a_n, \quad (7.1)$$

where n represents the basis function with a coefficient a_n assigned by the iterative solver. This summation represents only the aggregation process for the cluster C' and has to be performed for all other basis clusters to complete the aggregation in the two-level FMM. In the MLFMA, the aggregation process continues from the lowest level to the highest level.

Figure 7.1 shows the lowest level clusters, which contain the basis functions. After the aggregations at the centers of these clusters, a second aggregation continues towards the center of the parent cluster. This parent cluster belongs to the higher level, which may also be inside a larger cluster. The overall process continues up to the aggregations in the highest level.

It can be recalled from Chapter 5 that the number of points taken on the unit sphere for the angular integration depends on the number of harmonics used in the translation as

$$K = 2L^2. \quad (7.2)$$

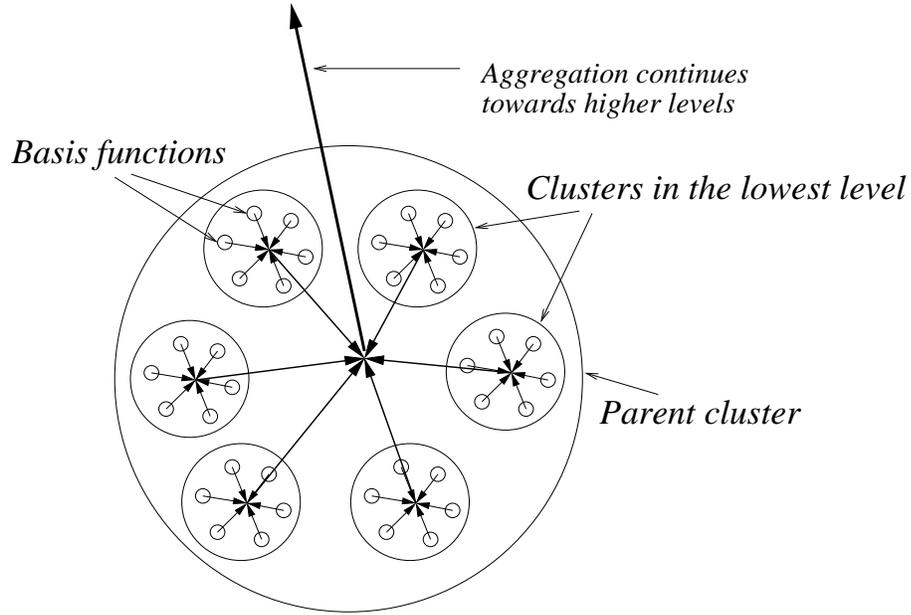


Figure 7.1: Aggregation process in the MLFMA.

In addition, L is proportional to the radius of the clusters involved in the translation, so that

$$K \propto (kd)^2. \quad (7.3)$$

Therefore, the number of angular samples for the radiation of a cluster depends on the size of the cluster. The clusters in the lower levels use coarser samplings, while the clusters in the upper levels require finer samplings. This brings about a disagreement between a cluster and its parent cluster since they use different numbers of samples. Therefore, as the aggregation process continues after the lowest level, interpolations have to be done before adding the contributions from clusters to their parent clusters.

Interpolation

Figure 7.2 shows the fine and coarse sampling grids on the surface of the unit sphere. By interpolation, the value of the function \mathbf{F} is desired to be found on

the sampling points of the fine grid. Various interpolation methods can be applied. However, local application of the Lagrange interpolation will be shown here, which is known to give good results with high efficiency.

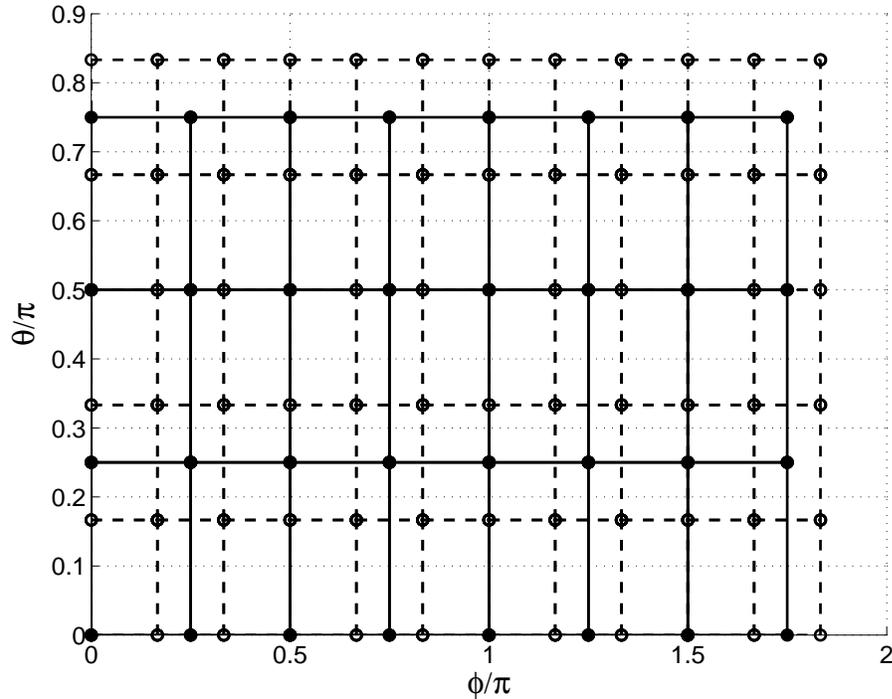


Figure 7.2: Fine and coarse grids on the unit sphere.

As shown in Figure 7.3, the Lagrange interpolation uses a predetermined number of points to calculate the value of the function at the target point, (θ_t, ϕ_t) . The source points on the coarse grid are chosen to be the nearest points to the target point on the finer grid. The 2-D interpolation starts with 1-D interpolation in the θ direction as

$$\hat{\mathbf{F}}(\theta_t, \phi_k) = \sum_{i=1}^m l_i(\theta_t) \mathbf{F}(\theta_i, \phi_k), \quad (7.4)$$

where

$$l_i(\theta_t) = \prod_{j \neq i} \frac{\theta_t - \theta_j}{\theta_i - \theta_j} \quad (7.5)$$

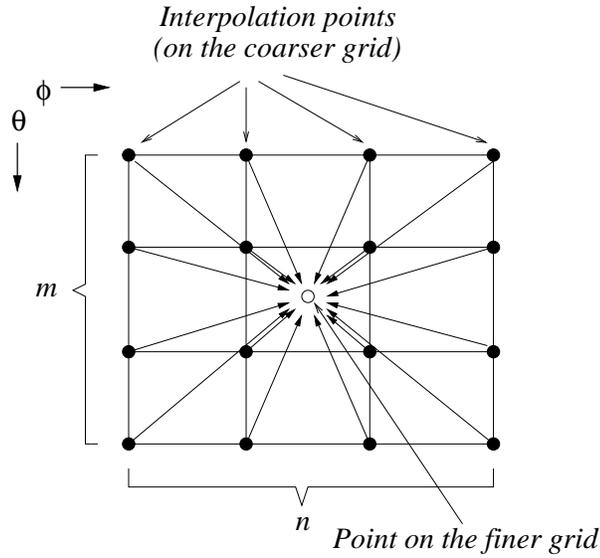


Figure 7.3: 2-D Interpolation.

and m is the number of samples taken in the θ direction for the interpolation. For each point in the ϕ direction, the values of the function at m points in the θ direction are used to reduce the 2-D interpolation into a 1-D interpolation. This reduced situation is shown in Figure 7.4, where there are n samples in the ϕ direction.

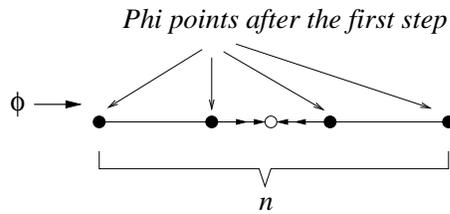


Figure 7.4: Interpolation points after the first step.

Then, the one-dimensional function $\hat{\mathbf{F}}$ can be used to complete the interpolation as

$$\tilde{\mathbf{F}}(\theta_t, \phi_t) = \sum_{i=1}^n l_i(\phi_t) \hat{\mathbf{F}}(\theta_t, \phi_i), \quad (7.6)$$

where

$$l_i(\phi_t) = \prod_{j \neq i} \frac{\phi_t - \phi_j}{\phi_i - \phi_j}. \quad (7.7)$$

Finally, the overall interpolation for a point can be written as

$$\tilde{\mathbf{F}}(\theta_t, \phi_t) = \sum_{i_1=1}^n \sum_{i_2=1}^m l_{i_1}(\phi_t) l_{i_2}(\theta_t) \mathbf{F}(\theta_{i_2}, \phi_{i_1}). \quad (7.8)$$

where $P = mn$ points are used to find the the value of the function at (θ_t, ϕ_t) .

This process have to be done for all $\tilde{P} = \tilde{m}\tilde{n}$ points in the fine grid. Consequently, the interpolation process between the two grids can be represented as

$$\tilde{\mathbf{F}} = \bar{\mathbf{I}}_{\tilde{P}P} \cdot \mathbf{F}, \quad (7.9)$$

where $\bar{\mathbf{I}}_{\tilde{P}P}$ is the interpolation matrix having a dimension of \tilde{P} by P . The function $\tilde{\mathbf{F}}$ is not exactly the same as \mathbf{F} since there exists some interpolation error. This error can be easily controlled by adjusting the number of sample points to be used in the interpolation at a point. This adjustment should be done carefully to balance the accuracy and the efficiency.

Considerations in Interpolation

If the target point is close to the boundaries of the sampling space, some of the interpolation points may fall outside. In other words, the interpolation may need to use the source data at (θ, ϕ) , which may have $\theta < 0$, $\theta > \pi$, $\phi < 0$, or $\phi > 2\pi$. To handle these cases, the extended coverage of the sample space is shown in Figure 7.5. In the figure, θ may have values from $-\pi$ to 2π , while ϕ has a range from -2π to 4π . Considering the unit sphere, these values are not actually allowed. However, this virtual extension should be done for a correct implementation of interpolation method.

In Figure 7.5, the vertex locations of the original space are marked with A , B , C , and D to follow the orientation of the replicas. Since the original space

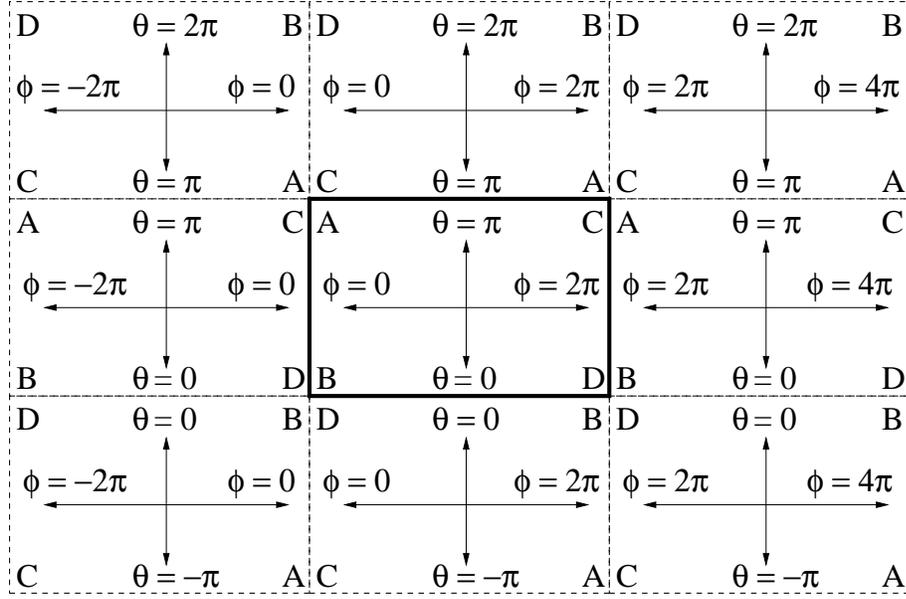


Figure 7.5: Extended sample space of unit sphere.

is periodic in the ϕ direction, the replicas in this direction do not change their orientations. In other words,

$$\begin{aligned}
 f(\theta, \phi) &= f(\theta, \phi + 2\pi), & \text{if } \phi < 0 \\
 f(\theta, \phi) &= f(\theta, \phi - 2\pi), & \text{if } \phi > 2\pi.
 \end{aligned} \tag{7.10}$$

The situation becomes more complex in the θ direction. The replicas are the double-rotated versions of the original space. In other words, rotation is required in both θ and ϕ directions. In addition, the value of the function has to be multiplied by -1 so that

$$\begin{aligned}
 f(\theta, \phi) &= -f(-\theta, \phi + \pi), & \text{if } \theta < 0, \quad 0 < \phi < \pi \\
 f(\theta, \phi) &= -f(-\theta, \phi - \pi), & \text{if } \theta < 0, \quad \pi < \phi < 2\pi \\
 f(\theta, \phi) &= -f(2\pi - \theta, \phi + \pi), & \text{if } \theta > \pi, \quad 0 < \phi < \pi \\
 f(\theta, \phi) &= -f(2\pi - \theta, \phi - \pi), & \text{if } \theta > \pi, \quad \pi < \phi < 2\pi.
 \end{aligned} \tag{7.11}$$

When the target point is close to the poles of the unit sphere, the quality of the interpolation can be improved by using the pole values of the function. The pole locations do not contribute to the angular integration; however, they can be carried during the aggregation process to reduce the interpolation error.

Introducing the interpolations, the aggregated value at the center of a basis cluster C' can be written as

$$\begin{aligned} \mathbf{F}^{C'}(\hat{\mathbf{k}}) = & \sum_{C'_1 \in G(C')} \beta_{C'_1 C'_1} \bar{\mathbf{I}}_{P'_1 P'_1} \cdot \sum_{C'_2 \in G(C'_1)} \beta_{C'_2 C'_1} \bar{\mathbf{I}}_{P'_1 P'_2} \cdot \sum_{C'_3 \in G(C'_2)} \beta_{C'_3 C'_2} \bar{\mathbf{I}}_{P'_2 P'_3} \cdot \dots \\ & \sum_{C'_f \in G(C'_{f-1})} \sum_{n \in C'_f} \mathbf{F}_{f n C'_f}(\hat{\mathbf{k}}) a_n, \end{aligned} \quad (7.12)$$

where $G(\cdot)$ represents the children list for the given cluster. The contribution of a cluster to its parent is performed by the shifting operators, which can be written as

$$\beta_{C_h C_l} = e^{i\mathbf{k} \cdot (\mathbf{r}_{C_h} - \mathbf{r}_{C_l})}, \quad (7.13)$$

where r_{C_h} is the center of the high-level cluster C_h while r_{C_l} is the center of its child, C_l . The aggregated value at a cluster center at any level has to be stored in the memory since, as it will be shown in the next subsection, all of the clusters are included in the translation step.

7.1.2 Translation in the MLFMA

Translation in the MLFMA should be done between the basis and testing clusters, which are far from each other. In addition to this, the parents of these clusters should be in the near-field of each other. Figure 7.6 sketches the situation for the clusters distributed in two levels. If the parents of the clusters are not in the near-field of each other, the translation between the clusters are included in a high-level translation, which is handled by the aggregation and disaggregation steps. Therefore, the direct translation between these clusters should be omitted.

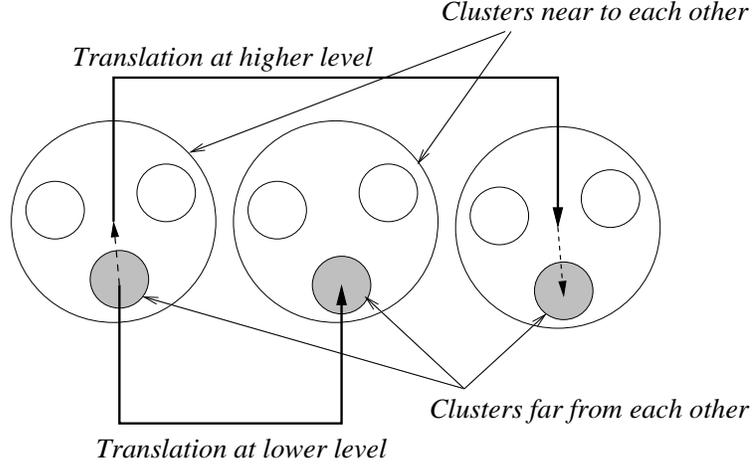


Figure 7.6: Translation process in the MLFMA.

After the translation step, each testing cluster has a summed value of translations from far-field clusters, which have parents close to the parent of the testing cluster. This value can be written for a testing cluster C as

$$\mathbf{F}_T^C(\hat{\mathbf{k}}) = \sum_{P(C') \in N(P(C))} \sum_{C' \in F(C)} T_L(k, |\mathbf{r}_{cc'}|, \hat{\mathbf{r}}_{cc'} \cdot \hat{\mathbf{k}}) \mathbf{F}^{C'}(\hat{\mathbf{k}}), \quad (7.14)$$

where $P(\cdot)$ represents the parent of the given cluster, while $N(\cdot)$ and $F(\cdot)$ represents the near-field and far-field clusters, as defined before. Finally, it should be recalled that $\mathbf{F}^{C'}(\hat{\mathbf{k}})$ is the aggregation result at the center of C' , calculated by using Equation (7.12).

7.1.3 Disaggregation in the MLFMA

After the translations, the summed values at the centers of the testing clusters have to be disaggregated towards the testing functions to be multiplied by the receiving patterns of the testing functions. It should be noted that the function to be multiplied by the receiving pattern of a testing function comes not only from its cluster, but also from the parent of its cluster. In general, disaggregation in

the MLFMA occurs by reversing the aggregation process, from high levels to the lowest level.

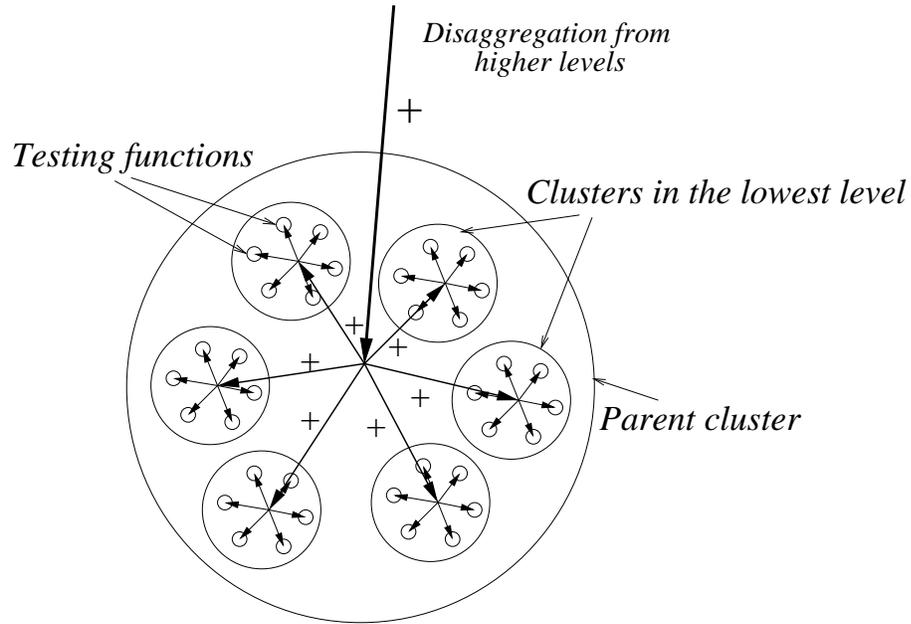


Figure 7.7: Disaggregation process in the MLFMA.

Figure 7.7 shows the disaggregation process for clusters at the lowest level. The values from the parent clusters are shifted to the centers of the clusters and added to the translation results associated with each testing-cluster center. Then, the disaggregation towards the testing functions proceeds as in the two-level FMM case. The parent cluster passes its values to all of its children, after it gets the value from its own parent.

The disagreement in the sampling of the unit sphere also appears in the disaggregation process. At this stage, one may choose to decimate the highly-sampled data into the coarser grid and continue the disaggregation towards the lower levels. However, this results in the angular integration to be done with the low sampling rate of the lowest level. A clever method, called antepolation, does

not disturb the general structure of the disaggregation, but allows the angular integration to be done with the high sampling rates of the implementation.

With the addition of the disaggregation process, the matrix-vector multiplication in the MLFMA can be written as

$$\sum_{n=1}^N Z_{mn} a_n = \sum_{C' \in N(C)} \sum_{n \in C'} Z_{mn} a_n + \frac{1}{4\pi} \int d^2 \hat{\mathbf{k}} \mathbf{F}_{fmc}(k) \cdot \mathbf{F}_{TD}^C(\hat{\mathbf{k}}), \quad (7.15)$$

where the testing function m resides in the lowest-level cluster C , while $\mathbf{F}_{TD}^C(\hat{\mathbf{k}})$ represents the obtained function at the center of C after the translation and disaggregation processes. The angular integration is evaluated numerically as

$$\int d^2 \hat{\mathbf{k}} \mathbf{F}_{fmc}(\hat{\mathbf{k}}) \cdot \mathbf{F}_{TD}^m(\hat{\mathbf{k}}) \approx \sum_i \sum_j w(\theta_i, \phi_j) \mathbf{F}_{fmc}(\theta_i, \phi_j) \cdot \mathbf{F}_{TD}^C(\theta_i, \phi_j), \quad (7.16)$$

where $w(\theta_i, \phi_j)$ represents the integration weights for the sampling rate of the lowest level.

If the decimation operation is applied during the disaggregation, the function \mathbf{F}_{TD}^C can be written in the recursive form as

$$\mathbf{F}_{TD}^C = \mathbf{F}_T^C + \bar{\mathbf{I}}_{P\hat{P}} \cdot \left(\beta_{CP(C)} \tilde{\mathbf{F}}_{TD}^{P(C)} \right), \quad (7.17)$$

where \mathbf{F}_T^C represents the contribution from the translation at the lowest level, while $\tilde{\mathbf{F}}_{TD}^{P(C)}$ is the inherited value from the parent by applying the shifting operator $\beta_{CP(C)}$. The interpolation matrix $\bar{\mathbf{I}}_{P\hat{P}}$ is applied on the finer grid of the parent function to obtain a coarser sampling. This process is different from the interpolation in the aggregation process, where a finer grid is obtained from a coarser grid. However, the rules do not change and the decimation operation can be thought of as a kind of interpolation.

Inserting (7.17) into (7.16), the numerical integration can be written as

$$\begin{aligned} \int d^2\hat{\mathbf{k}} \mathbf{F}_{fmc}(\hat{\mathbf{k}}) \cdot \mathbf{F}_{TD}^m(\hat{\mathbf{k}}) &\approx \sum_i \sum_j w(\theta_i, \phi_j) \mathbf{F}_{fmc}(\theta_i, \phi_j) \cdot \mathbf{F}_T^C(\theta_i, \phi_j) \\ &+ \sum_i \sum_j w(\theta_i, \phi_j) \mathbf{F}_{fmc}(\theta_i, \phi_j) \cdot \left[\bar{\mathbf{I}}_{P\bar{P}} \cdot \left(\beta_{CP(C)} \tilde{\mathbf{F}}_{TD}^{P(C)} \right) \right] (\theta_i, \phi_j). \end{aligned} \quad (7.18)$$

It is obvious that the decimation method leads to the angular integration to be done at the lowest sampling rate of the implementation.

Anterpolation

If the second part of the angular integration in (7.18) was evaluated in the finer level, then the interpolation of \mathbf{F}_{fmc} would be required so that

$$\begin{aligned} \int d^2\hat{\mathbf{k}} \mathbf{F}_{fmc}(\hat{\mathbf{k}}) \cdot \mathbf{F}_{TD}^m(\hat{\mathbf{k}}) &\approx \sum_i \sum_j w(\theta_i, \phi_j) \mathbf{F}_{fmc}(\theta_i, \phi_j) \cdot \mathbf{F}_T^C(\theta_i, \phi_j) \\ &+ \sum_{\tilde{i}} \sum_{\tilde{j}} \tilde{w}(\theta_{\tilde{i}}, \phi_{\tilde{j}}) \left[\bar{\mathbf{I}}_{\tilde{P}\tilde{P}} \cdot \mathbf{F}_{fmc} \right] (\theta_{\tilde{i}}, \phi_{\tilde{j}}) \cdot \beta_{CP(C)} \tilde{\mathbf{F}}_{TD}^{P(C)}(\theta_{\tilde{i}}, \phi_{\tilde{j}}), \end{aligned} \quad (7.19)$$

where $\tilde{w}(\theta_i, \phi_j)$ represents the integration weights for the fine level. Although such a change destroys the disaggregation process and does not seem to be practical, the idea leads to the anterpolation method. Rearrangement of the interpolation and the summations gives rise to new expression as

$$\begin{aligned} \int d^2\hat{\mathbf{k}} \mathbf{F}_{fmc}(\hat{\mathbf{k}}) \cdot \mathbf{F}_{TD}^m(\hat{\mathbf{k}}) &\approx \sum_i \sum_j w(\theta_i, \phi_j) \mathbf{F}_{fmc}(\theta_i, \phi_j) \cdot \mathbf{F}_T^C(\theta_i, \phi_j) \\ &+ \sum_i \sum_j \mathbf{F}_{fmc}(\theta_i, \phi_i) \cdot \left\{ \bar{\mathbf{I}}_{\tilde{P}\tilde{P}}^T \cdot \left[\tilde{w} \beta_{CP(C)} \tilde{\mathbf{F}}_{TD}^{P(C)} \right] \right\} (\theta_i, \phi_i), \end{aligned} \quad (7.20)$$

where $\bar{\mathbf{I}}_{\tilde{P}\tilde{P}}^T$ is the transpose of the coarse-to-fine interpolation matrix. It should be noted that the summations are performed with low-level indexing. However, the accuracy of the angular integration is improved since the evaluation is based on the finer grid of the higher level. The operation on the $\tilde{w} \beta_{CP(C)} \tilde{\mathbf{F}}_{TD}^{P(C)}$ is generally called anterpolation. In this operation, fine-grid functions are multiplied

by the transpose of the interpolation matrix and reduced to coarse-grid functions.

Figure 7.8 shows the interpolation, decimation, and anterpolation operations applied on a one-dimensional function. As shown in 7.8(a), the interpolation results in higher sampling of the function, which was previously on a coarse grid. The anterpolation operation from coarse to fine grid can be derived directly from the interpolation. The only difference is changing the directions of interpolation arrows. It should be noted that the weights used in the interpolation are not changed. Therefore, the resulting function after the anterpolation is completely different from the applied function.

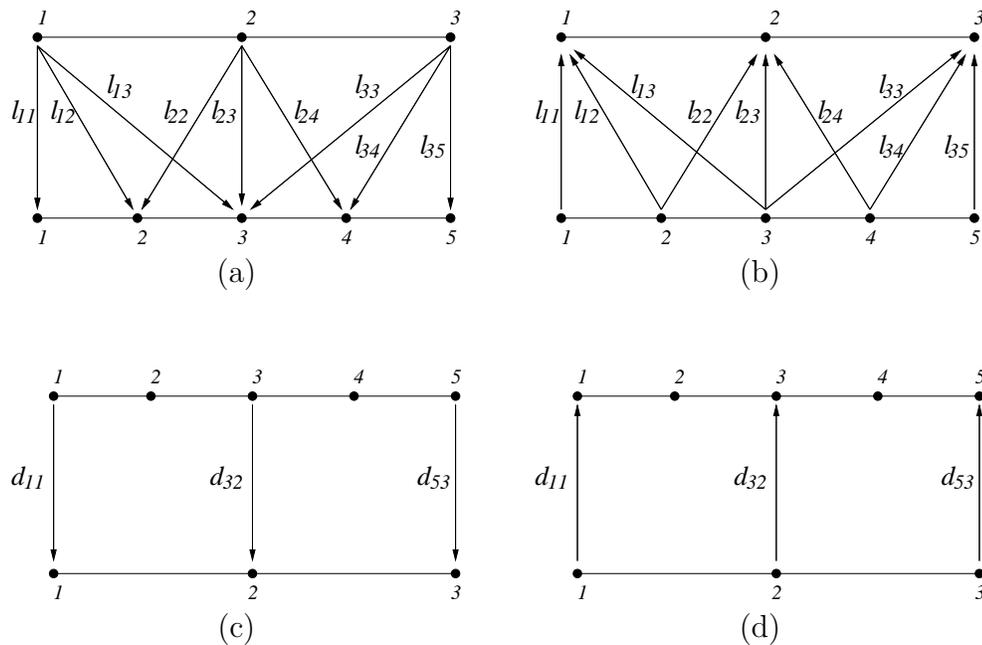


Figure 7.8: Operations on function samplings: (a) interpolation, (b) anterpolation from coarse to fine grid, (c) decimation, and (d) anterpolation from fine to coarse grid.

Decimation shown in 7.8(c) is actually a kind of interpolation applied on a fine grid to obtain a coarser grid of the same function. The anterpolation operation

from fine to coarse grid can be derived directly from decimation by simply changing the arrow directions. In the given example, it is obvious that this operation changes the function completely, since the value of the resulting function appears to be zero at the second and fourth points. The function may not be zero at these points, where the coarse grid does not give any value.

In general, the operations in Figure 7.8 can be used in the numerical evaluation of the integrals as

$$\int dx \mathbf{f}(x) \cdot \mathbf{g}(x) \quad (7.21)$$

when the functions $\mathbf{f}(x)$ and $\mathbf{g}(x)$ have different number of samplings in the integration interval. Let the function $\mathbf{f}(x)$ have a finer grid. Then, the following methods can be used to evaluate the integral in (7.21):

1. Interpolate the function $\mathbf{g}(x)$ and evaluate the integration with high sampling rate:

$$\int dx \mathbf{f}(x) \cdot \mathbf{g}(x) \approx \sum_{\tilde{i}} \tilde{w}(\tilde{i}) \tilde{\mathbf{f}}(\tilde{i}) \cdot [\bar{\mathbf{I}}_{\tilde{P}P} \cdot \mathbf{g}](\tilde{i}). \quad (7.22)$$

2. Anterpolate the function $\mathbf{f}(x)$ and evaluate the integration with low sampling rate. However, the accuracy will be the same as in the first method:

$$\int dx \mathbf{f}(x) \cdot \mathbf{g}(x) \approx \sum_i w(i) \mathbf{g}(i) \cdot [\bar{\mathbf{I}}_{P\tilde{P}}^T \cdot \mathbf{f}](i). \quad (7.23)$$

3. Decimate the function $\mathbf{f}(x)$ and evaluate the integration with low sampling rate. The accuracy of the evaluation drops as compared to the first and second methods:

$$\int dx \mathbf{f}(x) \cdot \mathbf{g}(x) \approx \sum_i w(i) \mathbf{g}(i) \cdot [\bar{\mathbf{I}}_{P\tilde{P}} \cdot \mathbf{f}](i). \quad (7.24)$$

4. Anterpolate the function $\mathbf{g}(x)$ and evaluate the integration with high sampling rate. This is the most inefficient method, which uses high sampling

rate but has the same accuracy as the third method:

$$\int dx \mathbf{f}(x) \cdot \mathbf{g}(x) \approx \sum_{\tilde{i}} \tilde{w}(\tilde{i}) \tilde{\mathbf{f}}(\tilde{i}) \cdot [\bar{\mathbf{I}}_{P\tilde{P}}^T \cdot \mathbf{g}](\tilde{i}). \quad (7.25)$$

7.2 Clustering and Structure of the MLFMA

A simple clustering in the MLFMA can be accomplished by dividing the geometry into equal cubes at each level. First, the geometry to be solved is put into a main cube as shown in Figure 7.9(a). Then, this main cube is divided into eight subcubes. The process continues by dividing the each subcube again into eight cubes, and the levels of the MLFMA are formed in this manner. At the end, the cubes at the lowest level become so small that each contain only a few triangular elements.

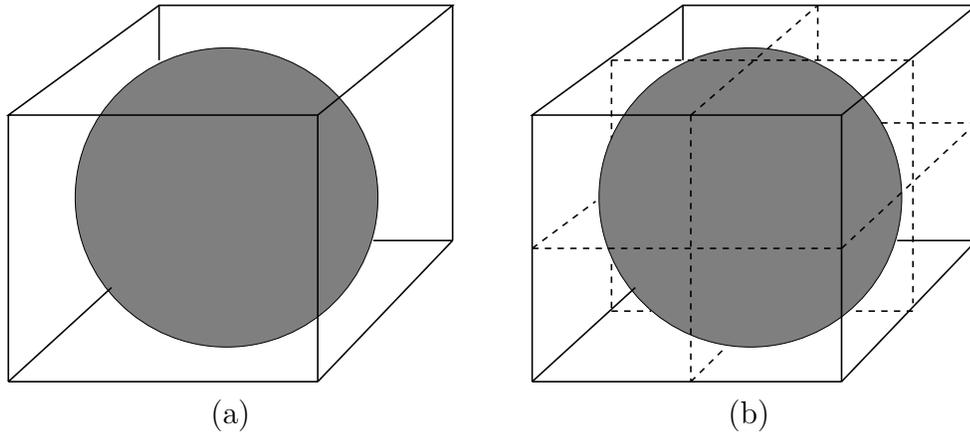


Figure 7.9: Clustering with cubes in the MLFMA: (a) the main cube, (b) subcubes of the main cube.

Forming the clustering by equal cubes makes it easier to determine the near-field and far-field clusters for a given cluster. Generally, it is enough to include the neighboring (touching) cubes in the near-field group, while the rest of the cubes in the same level can be assigned to be in the far-field list. As shown in

Figure 7.10, the distance between two far-field clusters is $2a$ in the worst case, where a is the edge length of the cubes. Then,

$$L \approx kd + 1.8d_0^{2/3}(kd)^{1/3} = ka \frac{\sqrt{3}}{2} + 1.8d_0^{2/3} \left(ka \frac{\sqrt{3}}{2} \right)^{1/3}, \quad (7.26)$$

which gives

$$ka > 1.8d_0 \quad (7.27)$$

to satisfy the criteria as $L < kr_{cc'}$. It should be noted that the translation error for given d_0 is the largest for the smallest boxes in the lowest level. The rule of thumb for the size of these boxes in the MLFMA is to choose $a \approx 0.25\lambda$.

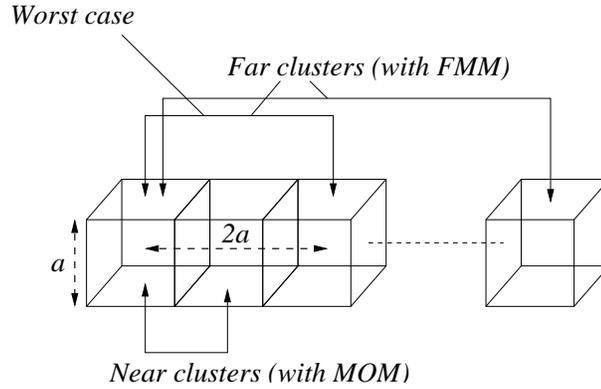


Figure 7.10: Near and far clusters in the clustering method with cubes. In the worst case, the distance between two far-field clusters is $2a$.

For an arbitrary geometry, some of the boxes in the cluster structure are empty, which means that these clusters does not include any elements. Those boxes should be marked at the beginning of the implementation to avoid unnecessary work during the MLFMA.

For efficiency, the clustering should be represented by a tree structure as shown in Figure 7.11, where the nonempty clusters are indicated. As also shown in the figure, the highest level of the MLFMA is the third level of the tree structure.

7.3.1 Computational Analysis of Aggregation and Disaggregation

Considering the lowest level, following observations can be made:

1. The radius of the clusters is $O(1)$.
2. $L \propto (kd)$ and $K = L^2$. Therefore, the number of harmonics in the translation function, L , as well as the number of angular samples for the radiation of the clusters, K , are also $O(1)$.
3. For $O(N)$ clusters in the lowest level, $O(N)$ FLOPs are required for the aggregation. The memory requirement is also $O(N)$ to store the aggregation results at the centers of the clusters.

Considering the highest level, following observations can be made:

1. The radius of the clusters is $O(N^{1/2})$.
2. $L \propto (kd)$ and $K = L^2$. Therefore, $K = O(N)$.
3. The number of clusters is $O(1)$. Therefore, the aggregation process requires $O(N)$ FLOPs. The memory requirement is also $O(N)$ to store the aggregation results at the centers of the clusters.

Considering the levels between the highest and lowest ones, it can be observed that the increase in the memory and processing time required by a cluster is balanced by the decrease in the number of clusters. Therefore, the memory requirement and processing time for the aggregation process appears to be $O(N)$ per level. There are $O(\log N)$ levels so that they become $O(N \log N)$ for the overall aggregation.

In these calculations, the interpolation operation for a point is assumed to be

done in $O(1)$ processing time, and the Lagrange interpolation satisfies this requirement. In addition, the disaggregation process is simply the reverse of the aggregation process, so that its memory requirement and processing time are also $O(N \log N)$.

It should be recalled that both aggregation and disaggregation steps are performed at each iteration. Therefore, for the solution of a problem, the processing time for these steps appears to be $O(IN \log N)$, where I is the number of iterations.

7.3.2 Computational Analysis of Translation

Considering the lowest level, following observations can be made:

1. For each basis cluster, there are $O(1)$ testing clusters for the translation process.
2. For each basis cluster, data to be translated is $O(1)$.
3. For $O(N)$ clusters in the lowest level, $O(N)$ FLOPs are required for the translations.

Considering the highest level, following observations can be made:

1. For each basis cluster, there are again $O(1)$ testing clusters for the translation process.
2. For each basis cluster, data to be translated is $O(N)$.
3. For $O(1)$ clusters in the highest level, $O(N)$ FLOPs are required for the translation.

In general, each level requires $O(N)$ processing time to complete the translation. Then, the total processing time for the translation operation becomes

$O(N \log N)$. This operation is also required to be done at each iteration.

On the other hand, the translation function has to be calculated and stored before the iterative algorithm. For the clusters in the lowest level, $K = O(1)$ and $L = O(1)$ so that the overall calculation requires $O(N)$ processing time. However, for the clusters in the highest level, $K = O(N)$ and $L = O(N^{1/2})$ so that the overall calculation for $O(1)$ clusters becomes $O(N^{3/2})$. Then, the calculation of the translation function for a problem appears to be $O(N^{3/2})$ with the domination of the highest level. The memory required to store the translation function is $O(N \log N)$, with $O(N)$ required for each level.

At the end, the near-field interactions can be included. Considering the clusters in the lowest level, the number of neighboring clusters is $O(1)$ for each cluster. In addition, all of these clusters have $O(1)$ unknowns. Consequently, both memory requirement and processing time to evaluate the near-field interactions becomes $O(N)$.

As a result, a MLFMA has a processing time as

$$\begin{aligned} C_{agg/dis}IN \log N + C_{trans1}IN \log N + C_{trans2}N^{3/2} + C_{near}N \\ = O(IN \log N) + O(N^{3/2}) \end{aligned} \quad (7.28)$$

while the memory requirement is

$$C_{agg/dis}N \log N + C_{trans}N \log N + C_{near}N = O(N \log N). \quad (7.29)$$

If the number of iterations has an order larger than $O(N^{1/2}/\log N)$, then the contribution from the calculation of the translation function is not dominant in the above expression. Otherwise, the computational cost of the calculation of the translation function becomes dominant as the problem size gets larger. The remedy is the interpolation of the translation function as given in [25]. By this

way, it is guaranteed to get $O(N \log N)$ processing time per iteration. On the other hand, it should be indicated that the constant term C_{trans2} is small and the $N^{3/2}$ term becomes significant only for very large problems.

7.4 Results

To check the accuracy and efficiency of the MLFMA, the sphere problems shown in Figure 4.11 are solved by an implementation with the following properties:

1. The CFIE with $\alpha = 0.2$ is used to formulate the problems. This choice was known to be the best from Chapter 5.
2. Near-field interactions are evaluated by using adaptive methods for the inner integrals with singularity extraction. The error criteria is chosen to be 1%. The outer integrals are evaluated by a 7-point Gaussian quadrature rule with the singularity extraction for the MFIE part. The first error source of the implementation is these numerical integrations.
3. d_0 is chosen to be 3 for the translations in the FMM. Translations are the second error source of the implementation, while the angular integration is the third one.
4. The size of the smallest boxes in the MLFMA structure is fixed to be 0.25λ . Since the size of the geometry is arbitrary, this setting requires the adjustment in the size of the largest box before constructing the tree structure.
5. For all interpolations, the number of interpolation points is fixed to 2 for all directions (down, up, right, and left). This leads to 16-point local Lagrange interpolations at every point. Interpolation is the fourth error source of the implementation.

7.4.1 Accuracy of the MLFMA Implementations

Figure 7.12 shows the RCS values on the x - y plane obtained by the MLFMA when the spheres are illuminated by a y -polarized plane wave propagating in the $-x$ direction. The analytical results are also plotted to show the accuracy of the implementation.

7.4.2 Efficiency of the MLFMA Implementations

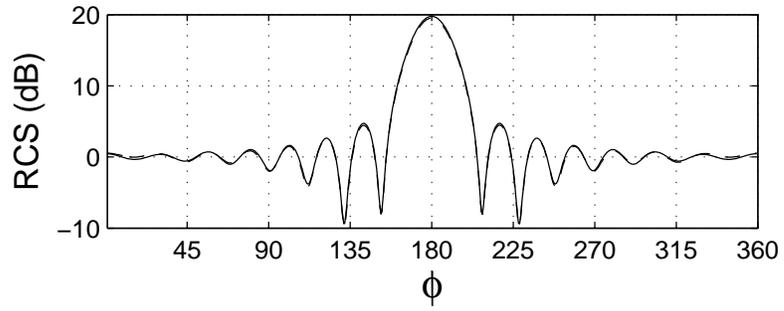
In Tables 7.1 and 7.2, the processing time and memory requirements are given for the two-level FMM and the MLFMA implementations. For the largest two problems, the values are not available for the FMM implementation, since the computational resources are not enough to solve these problems with this implementation. With these results, the higher efficiency of the MLFMA is obvious.

Table 7.1: Comparison of the processing times of the CFIE-FMM and the CFIE-MLFMA implementations.

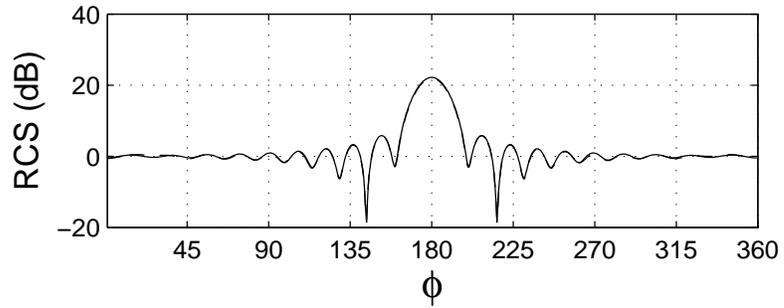
Radius of the Sphere	Processing Time for the CFIE-FMM (sec)	Processing Time for CFIE-MLFMA (sec)
1.5 λ	1853	313
2.0 λ	3991	714
3.0 λ	NA	1569
6.0 λ	NA	7516

Table 7.2: Comparison of the memory requirements of the CFIE-FMM and the CFIE-MLFMA implementations.

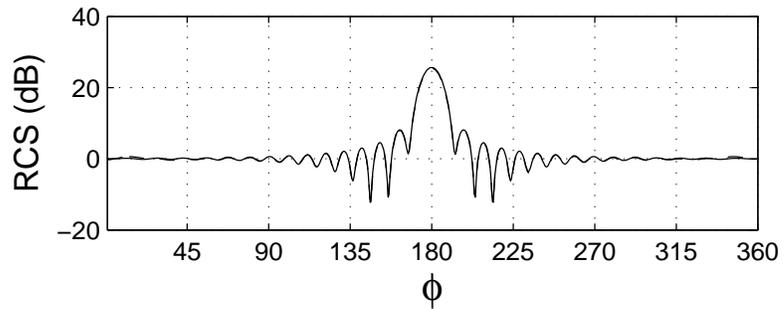
Radius of the Sphere	Memory Requirement for the CFIE-FMM (MB)	Memory Requirement for the CFIE-MLFMA (MB)
1.5 λ	260	78
2.0 λ	511	152
3.0 λ	NA	333
6.0 λ	NA	1403



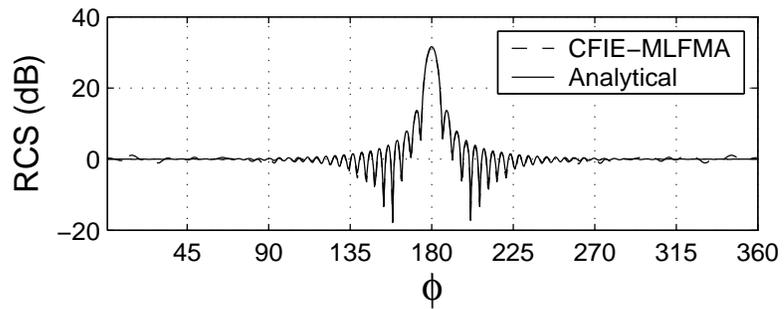
(a)



(b)



(c)



(d)

Figure 7.12: RCS values obtained by the CFIE-MLFMA implementation for a sphere having a radius of (a) 1.5λ , (b) 2.0λ , (c) 3.0λ , and (d) 6.0λ .

Chapter 8

Conclusion

In this thesis, the fast multipole method (FMM) is investigated in detail for the solution of scattering problems involving arbitrarily shaped three-dimensional conducting surfaces. To form the basis for the FMM, the method of moments (MOM) is presented with the formulation of three integral equations: the electric-field integral equation (EFIE), the magnetic-field integral equation (MFIE), and the combined-field integral equation (CFIE).

To define the basis and testing functions in the MOM and FMM implementations, the models are meshed with linear triangular elements. At this stage, two error sources are identified. One of them is the inexact modelling of the curved surfaces with linear triangles, while the other one is the discrete approximation of continuous unknown functions. Both of the errors can be controlled by adjusting the size of the triangular elements.

Chapter 3 presents an overview on the MOM implementation using the EFIE formulation and the Rao-Wilton-Glisson (RWG) functions to expand the current density. An efficient implementation of the MOM requires the elimination of the

computational redundancies, singularity extraction, and parallelism in the evaluation of the integrals. The procedure becomes more difficult when the MFIE is used since the singularity becomes stronger and the cross product increases the number of basic integrals, each of which has to be evaluated independently. An efficient implementation of the MOM using the MFIE formulation and the RWG functions is given in [10]. On the other hand, an alternative implementation is presented in Chapter 4, which is based on the procedure given for the EFIE in Chapter 3.

An investigation on the limit values of the analytical integrals appearing in the MFIE formulation is presented in Chapter 4. It is shown that the solid angle expression in the MFIE can be derived in these evaluations. In addition, a singularity extraction method is suggested for the unbounded integrand of the testing integrals in the neighboring interactions of the MFIE. Numerical integration is another error source introduced by the MOM. For the efficient evaluation of these integrals, adaptive integration algorithms are presented in Chapter 3, which make it possible to control the integration error.

Chapter 5 presents an overview on the FMM implementation using the EFIE formulation and the RWG functions. As the problem size gets larger, the FMM becomes more efficient than the MOM. However, the expected efficiency of the FMM is not guaranteed to be obtained. The EFIE systems are usually ill-conditioned, especially when the problem involves a closed surface, and the iterative solution of these systems becomes difficult. Chapter 6 shows that the EFIE may not be used with efficient preconditioners, such as the diagonal and block-diagonal preconditioners. Consequently, the ill-conditioned EFIE systems require large preconditioners in order to keep the number of iterations reasonably low. The use of these large preconditioners leads to significant increases in the processing

time and memory requirements, and the efficiency of the FMM drops significantly.

The CFIE is known to give better-conditioned systems for the closed geometries [4]. Consequently, it becomes possible to use more efficient preconditioners and the efficiency of the FMM can be recovered. Chapter 6 presents a detailed investigation on the efficiency of the FMM implementations, according to the choices of the integral equation, iterative solver, and preconditioning technique. It is shown that the increase in the number of iterations is very limited for the CFIE, even with the block-diagonal preconditioners, for larger problems.

The FMM brings about two additional error sources: translation operations and angular integrations. However, they can be controlled by adjusting the truncation number appropriately. Comparisons between the analytical and numerical results for the sphere problems are presented in order to show the accuracy of the implementations. The extension of the FMM is the MLFMA, which further reduces the complexity and memory requirements but adds another controllable error source: the interpolation and antinterpolation operations between the levels. With the MLFMA, it becomes possible to solve a sphere problem of 6λ radius and 132,000 unknowns with only about 1.4 GB of memory and about 7500 seconds of processing time, using a single Digital Alpha processor. The accuracy of this solution is shown to be sufficiently good.

The future work includes the parallelization of the MLFMA to solve larger electromagnetic scattering problems. In addition, the accuracy of the MFIE is currently under investigation. When the objects become sharper, it is observed that the accuracy of the MFIE drops, which also affects the accuracy of the CFIE implementations. For this purpose, the use of other types of the basis functions is being considered for the MFIE.

Bibliography

- [1] R. F. Harrington, *Field Computation by Moment Methods*. IEEE Press, 1993.
- [2] N. Engheta, W. D. Murphy, V. Rokhlin, and M. S. Vassiliou, “The fast multipole method (FMM) for electromagnetic scattering problems,” *IEEE Trans. Antennas Propagat.*, vol. 40, no. 6, pp. 634–641, June 1992.
- [3] R. Coifman, V. Rokhlin, and S. Wandzura, “The fast multipole method for the wave equation: a pedestrian prescription,” *IEEE Ant. Propag. Mag.*, vol. 35, no. 3, pp. 7–12, June 1993.
- [4] 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.
- [5] J. M. Song, C.-C. Lu, W. C. Chew, and S. W. Lee, “Fast Illinois solver code (FISC),” *IEEE Ant. Propag. Mag.*, vol. 40, no. 3, pp. 27–34, Oct. 1998.
- [6] W. C. Chew, J.-M. Jin, E. Michielssen, and J. M. Song, eds., *Fast and Efficient Algorithms in Computational Electromagnetics*. Boston-London: Artech House Publishers, 2001.
- [7] L. Gürel and W. C. Chew, “A recursive T-matrix algorithm for strips and patches,” *Radio Sci.*, vol. 27, pp. 387-401, 1992.
- [8] W. C. Chew, L. Gürel, Y. M. Wang, G. Otto, R. Wagner, and Q. H. Liu, “A generalized recursive T-matrix algorithm for wave scattering solutions in

- two dimensions,” *IEEE Trans. Microwave Theory Tech.*, vol. 40, pp. 716–723, 1992.
- [9] C.-C. Lu and W. C. Chew, “Fast far-field approximation for calculating the RCS of large objects,” *Micro. Opt. Tech. Lett.*, vol. 8, pp. 238–241, 1995.
- [10] R. E. Hodges and Y. Rahmat-Samii, “The evaluation of MFIE integrals with the use of vector triangle basis functions,” *Micro. Opt. Tech. Lett.*, vol. 14, no.1, pp. 9–14, Jan. 1997.
- [11] D. R. Wilton, S. M. Rao, A. W. Glisson, D. H. Schaubert, O. M. Al-Bundak, and C. M. Butler, “Potential integrals for uniform and linear source distributions on polygonal and polyhedral domains,” *IEEE Trans. Antennas Propagat.*, vol. AP-32, pp. 276–281, March 1984.
- [12] R. D. Graglia, “On the numerical integration of the linear shape functions times the 3-D Green’s function or its gradient on a plane triangle,” *IEEE Trans. Antennas Propagat.*, vol. 41, no. 10, pp. 1448–1454, Oct. 1993.
- [13] A. J. Poggio and E. K. Miller, “Integral equation solutions of three-dimensional scattering problems,” in R. Mittra, ed., *Computer Techniques for Electromagnetics*. Oxford: Pergamon Press, 1973, Chap. 4.
- [14] N. Morita, N. Kumagai, and J. R. Mautz, *Integral Equation Methods for Electromagnetics*. Norwood, MA: Artech House, 1990.
- [15] J. R. Mautz and R. F. Harrington, “H-field, E-field, and combined-field solutions for conducting bodies of revolution,” *AEÜ*, vol. 32, pp. 157–164, Apr. 1978.
- [16] L. M. Correia, “A comparison of integral equations with unique solution in the resonance region for scattering by conducting bodies,” *IEEE Trans. Antennas Propagat.*, vol. AP-41, pp. 52–58, Jan. 1993.

- [17] L. Gürel, K. Sertel, and İ. K. Şendur, “On the choice of basis functions to model surface electric current densities in computational electromagnetics,” *Radio Sci.*, vol. 34, no. 6, pp. 1373-1387, Nov.-Dec. 1999
- [18] S. M. Rao, D. R. Wilton, and A. W. Glisson, “Electromagnetic scattering by surfaces of arbitrary shape,” *IEEE Trans. Antennas Propagat.*, vol. AP-30, pp. 409–418, May 1982.
- [19] A. F. Peterson, S. L. Ray, and R. Mittra, *Computational Methods for Electromagnetics*. New York: IEEE Press, 1998.
- [20] B. Shanker, A. A. Ergin, K. Aygün, and E. Michielssen, “Analysis of transient electromagnetic scattering from closed surfaces using a combined field integral equation,” *IEEE Trans. Antennas Propagat.*, vol. 48, no. 7, pp. 1064–1074, July 2000.
- [21] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. Der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Philadelphia, PA: SIAM, 1993.
- [22] C. F. Smith, A. F. Peterson, and R. Mittra, “The biconjugate gradient method for electromagnetic scattering,” *IEEE Trans. Antennas Propagat.*, vol. 38, no. 6, pp. 938–940, June 1990.
- [23] A. Greenbaum, *Iterative Methods for Solving Linear Systems*. Philadelphia, PA: SIAM, 1997.
- [24] L. Gürel and H. Bağcı, “Fast-multipole-method solution of the electromagnetic scattering from a stealth geometry,” *Bilkent University Research Report*, 2002.

- [25] J. M. Song and W. C. Chew, "Interpolation of translation matrix in MLFMA," *Micro. Opt. Tech. Lett.*, vol. 30, no. 2, pp. 109–113, July 2001.

Glossary

a_n : Coefficient of the n^{th} basis function.

A_i : Area of the i^{th} triangle.

$\mathbf{b}_n(\cdot)$: n^{th} basis function.

$\mathbf{B}(\cdot)$: Magnetic flux density.

$\mathbf{D}(\cdot)$: Electric flux density.

$\mathbf{E}(\cdot)$: Electric field intensity.

$\mathbf{E}^{inc}(\cdot)$: Incident electric field intensity.

$\mathbf{E}^{sca}(\cdot)$: Scattered electric field intensity.

$\mathbf{F}(\cdot, \cdot)$: Vector current moment.

$\mathbf{F}_{fnc}(\cdot)$: Radiation pattern of the n^{th} basis function from the point c .

$\mathbf{F}_{fmc}(\cdot)$: Receiving pattern of the m^{th} testing function with respect to point c .

$\mathbf{F}^C(\cdot)$: Aggregated radiation at the center of the basis cluster C .

$\mathbf{F}_T^C(\cdot)$: Incoming wave at the center of the testing cluster C , after the translation step.

$\mathbf{F}_{TD}^C(\cdot)$: Incoming wave at the center of the testing cluster C , after the translation and disaggregation steps.

$F(C)$: Far-field clusters of the cluster C .

$g(\cdot, \cdot)$: Green's function.

$\overline{\mathbf{G}}(\cdot, \cdot)$: Dyadic Green's function.

$G(C)$: Children clusters for the cluster C .

$h_1^{(1)}(\cdot)$: Spherical Hankel function of the first kind.

\mathbf{H} : Magnetic field intensity.

$\mathbf{H}^{inc}(\cdot)$: Incident magnetic field intensity.

$\mathbf{H}^{sca}(\cdot)$: Scattered magnetic field intensity.

$\bar{\mathbf{I}}$: Identity matrix.

$\bar{\mathbf{I}}_{\bar{P}P}(\cdot)$: Interpolation matrix (fine-to-coarse).

$\bar{\mathbf{I}}_{P\bar{P}}(\cdot)$: Decimation matrix (coarse-to-fine).

$\bar{\mathbf{I}}_{\bar{P}P}^T(\cdot)$: Anterpolation (transpose interpolation) matrix (coarse-to-fine).

$j_l(\cdot)$: Spherical Bessel function.

$\mathbf{J}(\cdot)$: Electric current density.

k : Wavenumber.

$\hat{\mathbf{k}}$: Angular unit vector.

K : Number of samples on the unit sphere.

L : Truncation number.

$\bar{\mathbf{M}}$: Preconditioner matrix.

$\hat{\mathbf{n}}$: Normal unit vector on the surface.

N : Number of unknowns.

$N(C)$: Near-field clusters of the cluster C .

$P_l(\cdot)$: Legendre polynomial.

\mathbf{r} : Observation point.

\mathbf{r}' : Source point.

\mathbf{r}_i : Residual vector at i^{th} iteration.

S : Surface.

$\hat{\mathbf{t}}$: Tangential unit vector on the surface.

$\mathbf{t}_m(\cdot)$: m^{th} testing function.

$T_L(\cdot, \cdot, \cdot)$: Translation function.

\mathbf{v} : Excitation vector.

$\bar{\mathbf{Z}}$: Impedance matrix.

Z_{mn} : Interaction between the m^{th} testing and the n^{th} basis functions.

$Z_{ik,jl}$: Interaction between the i^{th} and j^{th} triangles, where the testing and basis functions are aligned with respect to the k^{th} and l^{th} edges, respectively, of the triangles.

α : CFIE constant.

$\delta(\cdot)$: Dirac delta function.

$\delta_i(\cdot)$: Shape function.

δ_{ij} : Kronecker delta function.

η : Intrinsic impedance.

λ : Wavelength.

μ : Permeability.

$\rho(\cdot)$: Electric charge density.

ω : Angular frequency.

Ω_i : Internal solid angle.

Ω_o : External solid angle.