HELMHOLTZ AND HIGH-FREQUENCY MAXWELL MULTILEVEL FAST MULTIPOLE ALGORITHMS WITH SELF-TUNING LIBRARY

by
Abdulkadir C. Yucel

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Science
(Electrical Engineering)
in The University of Michigan
2008

Thesis Committee:
Professor Eric Michielssen, Chair
Associate Professor Mahta Moghaddam
Assistant Professor Anthony Grbic

“İlm kesbiyle pâye-i rifat, Arzuyu muhâl imiş ancak,
Aşk imiş her ne var âlemde, İlm bir kîl-ü kâl imiş ancak.”
- Fuzûlî

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Professor Eric Michielssen, for his support and guidance. His sincerity, scientific enthusiasm and persistence have set a strong scientific model in this student's mind. I am also grateful to Professor Mahta Moghaddam and Professor Anthony Grbic for serving on my committee.

Many special thanks go to Professor Ahmet Arif Ergin from Gebze Institute of Technnology for his encouragements and close mentorship throughout the last seven years of my life. In addition, I would like to thank to Turkish Fulbright Commission for financially supporting my grad research.

Thanks to my collegeues in Professor Michielssen's research group with whom I've shared knowledge, experience, hard times, fun, and lunch/dinner tables. I owe particular acknowledgement to Dr. Francesco Andriulli, Dr. Hakan Bagci, Xi Lin, Pelumi Osoba, Felipe Valdes Valenzuela, and Onur Bakir. In addition, I am grateful to all radlabers, particularly Scott Rudolph, Amit Patel, and Morteza Nick for their valuable friendships.

Last but not least, I would like to thank to my father, Dr.T.Savas Yucel, from whom I've learned how to struggle for what I set to my mind, to my mother, S.Sevtap Yucel, who taught me how to be patient while she's raising me and my brothers up with her great endurance and love. Moreover, many thanks to my brothers Dr. Ali Bogachan Yucel and Abdullah Yucel for their encouragements and support. I desire to acknowledge some of friends who are like members of my family and always with me through their existences and encouragements. A special mention goes to Seyit Ahmet Sis, Atila Ucar, and Ali Burak Unlu.

Ann Arbor
2008

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background

The fast multipole method (FMM) has been selected as the one of the ten most significant algorithms discovered the in 20th century by the IEEE Computing in Science and Engineering Society [1]. It was cited along with algorithms such as the FFT, Quicksort, Monte Carlo methods …etc. The inventors of FMM, Professors Leslie Greengard and Vladimir Rokhlin, were the recipients of 2001 Leroy P. Steele Prize for a Seminal Contribution to Research of the American Mathematical Society for their paper titled "A Fast Algorithm for Particle Simulations" [2]. Although their paper focused on rapid evaluation of gravitational and Coulombic potentials, it comprised the fundamental ideas permitting the generalization of the method to a variety of applications. The last decade of 20th century have witnessed wide spread activity relating to the implementation of the FMM for kernels governing the

- ◊ Laplace and Poisson equations [3-7],
- ◊ Helmholtz equation [8, 9],
- ◊ Maxwell's equation [10-12],
- ◊ Stokes flow and Navier-Stokes equations [13-15],
- ◊ N-body problems [2, 16],
- ◊ Yukawa potentials [17, 18] …etc.

Implementations of FMM to these kernels make various large-scale scientific and engineering computations possible in areas such as computational astronomy, quantum mechanics, chemistry, biology, electromagnetics, electrostatics… and so on. In these

disciplines, large-scale problems become solvable through rapidly evaluating N^2 interactions by multipole approximations. These approximations lead to considerable reductions in the computational complexities of the problems from $O(N^2)$ to $O(N\text{Log}N)$ (or $O(N)$ in some cases).

The fast solution of dynamic problems in electromagnetics calls for the application of FMMs to the Helmholtz and Maxwell equations. Prof. Rokhlin proposed the FMM for 2D Helmholtz equation in 1990 [19]. Then, he extended it to the 3D Helmholtz equation [9]. That work was followed by others seeking the solution of Maxwell equations for the electromagnetic scattering problem [8, 10, 20]. Soon after, Prof. Chew's research group developed a multilevel version of the FMM, known as the multilevel fast multipole algorithm (MLFMA), and conducted pioneering research on this topic [11, 12, 21-36]. Moreover, Prof. Michielssen's research group developed time-domain counterparts of FMM [37] and MLFMA [38] for the wave equation and applied it in various real-world time-domain problems [39-43].

1.2 Motivation

MLFMA for the Helmholtz and Maxwell equations accelerate the integral equation-based iterative solution of acoustic and electromagnetic scattering problems by evaluating interactions between sources and observers in a hierarchical framework. The hierarchical framework requires (i) breaking up the source constellation into groups, (ii) computing each group's far field signature, and (iii) translating these far-field signatures between group centers to arrive at observer fields. The first step which falls out of the scope of this study can be accomplished by a hashed octree algorithm [44]. The second step, called aggregation, requires a set of stages termed as interpolation, and summation and shift of far-field signatures, which will be the focus of this study. The third step based on the procedures named translation, anterpolation (or filtering), and shift of far-field signatures will again be the focus of this thesis. Last two procedures in the third step are together called as disaggregation.

The MLFMA reduces the computational and memory requirements for solving Helmholtz and Maxwell problems from $O(N^2)$ to $O(N \log N)$. However, there also exists a multiplicative factor inherited in complexity estimate of MLFMA. This multiplicative factor heavily depends on effective implementation of the filters used during interpolation/anterpolation stages as well as the choice of some key parameters, which are the number of multipoles L used to compute MLFMA's translation operators, the over-sampling ratio s used to sample far-field signatures, and the number of interpolation points p used while locally interpolating fields during the upward traversal of the MLFMA tree.

In this thesis, Helmholtz and Maxwell MLFMA are explained in detail. Next, the effective filtering schemes used during interpolation/anterpolation stages are discussed. We noticed that some terms in the formulation of fast vector spherical filter used in Maxwell MLFMA are missing in the literature. Here, the formulation of fast vector spherical filter is derived from the scratch, missing terms in the literature are pointed out, and the implementations of vector and scalar spherical filters are explained in detail. Moreover, although some formulas exist for the optimum choice of the key parameters L, s, and p, they do not yield precise results and are not correct for many cases. Here, we propose a self-tuning library for Helmholtz and Maxwell MLFMA for the optimum choice of these parameters. The features of the self-tuning library are as follows:

- ◊ It enables full control of error in MLFMA,
- ◊ It works for all kinds of cluster sizes and spacings encountered in MLFMA and estimates the optimum parameters precisely,
- ◊ It allows significant memory reduction and time savings in MLFMA,
- ◊ It's executed as a preprocessor to the actual MLFMA call,
- ◊ It requires negligible time and memory for execution.

1.3 Organization of Chapters

In Chapter 2, surface integral equations for solving electromagnetic scattering problems are reviewed. Then, the gist of the FMM, factorization of the free-space Green's function, is discussed and the implementations of the FMM and MLFMA for the solution of surface integral equations are described. In addition, an efficient interpolation scheme is presented to be used for locally interpolating translation operator values in large-scale problems.

In Chapter 3, scalar and vector spherical filters are introduced for calculating field signatures during upward and downward traversals of MLFMA tree. For scalar spherical filter used in Helmholtz MLFMA, the formulation of standard slow method is derived and its complexity is analyzed. Then, the fast method for scalar spherical filtering procedure as well as its implementation are described. Similarly, vector spherical filter utilized in Maxwell MLFMA is discussed by first explaining the standard slow vector spherical filter. Then, the formulation of fast vector spherical filter is derived from scratch by indicating the missing terms in the literature. The implementations of fast scalar and vector spherical filters are elucidated by providing the pseudo codes at the end of the chapter.

Self-tuning library is presented in Chapter 4. A three-stage algorithm for estimating the optimum $L-$parameter is introduced after the behavior of factorized free-space Green's function with respect to varying L is investigated. After the explanation of three-stage algorithm with pseudo-codes, the numerical results that show the accuracy obtained by $L-$parameters estimated by the three-stage algorithm are demonstrated. Afterwards, algorithms for estimating optimum $p-$parameter and $s-$ parameter are introduced. Once the details of the algorithms are provided, the numerical tests that demonstrate the accuracy obtained by p and s parameters estimated by the proposed algorithms are provided.

Extensive test results in multilevel framework are provided in Chapter 5.

CHAPTER 2
FMM AND MLFMA

2.1 Introduction

Applications of the FMM to the Helmholtz equation and Maxwell equation are somewhat different from its applications in other disciplines cited in previous chapter. In applications of FMM to the several problems such as N-body problems, Laplace equation, Poisson equation, etc., the multipole expansions are applied to smooth and slowly varying kernels such as $1/|r-r'|^p$, where p is an integer and $p \geq 1$. In contrast, the application of FMM to the Helmholtz and Maxwell equations necessitates dealing with an oscillatory kernel such as $e^{ik|r-r'|}/|r-r'|$. This oscillatory kernel requires an elaborate mathematical analysis compared to the non-oscillatory ones [9]. For the rigid mathematical background of FMM for the Helmholtz equation, reader is referred to [9, 19, 45, 46].

The FMM for the scalar wave equation was proposed by Coifmann, Rokhlin, and Wandzura [8] to reduce the computational time and memory requirement of the method of moment (MoM) solution of electromagnetic scattering problems from $O(N^2)$ to $O(N^{3/2})$. Then it was used to solve the electric field integral equation (EFIE) [10, 32] and the combined field integral equation (CFIE) [11] for solving scattering from arbitrarily shaped perfect electric conductor (PEC) objects. Later on, multilevel implementation of FMM with the computational complexity and memory requirement of $O(N \log N)$ was introduced by Song and Chew [12, 32]. Different implementations of

MLFMA requiring $O(N \log^2 N)$ operations and storage were presented by Dembart, Yip [47, 48] and Gyure, Stalzer [49].

In this chapter, implementations of FMM and MLFMA to the CFIE are discussed in detail. The rest of this chapter is organized as follows: the integral equations that characterize the EM scattering phenomena from a PEC object are discussed in the second subsection. An approximation to the free-space Green's function that leads to discretization of the integral equations is derived in the third subsection. In the fourth subsection, the implementation of FMM for solving CFIE is described. Multilevel implementation of FMM is explained in the fifth subsection. In the final section, an efficient interpolation scheme for the translation operators is presented.

2.2 Integral Equations

Integral equations of first kind and second kind for solving electromagnetic scattering from an arbitrarily shaped PEC object are briefly reviewed before deriving the formulas for FMM and MLFMA.

Electric field integral equation (EFIE), which characterizes the time-harmonic electric field scattering by a PEC, is given by

$$\frac{ik\eta}{4\pi}\hat{\mathbf{n}}\times\int_{S'}\bar{\mathbf{G}}(\mathbf{r},\mathbf{r}')\mathbf{J}(\mathbf{r}')\,dS' = -\hat{\mathbf{n}}\times\mathbf{E}^i(\mathbf{r}), \tag{2.1}$$

for $\mathbf{r}'\in S'$ and $\mathbf{r}\in S$. Here, $i=\sqrt{-1}$, k is the free space wave number, η is the intrinsic impedance of free space, S is the surface of scatterer, S' is the integration area on the surface scatterer, $\hat{\mathbf{n}}$ is the unit normal vector to the surface S, $\mathbf{J}(\mathbf{r}')$ denotes the unknown induced surface electric current, $\hat{\mathbf{n}}\times$ operator indicates the tangential components of vector fields, $\mathbf{E}^i(\mathbf{r})$ is the incident electric field vector, and

$$\bar{\mathbf{G}}(\mathbf{r},\mathbf{r}') = \left(\bar{\mathbf{I}} - \frac{1}{k^2}\nabla\nabla'\right) g(\mathbf{r},\mathbf{r}'), \tag{2.2}$$

$$g(\mathbf{r},\mathbf{r}') = \frac{e^{ikR}}{R} \ , \ R = |\mathbf{r}-\mathbf{r}'| . \tag{2.3}$$

$\bar{\mathbf{I}}$ represents the unit dyad, and $\bar{\mathbf{G}}(\mathbf{r},\mathbf{r}')$ is called the dyadic form of the free space Green's function for Helmholtz equation ($g(\mathbf{r},\mathbf{r}')$). The unit dyad $\bar{\mathbf{I}}$ indicates the vector potential part of EFIE and the term $\frac{1}{k^2}\nabla\nabla'$ represents the scalar potential part of EFIE.

Magnetic field integral equation (MFIE), which formulates time-harmonic magnetic field scattering by a closed surface PEC, is expressed as

$$2\pi\,\hat{\mathbf{n}}\times\mathbf{J}(\mathbf{r}) - \hat{\mathbf{n}}\times\hat{\mathbf{n}}\times\nabla\times\int_{S'}\bar{\mathbf{G}}(\mathbf{r},\mathbf{r}')\,\mathbf{J}(\mathbf{r}')\,dS' = 4\pi\,\hat{\mathbf{n}}\times\hat{\mathbf{n}}\times\mathbf{H}^i(\mathbf{r}), \tag{2.4}$$

for $\mathbf{r}$ approaches to S from outside. $\mathbf{H}^i(\mathbf{r})$ denotes the incident magnetic field vector.

In theory, MFIE or EFIE can be used to solve the problem of electromagnetic scattering from a closed PEC object. However, both EFIE and MFIE suffer from internal resonance problem due to the presence of homogenous solutions corresponding to interior cavity modes [50]. A remedy to this cavity resonance problem is the combined field integral equation [51]. CFIE is the linear combination of EFIE and MFIE as

$$\text{CFIE} = \alpha\text{EFIE} + (1-\alpha)\frac{i}{k}\text{MFIE} , \quad (0<\alpha<1) . \tag{2.5}$$

The choice of α as 0.2 is proven to be the optimum selection [52]. The method of moments can be applied to the CFIE with Galerkin procedure. To do that, first, the unknown surface current distribution on S is approximated by N basis functions as

$$\mathbf{J}(\mathbf{r}') = \sum_{l=1}^{N} I_l \mathbf{f}_l(\mathbf{r}'), \tag{2.6}$$

where I_l denotes the unknown expansion coefficient, $\mathbf{f}_l(\mathbf{r}')$ denotes the *lth* basis function on the surface of scatterer; the basis functions can be chosen as Rao-Wilton-Glisson (RWG) bases [53], which defines a constant charge density on the surface without charge accumulations on the edges. Second, the basis functions used to discretize

the current $\mathbf{f}_l(\mathbf{r}')$ are tested with the same basis functions $\mathbf{f}_k(\mathbf{r})$, $k=1,...,N$. Testing yields a system of linear equations which can be written in a matrix-vector multiply form as

$$\bar{V} = \bar{\bar{Z}}\,\bar{I}\,, \tag{2.7}$$

where $\bar{V}$ is the excitation vector (N), $\bar{\bar{Z}}$ denotes the impedance matrix ($N \times N$), $\bar{I}$ is the expansion coefficient vector. The impedance matrix element corresponding to EFIE is given as

$$Z_{kl}^{E} = \frac{ik\eta}{4\pi}\int_S \mathbf{f}_k(\mathbf{r})\cdot\int_{S'}\bar{\mathbf{G}}(\mathbf{r},\mathbf{r}')\mathbf{f}_l(\mathbf{r}')\,dS'dS\,, \tag{2.8}$$

and that corresponding to MFIE is expressed as

$$Z_{kl}^{H} = -\frac{1}{2}\int_S \mathbf{f}_k(\mathbf{r})\cdot\mathbf{f}_l(\mathbf{r}')\,dS + \frac{1}{4\pi}\int_S \mathbf{f}_k(\mathbf{r})\cdot\hat{\mathbf{n}}\times\nabla\times\int_{S'} g(\mathbf{r},\mathbf{r}')\mathbf{f}_l(\mathbf{r}')\,dS'dS\,. \tag{2.9}$$

Since the basis functions are defined tangential to the surface of scatterer, the term $\hat{\mathbf{n}}\times$ appears in Eqn.(2.1) and Eqn.(2.4) is omitted here. The contributions from EFIE and MFIE to the impedance matrix of CFIE can be calculated as

$$Z_{kl} = \alpha Z_{kl}^{E} + \eta(1-\alpha)Z_{kl}^{H}\,. \tag{2.10}$$

The excitation vector element is

$$V_k = -\left(\alpha\int_S \mathbf{f}_k(\mathbf{r})\cdot\mathbf{E}^i(\mathbf{r})\,dS + \eta(1-\alpha)\int_S \mathbf{f}_k(\mathbf{r})\cdot\left(\hat{\mathbf{n}}\times\mathbf{H}^i(\mathbf{r})\right)dS\right). \tag{2.11}$$

The unknown expansion coefficient vector $\bar{I}$ is to be computed using an iterative solver (e.g. transpose-free quasi-minimal residuals method).

In next subsection, in lieu of using diagonal forms of translation operator for FMM derivation [9, 45, 46], a simple way that employs addition theorem and an elementary identity is followed to derive multipole approximation.

2.3 Approximating the Green's Function

The FMM relies on approximating Green's function for 3D Helmholtz equation with two elementary identities. These identities can be found in [8, 54-56]. The first identity is a form of Gegenbauer's addition theorem, which is an expansion of Green's function into an infinite series as

$$\begin{aligned}\frac{e^{ik|\mathbf{X}+\mathbf{d}|}}{|\mathbf{X}+\mathbf{d}|} &= ikh_0^1\left(k|\mathbf{X}+\mathbf{d}|\right) \\ &= ik\sum_{l=0}^{\infty}(-1)^l(2l+1)\,j_l(kd)\,h_l^1(kX)\,P_l\left(\hat{\mathbf{d}}\cdot\hat{\mathbf{X}}\right)\end{aligned}, \tag{2.12}$$

or alternatively,

$$\frac{e^{ik|\mathbf{X}+\mathbf{d}|}}{|\mathbf{X}+\mathbf{d}|} = ik4\pi\sum_{l=0}^{\infty}\sum_{m=-l}^{+l}(-1)^l(2l+1)\,j_l(kd)\,h_l^1(kX)\left(Y_l^m\left(\hat{\mathbf{d}}\right)\right)^*Y_l^m\left(\hat{\mathbf{X}}\right), \tag{2.13}$$

where $j_l(x)$ denotes the spherical Bessel function of the first kind, $h_l^1(x)$ is the spherical Hankel function of the first kind, $P_l(x)$ denotes the Legendre polynomial, and $Y_l^m(x)$ is the spherical harmonics function of the degree l and the order m. The expansion is valid for $d<X$, where $d=|\mathbf{d}|$ and $X=|\mathbf{X}|$ (see Figure 2-1 for the definitions of vector quantities). In addition, this expansion can be used to compute the field at a receiver location $\mathbf{r}$ generated by a source at location $\mathbf{r}'$ by simply choosing

$$\mathbf{d}=\mathbf{r}-\mathbf{r}_O+\mathbf{r}_S-\mathbf{r}',\ \mathbf{X}=\mathbf{r}_O-\mathbf{r}_S. \tag{2.14}$$

The condition $d<X$ implies that the $|\mathbf{r}'-\mathbf{r}_S|<X/2$ and $|\mathbf{r}-\mathbf{r}_O|<X/2$ and the source and the observer points $\mathbf{r}'$ and $\mathbf{r}$ must be in the close vicinity of the points $\mathbf{r}_S$ and $\mathbf{r}_O$. Due to these restrictions, the regions of convergence for both the source and the observer points are covered by spheres, namely source sphere and observer sphere which are centered at the points $\mathbf{r}_S$ and $\mathbf{r}_O$, respectively.

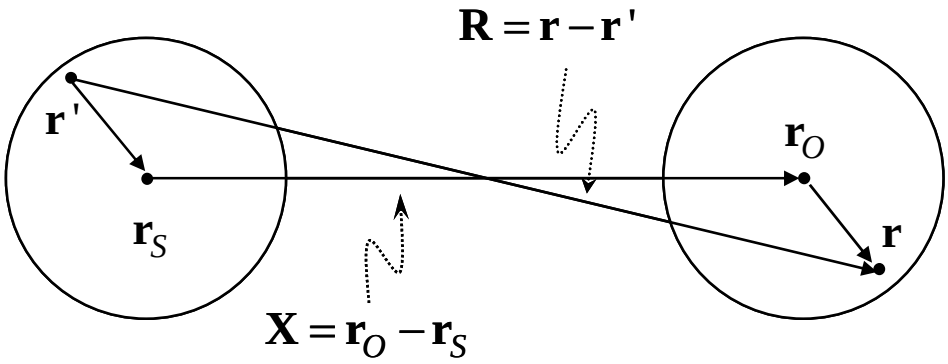

Figure 2-1: The vector quantities illustrating the vector decomposition between the source and the observer locations.

The second identity involves expanding the term $j_l(x)P_l(x)$ into an integral over propagating waves [56] as

$$j_l(kd)P_l\left(\hat{\mathbf{d}}\cdot\hat{\mathbf{X}}\right)=\frac{(i)^{-l}}{4\pi}\int_S e^{i\mathbf{k}\cdot\mathbf{d}}P_l\left(\hat{\mathbf{k}}\cdot\hat{\mathbf{X}}\right)d^2\hat{\mathbf{k}}\,, \tag{2.15}$$

where the integral is evaluated over unit sphere S and $\hat{\mathbf{k}}=\mathbf{k}/|\mathbf{k}|$ denotes the direction of propagating plane waves. Eqn.(2.15) is substituted into Eqn.(2.12) and the order of the summation and the integration is changed as

$$\frac{e^{ik|\mathbf{X}+\mathbf{d}|}}{|\mathbf{X}+\mathbf{d}|}=\frac{ik}{4\pi}\int_S d^2\hat{\mathbf{k}}\;e^{i\mathbf{k}\cdot\mathbf{d}}\sum_{l=0}^{\infty}(i)^l(2l+1)h_l^1(kX)P_l\left(\hat{\mathbf{k}}\cdot\hat{\mathbf{X}}\right). \tag{2.16}$$

When the infinite series is truncated at a degree L and the term $|\mathbf{X}+\mathbf{d}|$ is written more explicitly, one can get a very concise expression for approximating Green's function as

$$\frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|}\approx\frac{ik}{4\pi}\int_S e^{i\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}_O)}T_L(\mathbf{k},\mathbf{X})e^{-i\mathbf{k}\cdot(\mathbf{r}'-\mathbf{r}_S)}d^2\hat{\mathbf{k}}\,, \tag{2.17}$$

where $T_L(\mathbf{k},\mathbf{X})$ is the translation operator defined as

$$T_L(\mathbf{k},\mathbf{X})=\sum_{l=0}^{L}i^l(2l+1)h_l^1(kX)P_l\left(\hat{\mathbf{k}}\cdot\hat{\mathbf{X}}\right). \tag{2.18}$$

In Eqn (2.17), the first exponential term represents the incoming waves to the observer sphere. Similarly, the other exponential term formulates the outgoing waves from the

source sphere. In addition, it is apparent that the integral acts on the Legendre function and exponential terms. In order to find out the most efficient and accurate integration scheme, the exponential term in Eqn.(2.15) can be expanded in terms of special functions as [54]

$$e^{ikd\left(\hat{\mathbf{k}}\cdot\hat{\mathbf{d}}\right)}=\sum_{l=0}^{\infty}(i)^{l}\left(2l+1\right)j_{l}\left(kd\right)P_{l}\left(\hat{\mathbf{k}}\cdot\hat{\mathbf{d}}\right), \tag{2.19}$$

and

$$j_{l}\left(kd\right)P_{l}\left(\hat{\mathbf{d}}\cdot\hat{\mathbf{X}}\right)=\frac{(i)^{-l}}{4\pi}\sum_{l'=0}^{\infty}(i)^{l'}\left(2l'+1\right)j_{l'}\left(kd\right)\int_{S}P_{l}\left(\hat{\mathbf{k}}\cdot\hat{\mathbf{X}}\right)P_{l'}\left(\hat{\mathbf{k}}\cdot\hat{\mathbf{d}}\right)d^{2}\hat{\mathbf{k}}, \tag{2.20}$$

where the integrand is the product of two Legendre functions. By using addition theorem for Legendre functions, the integrand can be expressed in terms of spherical harmonics functions. An exact and efficient surface integration scheme for a function defined over sphere and expanded in terms of spherical harmonics is a hybrid quadrature scheme that uses the Gaussian quadrature rule in the elevation direction and the trapezoidal quadrature rule in the azimuth direction. In this hybrid quadrature scheme, the choice of $2L+1$ equidistant nodes in the azimuth direction and $L+1$ Gauss-Legendre quadrature nodes in the elevation direction yields exact results (accurate upto machine precision). The procedure to integrate a function $f\left(\hat{\mathbf{k}}\right)$ on a unit sphere with this hybrid quadrature scheme can be formulated as

$$\int_{S}f\left(\hat{\mathbf{k}}\right)d\hat{\mathbf{k}}=\frac{2\pi}{2L+1}\sum_{m=1}^{2L+1}\sum_{n=1}^{L+1}w_{n}f\left(\hat{\mathbf{k}}_{mn}\right), \tag{2.21}$$

where $\hat{\mathbf{k}}_{mn}=\left(\sin\theta_{n}\cos\phi_{m},\sin\theta_{n}\sin\phi_{m},\cos\theta_{n}\right)$, $\theta_{n}=\arccos\left(x_{n}\right)$, $\phi_{m}=m\left(2\pi\right)/\left(2L+1\right)$, and x_{n} and w_{n} denote the nodes and weights of Gaussian quadrature. Using this formula, Eqn.(2.17) can be rewritten as

$$\frac{e^{ik\left|\mathbf{r}-\mathbf{r}'\right|}}{\left|\mathbf{r}-\mathbf{r}'\right|}\approx\frac{ik}{4\pi}\frac{2\pi}{2L+1}\sum_{m=1}^{2L+1}\sum_{n=1}^{L+1}w_{n}e^{ik\hat{\mathbf{k}}_{mn}\cdot\left(\mathbf{r}-\mathbf{r}_{O}\right)}T_{L}\left(k\hat{\mathbf{k}}_{mn},\mathbf{X}\right)e^{-ik\hat{\mathbf{k}}_{mn}\cdot\left(\mathbf{r}'-\mathbf{r}_{S}\right)}. \tag{2.22}$$

Instead of employing the hybrid quadrature scheme to integrate the function on unit sphere, another efficient numerical quadrature given in [57] can also be used. Since the integration over sphere is exact due to these quadrature rules, the approximation error in FMM results from truncating the infinite series to a degree L. The optimum choice of the parameter L will be discussed in detail in Chapter 4. The parameter L must be chosen large enough to ensure the rapid convergence of $T_L\left(k\hat{\mathbf{k}}_{mn},\mathbf{X}\right)$ in Eqn. (2.22). Moreover, $L-$parameter could not be selected very large due to the characteristics of the addition theorem and the sampling criteria. In order to satisfy the convergence of partial-wave expansion, one must choose $L > kd = k\left|\mathbf{r}-\mathbf{r}'-\mathbf{X}\right|$. However, it can't be selected to be much larger than kX. This is because the spherical Hankel function $h_l\left(x\right)$ in the translation operator becomes highly oscillatory when $l \gg kX$, causing inaccuracies in the numerical integrations on the unit sphere. Another cause of the inaccuracies when L is much larger than kX is the aliasing due to over-sampling of the far field pattern. On the other hand, when L is smaller than kd, the number of samples selected on the surface is not sufficient to recover the actual pattern. As seen from this discussion, there are lower and upper bounds for the selection of $L-$ parameter in order to use the algorithm with a desired level of accuracy.

2.4 Fast Multipole Method for CFIE

The free space Green's function was approximated with the partial plane-wave expansion (in Eqn.(2.17)) in the previous subsection. In this subsection, implementation of Green's function approximation to the CFIE is discussed.

The free space Green's function approximation can be modified to its dyadic form as

$$\bar{\mathbf{G}}\left(\mathbf{r},\mathbf{r}'\right) \approx \frac{ik}{4\pi}\int_S \left(\bar{\mathbf{I}}-\hat{\mathbf{k}}\hat{\mathbf{k}}'\right)e^{i\mathbf{k}\cdot\left(\mathbf{r}-\mathbf{r}_o\right)}T_L\left(\mathbf{k},\mathbf{X}\right)e^{-i\mathbf{k}\cdot\left(\mathbf{r}'-\mathbf{r}_s\right)}d^2\hat{\mathbf{k}}\,. \qquad (2.23)$$

With the aid of this expression, the radiation pattern of lth basis function is written as

$$\mathbf{F}_l\left(\hat{\mathbf{k}}\right)=k\left(\bar{\mathbf{I}}-\hat{\mathbf{k}}\hat{\mathbf{k}}'\right)\cdot\int_{S'}e^{-i\mathbf{k}\cdot\left(\mathbf{r}'-\mathbf{r}_S\right)}\mathbf{f}_l\left(\mathbf{r}'\right)dS', \tag{2.24}$$

and the receiving pattern of *kth* basis function is stated as

$$\begin{aligned}\mathbf{R}_k\left(\hat{\mathbf{k}}\right)=&\alpha\left(\bar{\mathbf{I}}-\hat{\mathbf{k}}\hat{\mathbf{k}}'\right)\cdot\int_{S}e^{i\mathbf{k}\cdot\left(\mathbf{r}-\mathbf{r}_O\right)}\mathbf{f}_k\left(\mathbf{r}'\right)dS\\&-\left(1-\alpha\right)\hat{\mathbf{k}}\times\int_{S}e^{i\mathbf{k}\cdot\left(\mathbf{r}-\mathbf{r}_O\right)}\mathbf{f}_k\left(\mathbf{r}'\right)\times\hat{\mathbf{n}}dS\end{aligned}. \tag{2.25}$$

It should be noted that since the first term in Eqn.(2.9) is the self-contribution term, it doesn't appear here. And the receiving and radiation patterns have only θ and ϕ components. This is because $\bar{\mathbf{I}}-\hat{\mathbf{k}}\hat{\mathbf{k}}'=\hat{\boldsymbol{\theta}}\hat{\boldsymbol{\theta}}+\hat{\boldsymbol{\varphi}}\hat{\boldsymbol{\varphi}}$ and $\left(\bar{\mathbf{I}}-\hat{\mathbf{k}}\hat{\mathbf{k}}'\right)\cdot\mathbf{V}=V_\theta\hat{\boldsymbol{\theta}}+V_\phi\hat{\boldsymbol{\varphi}}$. Impedance matrix element is

$$Z_{kl}=\frac{-k^2\eta}{16\pi^2}\int_S\mathbf{R}_k\left(\hat{\mathbf{k}}\right)T_L\left(\mathbf{k},\mathbf{X}\right)\mathbf{F}_l\left(\hat{\mathbf{k}}\right)d^2\hat{\mathbf{k}}. \tag{2.26}$$

The excitation vector V_k in Eqn.(2.11) is used here without any change. The acceleration in FMM implementation of CFIE comes from the fact that the radiation patterns of several basis functions in the same source box are translated to the observer box, where several testing functions reside, with the same translation operator. More systematically, the first step in implementation is clustering the scatterer into the boxes. Then, the translation operators, defined between geometrical centers of boxes, are used to translate the radiation patterns of the basis functions within the source boxes into the centers of observer boxes (aggregation). Finally, the receiving patterns at the centers of observer boxes are distributed to the testing functions belonging to the observer boxes (disaggregation). Next, these steps are described in detail.

The domain of scatterer is partitioned into small boxes by a tree structure algorithm. In the beginning, the scatterer is enclosed by a fictitious box. Then, the edges of fictitious box are split into two parts along each direction (*x*, *y*, and *z*), resulting in eight boxes. Each box is recursively subdivided until the size of the edge of the smallest box becomes a fraction of wavelength. Each box contains a number of basis functions and keeps the list of basis functions. If it does not contain any basis function, then it

should be discarded from the list of interacting boxes. The boxes can be numbered with Morton ordering [58]. Figure 2-2 shows the partitioned scatterer (top view).

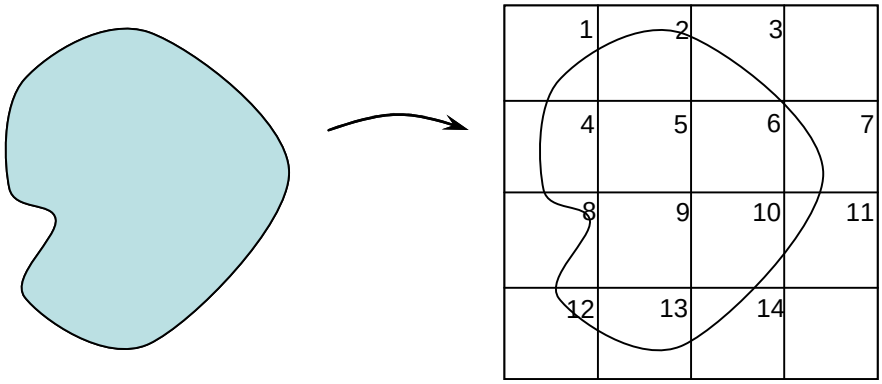

Figure 2-2: The box surrounding the object is subdivided hierarchically in each direction. Each numbered box contains basis functions.

Assume that there exist M boxes in total and B_m, $1 \le m \le M$, represents the box number. The diameter of the smallest sphere enclosing the smallest box is d. The boxes that satisfy the criteria $d < X$ for approximating Green's function given in previous subsection are in the far field of the box B_m. For these boxes, multipole approximation is used to calculate far-field contributions. For the remaining boxes in the near-field of the box B_m, the field contributions are computed with MoM. Therefore, for each box B_m, the boxes are grouped and recorded in the lists of the near-field interaction and the far-field interaction according to their proximity to the box B_m. For each box, the near-field interaction list and the far-field interaction list can be denoted as $I_m'^N$ and $I_m'^F$, respectively. For example, the near-field interaction list of the first box in Figure 2-2, $I_1'^N$, consists of the specifications of the boxes B_2, B_4 and B_5. The far-field interaction list of that box, $I_1'^F$, comprises the specifications of the remaining boxes. Once the box structure and interaction lists are formed, the rest is only evaluating the interactions between boxes. In what follows, the subscription m is used for the test box and the subscription n denotes the source box in the interactions.

First, the radiation patterns of the basis functions are calculated and stored in the memory. The far field of one box can be calculated by

$$\tilde{\mathbf{F}}_n\left(\hat{\mathbf{k}}\right)=\sum_{l\in B_n}\mathbf{F}_l\left(\hat{\mathbf{k}}\right)I_l\,, \tag{2.27}$$

where $\mathbf{F}_l\left(\hat{\mathbf{k}}\right)$ denotes the radiation pattern of each basis function shifted to the geometric center of the box B_n. Next, the radiation patterns of the boxes classified in the far field interaction list of the box B_m are translated to the geometrical center of the box B_m as

$$\mathbf{G}_m\left(\hat{\mathbf{k}}\right)=\sum_{B_n\in I_m'^F}T_L\left(\mathbf{k},\mathbf{X}\right)\tilde{\mathbf{F}}_n\left(\hat{\mathbf{k}}\right), \tag{2.28}$$

Finally, the receiving pattern of the box B_m is obtained as

$$\tilde{\mathbf{R}}_m\left(\hat{\mathbf{k}}\right)=\sum_{l\in B_m}\mathbf{R}_l\left(\hat{\mathbf{k}}\right), \tag{2.29}$$

where $\mathbf{R}_l\left(\hat{\mathbf{k}}\right)$ denotes the receiving pattern of each basis function shifted to the geometric center of the box B_m, is multiplied with the translated pattern and distributed on the basis functions residing in the box B_m as

$$Z_{kl}^{far}I_l=\int_S\tilde{\mathbf{R}}_m\left(\hat{\mathbf{k}}\right)\mathbf{G}_m\left(\hat{\mathbf{k}}\right)d^2\hat{\mathbf{k}}\,. \tag{2.30}$$

Contributions from the boxes near to B_m can be added to matrix-vector multiplication as

$$Z_{kl}I_l\approx Z_{kl}^{far}I_l+Z_{kl}^{near}I_l\,. \tag{2.31}$$

While using the iterative method to solve this matrix equation, the matrix vector multiply is performed at each iteration. Although, the current expansion coefficients are changed at each iteration, the translation operator, the radiation pattern and the receiving pattern of each basis functions remain the same.

The algorithm mentioned above is called single-stage FMM. The analysis in [8, 9] showed that the matrix-vector multiplication at each iteration requires $aNM+bN^2M^{-1}$ floating point operations, where a and b are the parameters involving with the machine specifications and the algorithm implementation. With the choice of $M=\sqrt{bNa^{-1}}$, the overall complexity of the algorithm is computed as $O(N^{3/2})$.

2.5 The Multilevel Fast Multipole Algorithm for CFIE

The complexity of FMM can be further reduced to $O(N \log N)$ by carrying out the procedures in a multilevel framework [32, 49, 50, 59]. Consider a multilevel tree structure with levels numbered between 1 and $\gamma_{\max}$; the first level γ_1 consists of smallest boxes and the last level $\gamma_{\max}$ is the level with the largest boxes, obtained after the first division of the box surrounding the scatterer. In this hierarchy, $B_{n\gamma}$ denotes the n^{th} box at the γth level, the father of the box $B_{m(\gamma-1)}$ if $\gamma \neq \gamma_1$ and the son of the box $B_{m(\gamma+1)}$ if $\gamma \neq \gamma_{\max}$. Note that, here and later on m and n are used to designate different boxes with different numberings. The principal idea behind MLFMA is performing far field interactions at each level of the tree and reducing the number of translations between boxes.

For each box in the tree structure, three different lists are to be kept to execute all interactions. The first list consists of the specifications of the near field neighbors of the box $B_{n\gamma}$ at level γ and denoted as $I'^{N}_{n\gamma}$. The second list $I'^{F}_{n\gamma}$ contains those of the far field neighbor boxes at the level γ whose fathers are the near neighbor of the father of the box $B_{n\gamma}$. The specifications of the basis functions pertinent to the box $B_{n\gamma}$ and the sons of the box $B_{n\gamma}$ are kept in the third list $I'^{S}_{n\gamma}$. The interactions between one sample observer box denoted as the first box at the first level and the remaining boxes are depicted in Figure 2-3 for a three-level FMM. In Figure 2-3 , the sample box is interacting with the near neighbors in level 1 in the list I'^{N}_{11}. The solid diamond boxes are the far neighbors of the sample box and their fathers are the near neighbor of the father of the sample box. Therefore those are kept in I'^{F}_{11}. The remaining large boxes are the neighbors of the grandfather of the sample box. Therefore, the contributions from larger boxes to the basis functions kept in I'^{S}_{11} are directly converted from the grandfather box to the sample box.

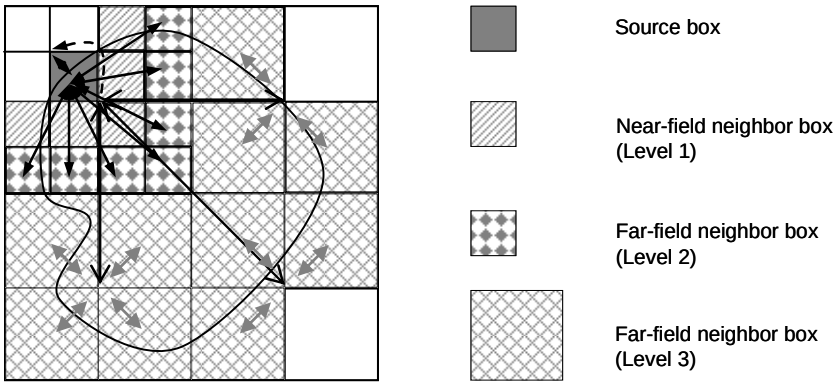

Figure 2-3: Multilevel interactions between an observer box and the remaining source boxes. The interactions with near field neighbors are directly computed by conventional MoM.

At the initialization of the algorithm, the following pre-computations are performed: (i) the near interactions of each box at level 1, B_{m1}, are computed by conventional MoM, (ii) the radiation patterns and the receiving patterns are calculated by Eqn. (2.27) and Eqn. (2.29) and stored for the same boxes, (iii) translation operators $T_L\left(\mathbf{k}, \mathbf{X}_{m\gamma} - \mathbf{X}_{n\gamma}\right)$ between boxes at each level are calculated and kept in memory. However, computation and storage of translation operators require large computational resources when the problem size grows and the number of levels increases. In the next subsection, an efficient interpolation scheme is explained to reduce the memory requirements and the computational time for computing translation operators at all levels.

After the pre-computations are finalized, the radiation patterns at the levels other than the lowest level can be computed by an interpolator. As the box size becomes larger while progressing from the level $\gamma-1$ to the level γ, the number of plane wave directions is increased in order to sample the radiation patterns correctly. This up-sampling procedure can be accomplished by either a local interpolator or a global interpolator. The local interpolator with the complexity $O(N)$ is based on the local interpolation functions such as Lagrange polynomials or the approximate prolate spheroidal functions. The global interpolator with the complexity $O(N \log N)$ is based on spherical harmonics transforms, which is discussed in next chapter. Either a local interpolator or global interpolator can be used to obtain radiation pattern at the box $B_{m\gamma}$ before the radiation pattern of the box $B_{n(\gamma-1)}$ is shifted to the center of its father $B_{m\gamma}$ as

$$\tilde{\mathbf{F}}_{m\gamma}\left(\hat{\mathbf{k}}\right)=e^{i\mathbf{k}\left(\mathbf{X}_{m\gamma}-\mathbf{X}_{n(\gamma-1)}\right)}P_{\gamma-1}^{\gamma}\left(\tilde{\mathbf{F}}_{n(\gamma-1)}\left(\hat{\mathbf{k}}\right)\right). \tag{2.32}$$

where $P_{\gamma-1}^{\gamma}(.)$ denotes the interpolation operator which interpolates the radiation pattern from the level $\gamma-1$ to the level γ. It is clear from the expression that the shifting operation only involves with the multiplication of radiation pattern with an exponential function and does not require any computational resources. This procedure is called up-tree traversal and can be summarized with the following expression

$$\tilde{\mathbf{F}}_{m\gamma}\left(\hat{\mathbf{k}}\right)=\begin{cases}\displaystyle\sum_{l\in B_{m\gamma_{\max}}}\mathbf{F}_l\left(\hat{\mathbf{k}}\right)I_l & \gamma=1\\ \displaystyle\sum_{B_{n(\gamma-1)}\in I_{m\gamma}'^{S}}e^{i\mathbf{k}\left(\mathbf{X}_{m\gamma}-\mathbf{X}_{n(\gamma-1)}\right)}P_{\gamma-1}^{\gamma}\left(\tilde{\mathbf{F}}_{n(\gamma-1)}\left(\hat{\mathbf{k}}\right)\right) & \gamma\neq 1\end{cases}. \tag{2.33}$$

Note that the radiation patterns from the sons of the box $B_{n(\gamma-1)}$ should be shifted, interpolated and summed in order to obtain the radiation pattern at the box $B_{m\gamma}$.

After all radiation patterns at every box are obtained, the next step is sweeping from the highest level to the lowest one. This can be achieved by translating and down sampling (or filtering) the radiation patterns to the boxes at lower levels. Then the field contributions to the basis functions at the lowest level can be obtained. In a succinct way, these operations can be outlined as

$$\mathbf{G}_{m\gamma}\left(\hat{\mathbf{k}}\right)=\begin{cases}\displaystyle\sum_{B_{n\gamma}\in I_{m\gamma}'^{F}}T_L\left(\mathbf{k},\mathbf{X}_m-\mathbf{X}_n\right)\tilde{\mathbf{F}}_{n\gamma}\left(\hat{\mathbf{k}}\right) & \gamma=\gamma_{\max}\\ e^{i\mathbf{k}\left(\mathbf{X}_{m\gamma}-\mathbf{X}_{n(\gamma+1)}\right)}P_{\gamma+1}'^{\gamma}\left(\mathbf{G}_{n(\gamma+1)}\left(\hat{\mathbf{k}}\right)\right)+ & \\ \displaystyle\sum_{B_{n\gamma}\in I_{m\gamma}'^{F}}T_L\left(\mathbf{k},\mathbf{X}_m-\mathbf{X}_n\right)\tilde{\mathbf{F}}_{n\gamma}\left(\hat{\mathbf{k}}\right) & \gamma\neq\gamma_{\max}\end{cases}, \tag{2.34}$$

where $P_{\gamma+1}'^{\gamma}(.)$ denotes the filtering (or down-sampling) operator which filters the field values requiring for level $\gamma+1$ to the field values requiring for level γ. This operator is also defined as the inverse of interpolation operator and named as “anterpolation operator” in [50]. Although the local interpolator described in [50] scales as $O(N)$, a

global filter with complexity $O(N \log N)$ which is used in this study. In addition, using the global filter does not affect the overall complexity MLFMA.

Translated radiation patterns at each level are down sampled to the one coarser level, shifted to the center of the corresponding box and summed with the contributions from the neighbors of that box. One efficient way that minimizes the number of operations requiring in the down-tree traversal is directly down sampling the translated radiation patterns to the lowest level (i.e., shifting all down-sampled fields to the centers of the corresponding boxes and summing the field contributions). After all contributions are added and the integral in Eqn.(2.30) is evaluated for each box , the scattered fields due to provided current expansions are obtained.

2.6 Optimum Local Interpolation Scheme

The translation operator $T_L(\mathbf{k}, \mathbf{X})$ formulated in Section 2.3 requires $(2L+1)(L+1)(L+1)$ operations due to plane wave expansions and infinite series truncation. In FMM algorithm, L is often selected as $\sqrt{N}$ in order to attain $O(N^{3/2})$ overall complexity [8]. Therefore, the computation of translation operator values requires $O(N^{3/2})$ operations. The requiring computational time to compute $T_L(\mathbf{k}, \mathbf{X})$ is negligible when N is small. However, when the problem size grows, the computational time required for directly calculating translation operator values increases significantly.

In lieu of directly computing all translation operator values, an interpolation scheme to calculate all translation operator values from a reduced amount of samples with a defined precision is presented in this subsection. The scheme has $O(N)$ complexity and is described in detail in [60]. There exist also other schemes used to reduce the CPU time for calculating the translation operator [61, 62]. The algorithm in [61] which can be used for interpolating translation operator values and is based on Legendre expansion on Chebyshev nodes has an intricate procedure. However, the scheme presented here is widely used and easy to implement.

The interpolation algorithm is based on the approximate prolate spheriodal series (APS) [63, 64]. The Lagrange polynomials and the Tschebyscheff sampling series can also be employed to interpolate translation operator values. It has been reported in [60] that the optimal interpolation of translation operator can be carried out with APS. The advantage of APS in terms of the computational time compared to the other interpolators has also been proved on a large scale problem in [34].

The idea of computing translation operator values by an interpolator originates from the fact that the translation operator is a band-limited function. This feature of translation operator can be revealed by simply letting $\hat{\mathbf{k}} \cdot \hat{\mathbf{X}} = \cos\theta$ and rewriting translation operator as

$$T_L(\theta, kX) = \sum_{l=0}^{L} i^l (2l+1) h_l^1(kX) P_l(\cos\theta). \tag{2.35}$$

This shows that the translation operator is a band-limited function of θ with order up to L for fixed kX. A rigorous spectrum analysis which has confirmed this feature of translation operator is explained in [60]. It should be noted that any band-limited function can be uniquely reconstructed from its uniformly spaced samples by using sinc function. Nevertheless, computing the function value at one point with sinc interpolator requires taking all sampling points in the band into account. For that reason, the sinc interpolator is called “global interpolator”. One the other hand, APS interpolator, “local interpolator”, allows generating function value at one point by using a few points in the neighborhood of that point. Although calculating translation operator values with sinc interpolator costs $O(N^{3/2})$ operations, calculating those values with APS interpolator takes $O(N)$ computational time. It should be stated that local interpolation scheme is approximate. However, its approximation error can be fully controllable. By properly setting over-sampling ratio s and the number of interpolation points p, the desired level of accuracy in interpolation scheme can be obtained. This will be discussed in Chapter 4.

The formula for interpolating translation operator samples with APS is given as [63]

$$\tilde{T}_L(\theta) = \sum_{m=m_0-p+1}^{m_0+p} T_L(m\Delta\theta)\, S_N(\theta - m\Delta\theta, \theta_0)\, D_M(\theta - m\Delta\theta), \tag{2.36}$$

where $D_M(\theta)$ denotes the periodic sinc function or Dirichlet kernel while $S_N(\theta, \theta_0)$ is the windowing function. The windowing function is defined as

$$S_N(\theta, \theta_0) = \frac{R_N(\theta, \theta_0)}{R_N(0, \theta_0)}, \tag{2.37}$$

$$R_N(\theta, \theta_0) = \frac{\sinh\left[(2N+1)\sinh^{-1}\sqrt{\sin^2(\theta_0/2) - \sin^2(\theta/2)}\right]}{\sqrt{\sin^2(\theta_0/2) - \sin^2(\theta/2)}}. \tag{2.38}$$

The Dirichlet kernel can be expressed as

$$D_M(\theta) = \frac{\sin\left[(2M+1)\theta/2\right]}{(2M+1)\sin(\theta/2)}. \tag{2.39}$$

The terms given in the expressions can be more explicitly described as follows: L is the truncation number or the polynomial degree of translation operator. $M = sL$ is the total number of sampling points, where s is the over-sampling ratio. Sampling points should be equally spaced in the band and the sample spacing must be $\Delta\theta = (2\pi)/(2M+1)$. $N = M - L = (s-1)L$ is the number of over-sampling points. $m_0 = \mathrm{Int}[\theta/\Delta\theta]$ shows the index of the nearest sampling point to the interpolating point. $\theta_0 = p\Delta\theta$ is the width of the window. $p = n/2$ is the number of interpolation points at one side of the interpolating point.

The choice of over-sampling ratio s and the number of interpolation points (or truncation number) p is crucial in APS interpolator. Properly choosing these parameters yields to perform the interpolation procedure in minimum execution time with a preferred level of accuracy. Two algorithms to find the optimum p and s parameters for a provided error tolerance will be presented in Chapter 4.

CHAPTER 3
SCALAR AND VECTOR SPHERICAL FILTERS

3.1 Introduction

MLFMA requires the upward and downward conversion of radiated and incoming fields between consecutive levels, i.e., aggregation and disaggregation. These conversions, from the finer level to the coarser level and from the coarser level to the finer level, are performed via interpolation/anterpolation(filtering), shifting, and summing operations. Shifting and summing operations are exact and linear operations. They do not require significant CPU resources. However, interpolation and filtering operations are based upon intricate schemes and require considerable CPU usage. Therefore, accurate and efficient schemes for interpolating and filtering fields are of paramount importance for the performance and accuracy of aggregation and disaggregation stages.

As briefly mentioned before, there exist two different schemes for interpolating and filtering field patterns: the one is global interpolation and filtering scheme based on the spherical harmonic transforms, the other one is local interpolation and anterpolation scheme based on local interpolation functions. Although, global interpolation and filtering scheme is exact, the local interpolation and anterpolation scheme is an approximate approach and error analysis due to this approach becomes non-trivial as the problem size grows and the levels of MLFMA tree increases [65]. This is because the interpolation and anterpolation procedures are carried out between all consecutive levels and the total error due to the aggregation and the disaggregation stages accumulates at each interpolation and filtering process.

Here, accurate and efficient global interpolation and filtering algorithms are revisited. For interpolating and filtering the scalar fields generated by a cluster of point sources, encountered in Helmholtz MLFMA, the scalar spherical filter is explained. For interpolating and filtering the vector fields due to a cluster of dipoles, which take place in Maxwell MLFMA, the vector spherical filter is described. At the end of the chapter, the pseudo codes for the fast versions of scalar and vector spherical filters are presented.

3.2 The Scalar Spherical Filter

3.2.1 Overview

Filtering a scalar function defined over a sphere is a common problem in many areas such as electromagnetic and acoustic scattering, weather and climate modeling, quantum mechanics, geophysics, and several others. Traditionally, filtering a scalar function via standard spectral transform procedures requires $O(N^3)$ operations, where N is the number of sampling points in the elevation direction. This computational requirement limits the applicability of standard spectral transform procedures to spheres with small number of samples. To filter a scalar function defined over a sphere with a large number of samples, the researchers have proposed several fast filtering schemes for the last decade [61, 66-79] and this research area is still active. The pioneering research on this topic has been conducted by Orzsag and Alpert and the first algorithms were presented in [73] and [61]. Although these algorithms exhibited high performance in low orders and degrees, they suffered from low performance and accuracy at very high orders or degrees [77]. The scheme summarized in [67] requires $O(N^2 \log^2(N^2))$ operations for filtering. It has been reported in [74] that although this scheme is exact, it tends to be lack of stability. Another filtering scheme based on a wavelet approach with complexity $O(N^2(\log(N))^2 \log(1/\varepsilon))$, where ε is the fixed precision, is represented in [71]. However, it has been claimed in [74] that the numerical results presented in [71] don't clearly show the efficiency of proposed method. The filtering scheme in [77], based on the similar ideas in [70], has the complexity of $O(N^2 \log(N) \log(1/\varepsilon))$, showed a

prominent advancement compared to [76], and became a comparable alternative to the work in [68], which has computational complexity of $O(N^2 \log(N))$.

In this subsection, the standard scalar spherical filter (based on standard spectral transforms) and its accelerated version (i.e., the fast scalar spherical filter) are elucidated. The standard method is explained in next subsection. And the fast method proposed in [68] is described in the fast scalar spherical filter subsection.

3.2.2 The Standard Scalar Spherical Filter

The scalar spherical harmonics (SSH), $Y_n^m(\theta,\phi)$, the eigenfunctions of the spherical Laplace operator and sometimes called "tesseral harmonics", of degree $n = 0,1,2,...$ and order $m = -n, -n+1, ..., n-1, n$ at latitude θ and longitude ϕ are defined as [80, 81]

$$Y_n^m(\theta,\phi) = \frac{1}{\sqrt{2\pi}} \bar{P}_n^m(\cos\theta) e^{im\phi}, \tag{3.1}$$

where $\bar{P}_n^m$ denotes normalized associated Legendre functions (NALF) and consists of associated Legendre functions (ALF), P_n^m, i.e.

$$\bar{P}_n^m(\mu) = \sqrt{(n+0.5)\frac{(n-m)!}{(n+m)!}} P_n^m(\mu), \tag{3.2}$$

for $\mu \in [-1,1]$. The square rooted term is called the normalization constant. ALF is defined in terms of the derivatives of Legendre polynomials $P_n(\mu)$ as

$$P_n^m(\mu) = (-1)^m \left(1-\mu^2\right)^{m/2} \frac{d^m}{d\mu^m}\left(P_n(\mu)\right), \tag{3.3}$$

where the phase factor $(-1)^m$ is referred as the Condon-Shortley phase. By using Rodrigues' formula,

$$P_n(\mu) = \frac{1}{2^n n!} \frac{d^n}{d\mu^n}\left[\left(\mu^2-1\right)^n\right], \tag{3.4}$$

more explicit form of NALF is obtained as

$$\bar{P}_n^m(\mu) = \sqrt{(n+0.5)\frac{(n-m)!}{(n+m)!}}\frac{(-1)^m}{2^n n!}\left(1-\mu^2\right)^{m/2}\frac{d^{m+n}}{d\mu^{m+n}}\left(\mu^2-1\right)^n. \tag{3.5}$$

Before discussing spectral transform extensively, a few observations about the algorithms for computing the values of NALF are in order. In literature, a few algorithms were proposed for fast calculation of NALF in double precision arithmetic [82-85]. In our tests, we see that the algorithm prescribed in [86] doesn't calculate the NALF values after degree 160. Another algorithm available in NETLIB relies on the computational schemes given in [83-85] breaks down and stops execution for arguments very near to 1 at the degree 500 and order 413. The algorithm described in [82] and provided in web site [87] accurately calculates NALF values up to very high degrees and orders (e.g. 2800). The algorithm given as built-in function in special function toolbox of MATLAB 7.0, very accurate and stable at very high degrees and orders, was converted to Fortran 90 for test purposes. It has been validated that the code in [87] is noticeably faster than the converted code for calculating all degrees and orders until a requested degree.

The spectral transform relies on the orthogonal projection of a scalar field $f(\theta,\phi)$ tabulated at latitude-longitude grid of a sphere onto the space spanned by SSH. As a square-integrable and band-limited spherical function defined in intervals $\{(\theta,\phi):(0\le\theta\le\pi, 0\le\phi\le 2\pi)\}$, the scalar field $f(\theta,\phi)$ can be expanded as a linear combination of SSH as

$$f(\theta,\phi) = \sum_{n=0}^{N}\sum_{m=-n}^{+n} f_n^m Y_n^m(\theta,\phi), \tag{3.6}$$

where N is the truncation degree and f_n^m denotes spherical harmonic coefficients. Orthogonality relation of SSH,

$$\int_0^{2\pi}\int_0^{\pi} Y_n^m(\theta,\phi)\,Y_{n'}^{m'*}(\theta,\phi)\sin\theta\, d\theta\, d\phi = \delta_{nn'}\delta_{mm'}, \tag{3.7}$$

can be used to obtain the spherical harmonic coefficients after both sides of Eqn.(3.6) are multiplied by $Y_n^{m*}(\theta,\phi)$, where asterisk (*) denotes the complex conjugate, and integrated over the surface of unit sphere, i.e.,

$$f_n^m = \int_0^{2\pi}\int_0^{\pi} f(\theta,\phi) Y_{n'}^{m'*}(\theta,\phi)\sin\theta d\theta d\phi . \tag{3.8}$$

The procedure to compute the spherical harmonic coefficients via Eqn.(3.8), i.e. transformation from grid space to spectral space, is called “spherical harmonics analysis”. The spherical harmonics analysis consists of two steps, namely, forward Fourier transform and forward Legendre transform. After spherical harmonic coefficients are obtained, the operation to compute spherical data over grid points via Eqn.(3.6), i.e. transformation from spectral space to grid space, is called as “spherical harmonics synthesis”. The spherical harmonics synthesis comprises two steps, specifically, backward Legendre transform and backward Fourier transform [88]. The forward Fourier transform, the forward Legendre transform, the backward Legendre transform and the backward Fourier transform steps will be explained in detail after a truncation method for spherical harmonic coefficients is discussed.

To filter or interpolate a scalar field data (e.g. incoming or outgoing fields in Helmholtz MLFMA) from a degree N to a degree K via the standard scalar spherical filter, one can first calculate spherical harmonic coefficients up to degree N and store these in a triangular matrix like

$$f_n^m = \begin{bmatrix} 0 & \cdots & \cdots & 0 & f_0^0 & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & 0 & f_1^{-1} & f_1^0 & f_1^1 & 0 & \vdots & \vdots \\ \vdots & 0 & \iddots & \vdots & \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & \iddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ f_N^{-N} & \cdots & \cdots & \cdots & f_N^0 & \cdots & \cdots & \cdots & f_N^N \end{bmatrix}_{(N+1)\times(2N+1)} . \tag{3.9}$$

For filtering grid space data from the degree N to the degree K, ($N \geq K$), in spectral domain, the sub-matrix with dimensions $(K+1)\times(2K+1)$ whose middle column is aligned with f_n^0 column can be assigned as new truncated spectral data $\tilde{f}_n^m$. For interpolating grid space data from the degree N to the degree K, ($K \geq N$), in spectral

domain, the spectral data matrix f_n^m must be expanded to the dimensions $(K+1)\times(2K+1)$ with keeping the relative positions of spherical harmonic coefficients the same according to f_n^0 column and adding zeros to the rows under the $(N+1)^{\text{th}}$ row. The data in new matrix can be assigned as new truncated spectral data $\tilde{f}_n^m$. For both operations (filtering and interpolation), the scalar field values due to the degree K is obtained on a new grid (θ',ϕ') by performing spherical harmonic synthesis

$$\tilde{f}(\theta',\phi') = \sum_{n=0}^{K} \sum_{m=-n}^{+n} \tilde{f}_n^m Y_n^m(\theta',\phi'). \tag{3.10}$$

In practice, spherical harmonics analysis (Eqn.(3.8)), which can more explicitly stated as

$$f_n^m = \int_0^{\pi} \bar{P}_n^m(\cos\theta)\sin\theta d\theta \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} f(\theta,\phi) e^{-im\phi} d\phi \quad , \tag{3.11}$$

can be carried out in two steps: first, performing a forward Fourier transform of scalar field values along latitudes

$$f^m(\theta) = \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} f(\theta,\phi) e^{-im\phi} d\phi, \tag{3.12}$$

and second, performing the forward Legendre transform to obtain spherical harmonic coefficients as

$$f_n^m = \int_0^{\pi} f^m(\theta) \bar{P}_n^m(\cos\theta)\sin\theta d\theta . \tag{3.13}$$

Eqn. (3.12) can be evaluated via trapezoidal quadrature formula

$$f^m(\theta) = \frac{\sqrt{2\pi}}{I} \sum_{i=1}^{I} f(\theta,\phi_i) e^{-im\phi_i} , \tag{3.14}$$

where I is the number of grid points in one longitude and $\phi_i = 2\pi i / I$ for $i = 1,...,I$. This integration is exact for the band-limit N and the wave numbers $|m| \le N$ and can be

effectively carried out by the fast Fourier transform (FFT) via the routine provided in [89].

The forward Legendre transform expressed in Eqn.(3.13) can be performed using the Gaussian quadrature as

$$f_n^m = \sum_{j=1}^{J} f^m(\theta_j)\bar{P}_n^m(\mu_j)w_j, \tag{3.15}$$

where J denotes the number of grid points along one latitude and $w_1, w_2, ..., w_J$ are the Gaussian weights corresponding to the Gaussian nodes $\mu_j = \cos\theta_j$. Gaussian quadrature is exact for any polynomial of degree $2J-1$ [90].

After using the abovementioned spectral data truncation method and obtaining the truncated coefficients $\tilde{f}_n^m$, the backward Legendre transform is applied to compute filtered or interpolated Fourier coefficients as

$$\tilde{f}^m(\theta') = \sum_{n=|m|}^{K} \tilde{f}_n^m \bar{P}_n^m(\mu'). \tag{3.16}$$

where $\mu' = \cos\theta'$ is defined in new spherical grid (θ', ϕ'). The interpolated or filtered scalar field values can be generated by using a backward fast Fourier transform,

$$\tilde{f}(\theta', \phi') = \frac{1}{\sqrt{2\pi}} \sum_{m=-K}^{K} \tilde{f}^m(\theta') e^{im\phi'}, \tag{3.17}$$

that completes spherical harmonics synthesis. Computational complexity analysis can be performed for each step in the spherical harmonics analysis and synthesis as follows: (i) the forward fast Fourier transforms performed consecutively for all latitudes have cost of $O(JI\log I)$, (ii) the forward Legendre transforms for computing all spherical harmonic coefficients up to degree N require $O(JN^2)$ operations, (iii) the backward Legendre transforms up to degree K scale as $O(PK^2)$, (iv) the backward fast Fourier transforms for all latitudes require $O(PQ\log Q)$ operations. In practice, the number of points along latitude and longitude in the old grid (θ, ϕ) and in the new grid (θ', ϕ') are selected as $J = N+1, I = 2N+1$ and $P = K+1, Q = 2K+1$. Overall complexity of the standard

scalar spherical filter is of $O(N^3)$ for filtering and $O(K^3)$ for interpolation over $O(N^2)$ grid points. In next subsection, a method to reduce overall complexity of the algorithm to $O(M^2 \log M)$ is introduced where $M = K$ or $M = N$ is for interpolation or filtering.

3.2.3 The Fast Scalar Spherical Filter

As clearly seen from the cost analysis of standard scalar spherical filter, the forward and backward Legendre transforms require $O(N^3)$ operations. The Legendre transforms become a bottleneck in performance sight of view during performing interpolation/filtering operations at high degrees (e.g. 100 and above). Therefore, the method for rapid evaluation of Legendre transforms introduced in [68] must be employed to reduce the overall complexity of algorithm from $O(N^3)$ to $O(N^2 \log N)$. The method is simply based on the combination of the forward and backward Legendre transforms and the simplification of sum operator with Christoffel-Darboux formula for NALF. The resulting sum is evaluated with the fast multipole method (see [91]), or with the generalized fast multipole method proposed in [72]. Since the implementations of direct and generalized fast multipole methods for spherical filter are quite technical, the explanations of those are not included here.

Combination of the forward and the backward transforms results in

$$\tilde{f}^m(\theta') = \sum_{n=|m|}^{K} \left(\sum_{j=1}^{J} f^m(\theta_j) \bar{P}_n^m(\mu_j) w_j \right) \bar{P}_n^m(\mu'). \tag{3.18}$$

And interchanging the summations yields

$$\tilde{f}^m(\theta') = \sum_{j=1}^{J} f^m(\theta_j) w_j \sum_{n=|m|}^{K} \bar{P}_n^m(\mu_j) \bar{P}_n^m(\mu'). \tag{3.19}$$

Inner summation can be simplified by Christoffel-Darboux formula defined for NALF as

$$\sum_{n=|m|}^{K} \bar{P}_n^m(\mu_j) \bar{P}_n^m(\mu') = \in_{K+1}^m \frac{\bar{P}_{K+1}^m(\mu') \bar{P}_K^m(\mu_j) - \bar{P}_K^m(\mu') \bar{P}_{K+1}^m(\mu_j)}{(\mu' - \mu_j)}, \tag{3.20}$$

where $\in_n^m = \sqrt{\left(n^2 - m^2\right)/\left(4n^2 - 1\right)}$, which is straightforwardly derived by using the three term recurrence relation [92],

$$\mu \bar{P}_n^m(\mu) = \in_{n+1}^m \bar{P}_{n+1}^m(\mu) + \in_n^m \bar{P}_{n-1}^m(\mu), \tag{3.21}$$

and the mathematical induction for varying degree (see, for example, [93] pg. 318). Substituting Eqn.(3.20) in Eqn.(3.19) yields

$$\tilde{f}^m(\theta') = \sum_{j=1}^{J} f^m(\theta_j) w_j \in_{K+1}^m \frac{\bar{P}_{K+1}^m(\mu') \bar{P}_K^m(\mu_j) - \bar{P}_K^m(\mu') \bar{P}_{K+1}^m(\mu_j)}{(\mu' - \mu_j)}, \tag{3.22}$$

which can be written in a compact form

$$\frac{\tilde{f}^m(\theta')}{\in_{K+1}^m} = \bar{P}_{K+1}^m(\mu') \sum_{j=1}^{J} \frac{f^m(\theta_j) w_j \bar{P}_K^m(\mu_j)}{(\mu' - \mu_j)} - \bar{P}_K^m(\mu') \sum_{j=1}^{J} \frac{f^m(\theta_j) w_j \bar{P}_{K+1}^m(\mu_j)}{(\mu' - \mu_j)}. \tag{3.23}$$

It is apparent from Eqn.(3.23) that the new formulation based on Christoffel-Darboux formula requires $O(JK^2)$ operations. The complexity of this step can be reduced to $O(N^2)$ scale using the direct/generalized fast multipole method applied to the matrix form of Eqn.(3.23). Technical details of these operations can be found in [72, 91]. In case $\mu' = \mu_j$, the quotient can be evaluated with L'Hôpital's rule and the truncated Fourier coefficients can be obtained via

$$\frac{\tilde{f}^m(\theta')}{\in_{K+1}^m} = \frac{d\bar{P}_{K+1}^m(\mu')}{d\mu'} \sum_{j=1}^{J} f^m(\theta_j) w_j \bar{P}_K^m(\mu_j) - \frac{d\bar{P}_K^m(\mu')}{d\mu'} \sum_{j=1}^{J} f^m(\theta_j) w_j \bar{P}_{K+1}^m(\mu_j), \tag{3.24}$$

where the derivative of NALF $d\bar{P}_n^m(\mu')/d\mu'$ is defined as

$$\frac{d\bar{P}_n^m(\mu')}{d\mu'} = -m \frac{\mu'}{1-\mu'^2} \bar{P}_n^m(\mu') + \frac{\left((n+m+1)(n-m)\right)^{1/2}}{\left(1-\mu'^2\right)^{1/2}} \bar{P}_n^{m+1}(\mu'). \tag{3.25}$$

The computations of the Fourier coefficients $f^m(\theta)$ and the truncated Fourier coefficient $\tilde{f}^m(\theta')$ are carried out with the forward and backward FFTs as described

before. The complexities of these operations are of $O(N^2 \log N)$. So, the overall complexity of the algorithm is proportional to $O(N^2 \log N)$.

3.3 The Vector Spherical Filter

3.3.1 Overview

Filtering a vector function defined over a sphere can be accomplished via translating spherical components of that function to the three Cartesian components and invoking the scalar standard/fast spherical filter to filter or interpolate all components individually. However, the vector function expressed with only its traverse components (θ and ϕ), which doesn't have radial component r, can also be filtered via standard/fast vector spherical filter without translation to the Cartesian coordinates. Filtering the vector function with only its transverse components allows to gain from the storage and computational time by a factor 1/3 compared to conventional filtering performed for each component of Cartesian coordinates.

The idea of filtering a vector function with only transverse components by a fast scheme was proposed by Professor Michielssen's former research group [38]. The idea relies on the projection of a vector function to the vector spherical harmonics space and utilization of the fast scalar spherical filter mentioned at previous section. The fast vector spherical filter with complexity $O(N^2 \log N)$ is build upon the formulation of a slow vector spherical filter with complexity $O(N^3)$, which is explained first in this section. Then, the fast vector spherical filter algorithm is described while discussing some minor modifications in order to make it fully consistent with the fast scalar spherical filter.

3.3.2 The Standard Vector Spherical Filter

The vector spherical harmonics (VSH), $\mathbf{\Psi}_n^m(\theta,\phi)$ and $\mathbf{\Phi}_n^m(\theta,\phi)$, of degree $n = 0,1,2,...$ and order $m = -n, -n+1, ..., n-1, n$ at latitude θ and longitude ϕ are defined as [94]

$$\mathbf{\Psi}_n^m(\theta,\phi) = r\nabla Y_n^m(\theta,\phi), \tag{3.26}$$

and

$$\mathbf{\Phi}_n^m(\theta,\phi)=\hat{\mathbf{r}}\times\mathbf{\Psi}_n^m(\theta,\phi), \tag{3.27}$$

where $Y_n^m(\theta,\phi)$ are the scalar spherical harmonics (SSH) as defined before and $r\nabla$ simply denotes the surface gradient operator in spherical coordinates. After surface gradient operator is defined as

$$r\nabla f=\nabla_S f=\frac{\partial f_\theta}{\partial\theta}\hat{\boldsymbol{\theta}}+\frac{1}{\sin\theta}\frac{\partial f_\phi}{\partial\phi}\hat{\boldsymbol{\varphi}}, \tag{3.28}$$

the varieties of VSH can be more explicitly stated as

$$\mathbf{\Psi}_n^m(\theta,\phi)=\frac{\partial Y_n^m(\theta,\phi)}{\partial\theta}\hat{\boldsymbol{\theta}}+\frac{im}{\sin\theta}Y_n^m(\theta,\phi)\hat{\boldsymbol{\varphi}}, \tag{3.29}$$

and

$$\mathbf{\Phi}_n^m(\theta,\phi)=\frac{im}{\sin\theta}Y_n^m(\theta,\phi)\hat{\boldsymbol{\theta}}-\frac{\partial Y_n^m(\theta,\phi)}{\partial\theta}\hat{\boldsymbol{\varphi}}. \tag{3.30}$$

By the assumptions that the varieties of VSH constitutes a complete orthogonal basis, and the vector field $\mathbf{F}(\theta,\phi)$, a square-integrable and band-limited vector spherical function, defined in intervals $\{(\theta,\phi):(0\le\theta\le\pi,0\le\phi\le2\pi)\}$, the vector field $\mathbf{F}(\theta,\phi)$ can be expanded as the linear combinations of varieties of VSH as

$$\begin{aligned}\mathbf{F}(\theta,\phi)&=F_\theta(\theta,\phi)\hat{\boldsymbol{\theta}}+F_\phi(\theta,\phi)\hat{\boldsymbol{\varphi}}\\&=\sum_{n=1}^{N}\sum_{m=-n}^{+n}a_n^m\mathbf{\Psi}_n^m(\theta,\phi)+b_n^m\mathbf{\Phi}_n^m(\theta,\phi)\end{aligned} \tag{3.31}$$

where a_n^m and b_n^m are expansion coefficients analogous to the spherical harmonics coefficients f_n^m in Eqn.(3.6). Orthogonality relations of VSH,

$$\int_0^{2\pi}\int_0^{\pi}\begin{Bmatrix}\mathbf{\Psi}_n^m(\theta,\phi)\cdot\mathbf{\Psi}_n^{m*}(\theta,\phi)\\\mathbf{\Phi}_n^m(\theta,\phi)\cdot\mathbf{\Phi}_n^{m*}(\theta,\phi)\end{Bmatrix}\sin\theta d\theta d\phi=\big(n(n+1)\big)\delta_{nn'}\delta_{mm'}, \tag{3.32}$$

where asterisk (*) denotes complex conjugate, can be used to obtain expansion coefficients a_n^m and b_n^m from the samples of $\mathbf{F}(\theta,\phi)$. With the knowledge of

$$\int_0^{2\pi}\int_0^{\pi}\begin{Bmatrix}\boldsymbol{\Psi}_n^m(\theta,\phi)\cdot\boldsymbol{\Phi}_n^{m*}(\theta,\phi)\\ \boldsymbol{\Phi}_n^m(\theta,\phi)\cdot\boldsymbol{\Psi}_n^{m*}(\theta,\phi)\end{Bmatrix}\sin\theta d\theta d\phi = 0\,, \tag{3.33}$$

both sides of Eqn.(3.31) are multiplied (dot product) by $\boldsymbol{\Psi}_n^{m*}(\theta,\phi)$ and integrated over unit sphere to compute expansion coefficients a_n^m. Same procedure is applied to obtain b_n^m by multiplying $\boldsymbol{\Phi}_n^{m*}(\theta,\phi)$ and resulting expressions are obtained as

$$\begin{Bmatrix}a_n^m\\ b_n^m\end{Bmatrix} = \frac{1}{n(n+1)}\int_0^{2\pi}\int_0^{\pi}\mathbf{F}(\theta,\phi)\bullet\begin{Bmatrix}\boldsymbol{\Psi}_n^{m*}(\theta,\phi)\\ \boldsymbol{\Phi}_n^{m*}(\theta,\phi)\end{Bmatrix}\sin\theta d\theta d\phi\,. \tag{3.34}$$

The procedure to obtain expansion coefficients a_n^m and b_n^m is similar to the procedure explained in standard scalar spherical filter. To this end, this procedure can also be called as spherical harmonic analysis and it consists of the same steps, namely, the forward Fourier transform and the forward Legendre transform, described in Section 3.2.2. After expansion coefficients are computed, they can be truncated by the spectral data truncation method explained in Section 3.2.2. The new truncated expansion coefficients $\tilde{a}_n^m$ and $\tilde{b}_n^m$ are transformed to the grid space by spherical harmonic synthesis, which comprises two steps: the backward Legendre transform and the backward Fourier transform. All these steps are explained in detail as follows.

The spherical harmonic analysis for the vector function can be expressed for each expansion coefficients individually by explicitly writing $\boldsymbol{\Psi}_n^{m*}(\theta,\phi)$ and $\boldsymbol{\Phi}_n^{m*}(\theta,\phi)$ in Eqn.(3.34) as

$$\begin{aligned} a_n^m = {} & \frac{1}{n(n+1)}\int_0^{2\pi}\int_0^{\pi}\frac{1}{\sqrt{2\pi}}\left(F_\theta(\theta,\phi)\frac{\partial\bar{P}_n^m(\cos\theta)}{\partial\theta}e^{-im\phi}\right)\sin\theta d\theta d\phi + \\ & \frac{1}{n(n+1)}\int_0^{2\pi}\int_0^{\pi}\frac{1}{\sqrt{2\pi}}\left(F_\phi(\theta,\phi)\frac{(-im)}{\sin\theta}\bar{P}_n^m(\cos\theta)e^{-im\phi}\right)\sin\theta d\theta d\phi \end{aligned}\,, \tag{3.35}$$

$$b_n^m = \frac{1}{n(n+1)} \int_0^{2\pi} \int_0^{\pi} \frac{1}{\sqrt{2\pi}} \left(F_\theta(\theta,\phi) \frac{(-im)}{\sin\theta} \bar{P}_n^m(\cos\theta) e^{-im\phi} \right) \sin\theta d\theta d\phi - \frac{1}{n(n+1)} \int_0^{2\pi} \int_0^{\pi} \frac{1}{\sqrt{2\pi}} \left(F_\phi(\theta,\phi) \frac{\partial \bar{P}_n^m(\cos\theta)}{\partial\theta} e^{-im\phi} \right) \sin\theta d\theta d\phi \quad . \tag{3.36}$$

Integrations along latitude act on the scalar function values ($F_\theta(\theta,\phi), F_\phi(\theta,\phi)$) and the exponential terms ($e^{-im\phi}$). Therefore, they can be separated from integrations along longitude by defining Fourier coefficients according to theta component of vector field, $f_\theta^m(\theta)$, and phi component of vector field, $f_\phi^m(\theta)$, as

$$\begin{Bmatrix} f_\theta^m(\theta) \\ f_\phi^m(\theta) \end{Bmatrix} = \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} \begin{Bmatrix} F_\theta(\theta,\phi) \\ F_\phi(\theta,\phi) \end{Bmatrix} e^{-im\phi} d\phi = \frac{\sqrt{2\pi}}{I} \sum_{i=1}^{I} \begin{Bmatrix} F_\theta(\theta,\phi) \\ F_\phi(\theta,\phi) \end{Bmatrix} e^{-im\phi_i} \quad , \tag{3.37}$$

where I is the number of grid points in one longitude and $\phi_i = 2\pi i / I$ for $i = 1,...,I$. As stated in Section 4.2.2, Fourier coefficients can be effectively calculated via fast Fourier transform. Substituting Fourier coefficients in Eqn.(3.35)-(3.36) yields

$$a_n^m = \frac{1}{n(n+1)} \int_0^{\pi} \left(\frac{\partial \bar{P}_n^m(\cos\theta)}{\partial\theta} f_\theta^m(\theta) + \frac{(-im)}{\sin\theta} \bar{P}_n^m(\cos\theta) f_\phi^m(\theta) \right) \sin\theta d\theta \,, \tag{3.38}$$

$$b_n^m = \frac{1}{n(n+1)} \int_0^{\pi} \left(\frac{(-im)}{\sin\theta} \bar{P}_n^m(\cos\theta) f_\theta^m(\theta) - \frac{\partial \bar{P}_n^m(\cos\theta)}{\partial\theta} f_\phi^m(\theta) \right) \sin\theta d\theta \,. \tag{3.39}$$

Integrations along latitude can be performed via Gaussian quadrature as

$$a_n^m = \frac{1}{n(n+1)} \sum_{j=1}^{J} \left(\frac{\partial \bar{P}_n^m(\mu_j)}{\partial\theta} f_\theta^m(\theta_j) + \frac{(-im)}{\sqrt{1-\mu_j^2}} \bar{P}_n^m(\mu_j) f_\phi^m(\theta_j) \right) w_j \,, \tag{3.40}$$

$$b_n^m = \frac{1}{n(n+1)} \sum_{j=1}^{J} \left(\frac{(-im)}{\sqrt{1-\mu_j^2}} \bar{P}_n^m(\mu_j) f_\theta^m(\theta_j) - \frac{\partial \bar{P}_n^m(\mu_j)}{\partial\theta} f_\phi^m(\theta_j) \right) w_j \,, \tag{3.41}$$

where J denotes the number of grid points along one latitude and $w_1, w_2, ..., w_J$ are the Gaussian weights corresponding to the Gaussian nodes $\mu_j = \cos\theta_j$.

After utilizing the spectral data truncation method described in Section 4.2.2 and obtaining the truncated expansion coefficients $\tilde{a}_n^m$ and $\tilde{b}_n^m$, the backward Legendre transforms are applied to compute filtered or interpolated Fourier coefficients, $\tilde{f}_\theta^m(\theta')$ and $\tilde{f}_\phi^m(\theta')$, corresponding to scalar components $\tilde{F}_\theta(\theta',\phi')$ and $\tilde{F}_\phi(\theta',\phi')$, as

$$\tilde{f}_\theta^m(\theta') = \sum_{n=|m|}^{K} a_n^m \frac{\partial \bar{P}_n^m(\mu')}{\partial\theta} + b_n^m \frac{(im)}{\sqrt{1-\mu'^2}} \bar{P}_n^m(\mu'), \tag{3.42}$$

$$\tilde{f}_\phi^m(\theta') = \sum_{n=|m|}^{K} a_n^m \frac{(im)}{\sqrt{1-\mu'^2}} \bar{P}_n^m(\mu') - b_n^m \frac{\partial \bar{P}_n^m(\mu')}{\partial\theta}, \tag{3.43}$$

where K is the truncation degree and $\mu' = \cos\theta'$ is defined in new spherical grid (θ', ϕ'). The interpolated or filtered vector field values can now be generated by using the backward fast Fourier transforms as

$$\begin{Bmatrix} \tilde{F}_\theta(\theta',\phi') \\ \tilde{F}_\phi(\theta',\phi') \end{Bmatrix} = \frac{1}{\sqrt{2\pi}} \sum_{m=-K}^{K} \begin{Bmatrix} \tilde{f}_\theta^m(\theta') \\ \tilde{f}_\phi^m(\theta') \end{Bmatrix} e^{im\phi'}. \tag{3.44}$$

This completes spherical harmonics synthesis for vector fields. The computational cost of the forward and backward Fourier transforms is fourfold according to total cost of corresponding steps in standard scalar spherical filter. The forward and backward Legendre transforms require eight times $O(N^3)$ operation and their complexity will be reduced to $O(N^2 \log N)$ by the algorithm explained in next subsection which employs the method described in fast scalar spherical filter subsection.

3.3.3 The Fast Vector Spherical Filter

As seen from the complexity analysis of standard vector spherical filter, Legendre transforms cost $O(N^3)$ operations and this renders the filtering and interpolation operations impossible at high degrees. Therefore, a fast scheme to interpolate and filter

vector fields is needed to retain the overall efficiency of MLFMA algorithm. In this subsection, implementation of the fast scalar spherical filter to the standard vector spherical filter is discussed. The correction terms to make fast scalar spherical filter fully consistent with the standard vector spherical filter are derived.

As seen in spectral data truncation method, the interpolation operation is nothing more than properly padding zeros to the spectral spectrum of field data. During interpolation process, since there is no spectral data content modification other than adding zeros, each scalar component of vector far field can be separately interpolated by simply calling fast scalar spherical filter. However, the same is not true for filtering process. Individually employing fast scalar spherical filter for each scalar component of vector field is not sufficient for filtering operation. It requires an additional operation such as adding proper higher order and degree terms of spectral data (or correction terms) to the truncated spectral data. This extra operation causes from the derivatives of Legendre functions that appear in Legendre transforms in Eqn.(3.35)-(3.36)-(3.42)-(3.43) and is explained in the following paragraphs after an explicit form of derivative of NALF is derived.

Some important recurrence relations for associated Legendre functions and their derivative are [54, 56],

$$\frac{\partial P_n^m(\cos\theta)}{\partial\theta} = \frac{m\cos\theta}{\sin\theta} P_n^m(\cos\theta) - P_n^{m+1}(\cos\theta), \tag{3.45}$$

$$-\sin\theta P_n^{m+1}(\cos\theta) = (n-m)\cos\theta P_n^m(\cos\theta) - (n+m) P_{n-1}^m(\cos\theta), \tag{3.46}$$

$$n\cos\theta P_n^m(\cos\theta) = \frac{n}{2n+1}\left[(n-m+1) P_{n+1}^m(\cos\theta) + (n+m) P_{n-1}^m(\cos\theta)\right]. \tag{3.47}$$

To obtain the explicit form of derivative of associated Legendre function, Eqn.(3.45) can be multiplied by the term $\sin\theta$ as

$$\sin\theta\frac{\partial P_n^m(\cos\theta)}{\partial\theta} = m\cos\theta P_n^m(\cos\theta) - \sin\theta P_n^{m+1}(\cos\theta). \tag{3.48}$$

Then, Eqn.(3.46) can be substituted into Eqn.(3.48) and Eqn.(3.48) can be rewritten as

$$\sin\theta \frac{\partial P_n^m(\cos\theta)}{\partial\theta} = n\cos\theta P_n^m(\cos\theta) - (n+m) P_{n-1}^m(\cos\theta). \qquad (3.49)$$

Third recurrence relation (Eqn.(3.47)) can be employed in Eqn.(3.49) and resulting expression can be written as

$$\sin\theta \frac{\partial P_n^m(\cos\theta)}{\partial\theta} = \frac{n(n-m+1)}{(2n+1)} P_{n+1}^m(\cos\theta) - \frac{(n+1)(n+m)}{(2n+1)} P_{n-1}^m(\cos\theta). \qquad (3.50)$$

This is the explicit expression for the derivative of associated Legendre functions. To obtain a similar expression for the derivative of NALF, one can simply multiply both sides of the equation with the normalization factor as

$$\begin{aligned}\sqrt{(n+0.5)\frac{(n-m)!}{(n+m)!}}\sin\theta \frac{\partial P_n^m(\cos\theta)}{\partial\theta} = &\sqrt{(n+0.5)\frac{(n-m)!}{(n+m)!}}\frac{n(n-m+1)}{(2n+1)} P_{n+1}^m(\cos\theta) \\ &-\sqrt{(n+0.5)\frac{(n-m)!}{(n+m)!}}\frac{(n+1)(n+m)}{(2n+1)} P_{n-1}^m(\cos\theta)\end{aligned} \qquad (3.51)$$

and modify the expression in the view of Eqn.(3.2) as

$$\begin{aligned}\sin\theta \frac{\partial \bar{P}_n^m(\cos\theta)}{\partial\theta} = &\sqrt{\frac{(n+0.5)(n+1+m)}{(n+1.5)(n+1-m)}}\frac{n(n-m+1)}{(2n+1)} \bar{P}_{n+1}^m(\cos\theta) \\ &-\sqrt{\frac{(n+0.5)(n-m)}{(n-0.5)(n+m)}}\frac{(n+1)(n+m)}{(2n+1)} \bar{P}_{n-1}^m(\cos\theta)\end{aligned}. \qquad (3.52)$$

Now, consider only the θ component of vector field and its spherical harmonic representation multiplied by the term $\sin\theta$, i.e.,

$$\sin\theta\, F_\theta(\theta,\phi) = \sum_{n=1}^{N}\sum_{m=-n}^{+n} a_n^m \left(\sin\theta \frac{\partial Y_n^m(\theta,\phi)}{\partial\theta}\right) + b_n^m \left(imY_n^m(\theta,\phi)\right), \qquad (3.53)$$

which can be further expanded with the help of Eqn.(3.52) as

$$\sin\theta\, F_\theta(\theta,\phi)=\sum_{n=1}^{N}\sum_{m=-n}^{+n} a_n^m\sqrt{\frac{(n+0.5)(n+1+m)}{(n+1.5)(n+1-m)}}\frac{n(n-m+1)}{(2n+1)}Y_{n+1}^m(\theta,\phi)$$
$$-\sum_{n=1}^{N}\sum_{m=-n}^{+n} a_n^m\sqrt{\frac{(n+0.5)(n-m)}{(n-0.5)(n+m)}}\frac{(n+1)(n+m)}{(2n+1)}Y_{n-1}^m(\theta,\phi)\quad .\qquad (3.54)$$
$$+\sum_{n=1}^{N}\sum_{m=-n}^{+n} b_n^m\left(imY_n^m(\theta,\phi)\right)$$

To truncate this expansion at a degree K, outer summation variables are changed to k by assigning $n+1=k$ for the first summation. For the second and third summations, the variables are set to $n-1=k$ and $n=k$ respectively as

$$\sin\theta'\,\tilde{F}_\theta(\theta',\phi')=\sum_{k=2}^{K}\sum_{m=-(k-1)}^{+(k-1)}\tilde{a}_{k-1}^m\sqrt{\frac{(k-0.5)(k+m)}{(k+0.5)(k-m)}}\frac{(k-1)(k-m)}{(2k-1)}Y_k^m(\theta',\phi')$$
$$-\sum_{k=0}^{K-1}\sum_{m=-(k+1)}^{+(k+1)}\tilde{a}_{k+1}^m\sqrt{\frac{(k+1.5)(k+1-m)}{(k+0.5)(k+1+m)}}\frac{(k+2)(k+m+1)}{(2k+3)}Y_k^m(\theta',\phi').\qquad (3.55)$$
$$+\sum_{k=1}^{K}\sum_{m=-k}^{+k}\tilde{b}_k^m\left(imY_k^m(\theta',\phi')\right)$$

Expression can be rewritten in a concise way as

$$\sin\theta'\,\tilde{F}_\theta(\theta',\phi')=\sum_{k=0}^{K-1}\sum_{m=-k}^{+k}\tilde{c}_k^m Y_k^m(\theta',\phi')+\sum_{m=-K}^{+K}\left(\tilde{e}_K^m+\tilde{e}_{K+1}^m\right)Y_K^m(\theta',\phi'),\qquad (3.56)$$

where the coefficients $\tilde{c}_k^m$, $\tilde{e}_K^m$ and $\tilde{e}_{K+1}^m$ are

$$\tilde{c}_k^m=\tilde{a}_{k-1}^m\sqrt{\frac{(k-0.5)(k+m)}{(k+0.5)(k-m)}}\frac{(k-1)(k-m)}{(2k-1)}-$$
$$\tilde{a}_{k+1}^m\sqrt{\frac{(k+1.5)(k+1-m)}{(k+0.5)(k+1+m)}}\frac{(k+2)(k+m+1)}{(2k+3)}+,\qquad (3.57)$$
$$\tilde{b}_k^m(im)$$

$$\tilde{e}_K^m=\tilde{b}_K^m(im)+\tilde{a}_{K-1}^m\sqrt{\frac{(K-0.5)(K+m)}{(K+0.5)(K-m)}}\frac{(K-1)(K-m)}{(2K-1)},\qquad (3.58)$$

$$\tilde{e}_{K+1}^{m}=\tilde{a}_{K}^{m}\sqrt{\frac{(K+0.5)(K+1+m)}{(K+1.5)(K+1-m)}}\frac{K(K+1-m)}{(2K+1)}. \tag{3.59}$$

Same procedure is applied to the ϕ component of the vector function. The filtered ϕ component of the vector field and its spherical harmonic expansion can be stated as

$$\sin\theta'\tilde{F}_{\phi}(\theta',\phi')=\sum_{k=0}^{K-1}\sum_{m=-k}^{+k}\tilde{d}_{k}^{m}Y_{k}^{m}(\theta',\phi')+\sum_{m=-K}^{+K}\left(\tilde{f}_{K}^{m}+\tilde{f}_{K+1}^{m}\right)Y_{K}^{m}(\theta',\phi') \tag{3.60}$$

where the coefficients $\tilde{d}_{k}^{m}$, $\tilde{f}_{K}^{m}$ and $\tilde{f}_{K+1}^{m}$ are

$$\begin{aligned}\tilde{d}_{k}^{m}=\tilde{a}_{k}^{m}(im)-\tilde{b}_{k-1}^{m}\sqrt{\frac{(k-0.5)(k+m)}{(k+0.5)(k-m)}}\frac{(k-1)(k-m)}{(2k-1)}+\\ \tilde{b}_{k+1}^{m}\sqrt{\frac{(k+1.5)(k+1-m)}{(k+0.5)(k+1+m)}}\frac{(k+2)(k+m+1)}{(2k+3)}\end{aligned}, \tag{3.61}$$

$$\tilde{f}_{K}^{m}=\tilde{a}_{K}^{m}(im)-\tilde{b}_{K-1}^{m}\sqrt{\frac{(K-0.5)(K+m)}{(K+0.5)(K-m)}}\frac{(K-1)(K-m)}{(2K-1)}, \tag{3.62}$$

$$\tilde{f}_{K+1}^{m}=-\tilde{b}_{K}^{m}\sqrt{\frac{(K+0.5)(K+1+m)}{(K+1.5)(K+1-m)}}\frac{K(K+1-m)}{(2K+1)}. \tag{3.63}$$

First summations at the right hand side of Eqn.(3.56) and (3.60) can be calculated by fast scalar spherical filter algorithm described in Section 3.2.3 . The remaining summation adds the spherical data contributions (or correction terms) resulting from the subsequent two degrees after the degree K in the spectral domain. In case the fast scalar spherical filter is invoked for filtering at degree K, contributions of the following correction terms are also be added to grid data

$$\tilde{e}_{K+1}^{m}=\tilde{b}_{K+1}^{m}(im)+\tilde{a}_{K}^{m}\sqrt{\frac{(K+0.5)(K+1+m)}{(K+1.5)(K+1-m)}}\frac{K(K+1-m)}{(2K+1)}, \tag{3.64}$$

$$\tilde{e}_{K+2}^{m}=\tilde{a}_{K+1}^{m}\sqrt{\frac{(K+1.5)(K+2+m)}{(K+2.5)(K+2-m)}}\frac{(K+1)(K+2-m)}{(2K+3)}, \tag{3.65}$$

$$\tilde{f}_{K+1}^{m} = \tilde{a}_{K+1}^{m}(im) - \tilde{b}_{K}^{m}\sqrt{\frac{(K+0.5)(K+1+m)}{(K+1.5)(K+1-m)}}\frac{K(K+1-m)}{(2K+1)}, \tag{3.66}$$

$$\tilde{f}_{K+2}^{m} = -\tilde{b}_{K+1}^{m}\sqrt{\frac{(K+1.5)(K+2+m)}{(K+2.5)(K+2-m)}}\frac{(K+1)(K+2-m)}{(2K+3)}, \tag{3.67}$$

where m varies between $-K$ and $+K$ for Eqn.(3.64)-(3.66) and between $-(K+1)$ and $(K+1)$ for Eqn.(3.65)-(3.67). It's apparent that the calculation of correction terms calls for the computation of coefficients $\tilde{a}_{K+1}^{m}$, $\tilde{a}_{K+2}^{m}$, $\tilde{b}_{K+1}^{m}$, and $\tilde{b}_{K+2}^{m}$. Note that the square rooted terms in Eqns.(3.61)-(3.67) are missing in the formulation of fast vector spherical filter presented in [38].

In spite the fact that the desired degree of truncation is K, the final spherical data obtained by combining both direct filtering and correction contributions is of degree $K+1$. Therefore, the number of grid points along both directions should be selected due to the degree $K+1$ during spherical harmonic synthesis.

The computational time required to calculate the correction terms is of $O(N^2)$. The overall complexity of algorithm is at the orders of $O(N^2 \log N)$ with FMM accelerator, which is the same as the complexity of the fast scalar spherical filter.

3.4 Implementation

In this section, implementations of the fast scalar spherical filter and the fast vector spherical filter are described in detail via pseudo codes.

Algorithm 1 The fast scalar spherical filter for Helmholtz MLFMA

Input and Output.

Assumption [The scalar field pattern is sampled at $J = N+1$ latitudes and $I = 2N+1$ longitudes. The filtered field pattern is tabulated at $P = K+1$ latitudes and $Q = 2K+1$ longitudes]

Input is the scalar field values sampled at degree N ($f(\theta_1,\phi_1), f(\theta_1,\phi_2),$ ..., $f(\theta_J,\phi_I)$).

Output is the scalar field values sampled at degree K ($\tilde{f}(\theta'_1,\phi'_1), \tilde{f}(\theta'_1,\phi'_2),$ ..., $\tilde{f}(\theta'_P,\phi'_Q)$).

Pre-computations.

Comment [Compute all data that depend on the degrees (N and K) and doesn't depend on the grid data $f(\theta,\phi)$.]

Locate the Gaussian nodes θ_j and θ'_k ($j = 1,...,J$, $k = 1,...,K$) in grids (θ,ϕ), (θ',ϕ').

Calculate the Gaussian weights w_j.

Fill out the matrices $\bar{P}'^{m}_{K+1}(\mu',m)$, $\bar{P}^{m}_{K+1}(\mu,m)$, $\bar{P}'^{m}_{K}(\mu',m)$, and $\bar{P}^{m}_{K}(\mu,m)$, where $\mu' = \cos(\theta')$ and $\mu = \cos(\theta)$, with NALF values at all orders of degrees $K+1$ and K .

Step 1 The forward Fourier transform.

Comment [Convert the scalar field values $f(\theta_1,\phi_1), f(\theta_1,\phi_2),..., f(\theta_J,\phi_I)$ into the Fourier coefficients $f^{-N}(\theta_1),..., f^{-N}(\theta_J)$, $f^{-N+1}(\theta_1),..., f^{N}(\theta_J)$.]

do $j = 1,..., J$

Perform the fast Fourier transform for the scalar field values at the same latitude $f(\theta_j,\phi_1)$, ..., $f(\theta_j,\phi_I)$.

end do

Step 2 and 3 The forward and the backward Legendre transforms.

Comment [Convert the Fourier coefficients $f^{-N}(\theta_1),..., f^{-N}(\theta_J)$, $f^{-N+1}(\theta_1),..., f^{N}(\theta_J)$ into the truncated Fourier coefficients $\tilde{f}^{-K}(\theta'_1),..., \tilde{f}^{-K}(\theta'_P)$, $\tilde{f}^{-K+1}(\theta'_1),..., \tilde{f}^{K}(\theta'_P)$.]

Perform the operation $1/\mu'_p - \mu_j$ for Gaussian nodes in grids (θ,ϕ), (θ',ϕ').

do $m = -K,...,+K$

Calculate $\tilde{f}^{m}$ (Eqn.(3.23)) at each wave number for all nodes.

end do

Step 4 The backward Fourier transform.

Comment [Convert the truncated Fourier coefficients $\tilde{f}^{-K}(\theta'_1),..., \tilde{f}^{-K}(\theta'_P)$, $\tilde{f}^{-K+1}(\theta'_1),..., \tilde{f}^{K}(\theta'_P)$ into the scalar field values $\tilde{f}(\theta'_1,\phi'_1), \tilde{f}(\theta'_1,\phi'_2),..., \tilde{f}(\theta'_P,\phi'_Q)$.]

do $p = 1,..., P$

Perform the fast backward Fourier transform for the truncated Fourier coefficients at the same latitude $\tilde{f}^{-K}(\theta'_p),..., \tilde{f}^{K}(\theta'_p)$.

end do

End of Algorithm

Algorithm 2 The fast vector spherical filter for Maxwell MLFMA

Input and Output.

Assumption [The vector field pattern is sampled at $J=N+1$ latitudes and $I=2N+1$ longitudes. The filtered vector field pattern is tabulated at $P=(K+1)+1$ latitudes and $Q=2(K+1)+1$ longitudes]

Input is the vector field values sampled at degree N ($F_\theta(\theta_1,\phi_1), F_\theta(\theta_1,\phi_2)$, ..., $F_\theta(\theta_J,\phi_I)$, $F_\phi(\theta_1,\phi_1), F_\phi(\theta_1,\phi_2)$, ..., $F_\phi(\theta_J,\phi_I)$).

Output is the vector field values sampled at degree $K+1$ ($\tilde{F}_\theta(\theta_1',\phi_1'), \tilde{F}_\theta(\theta_1',\phi_2')$, ..., $\tilde{F}_\theta(\theta_P',\phi_Q')$, $\tilde{F}_\phi(\theta_1',\phi_1'), \tilde{F}_\phi(\theta_1',\phi_2')$, ..., $\tilde{F}_\phi(\theta_P',\phi_Q')$).

Step 1 The forward Fourier transform.

Comment [Convert the vector field values $F_\theta(\theta_1,\phi_1), F_\theta(\theta_1,\phi_2),,..., F_\theta(\theta_J,\phi_I)$, $F_\phi(\theta_1,\phi_1), F_\phi(\theta_1,\phi_2),..., F_\phi(\theta_J,\phi_I)$ into the Fourier coefficients $f_\theta^{-N}(\theta_1)$, ..., $f_\theta^{-N}(\theta_J)$, $f_\theta^{-N+1}(\theta_1),..., f_\theta^{N}(\theta_J)$, $f_\phi^{-N}(\theta_1),..., f_\phi^{-N}(\theta_J)$, $f_\phi^{-N+1}(\theta_1),..., f_\phi^{N}(\theta_J)$.]

Call the subroutine written for the first step of fast scalar spherical filter separately for θ and ϕ components of the vector field.

Step 2

Comment [Calculate the contributions stemming from the correction terms to the spherical data $\tilde{F}_\theta^C(\theta_1',\phi_1'), \tilde{F}_\theta^C(\theta_1',\phi_2'),..., \tilde{F}_\theta^C(\theta_P',\phi_Q')$, $\tilde{F}_\phi^C(\theta_1',\phi_1'), \tilde{F}_\phi^C(\theta_1',\phi_2'),...,$ ..., $\tilde{F}_\phi^C(\theta_P',\phi_Q')$.]

Compute the coefficients $\tilde{a}_{K+1}^m$, $\tilde{a}_{K+2}^m$, $\tilde{b}_{K+1}^m$, and $\tilde{b}_{K+2}^m$ by using Eqn.(3.40)-(3.41).

Calculate the coefficients $\tilde{e}_{K+1}^m$, $\tilde{e}_{K+2}^m$, $\tilde{f}_{K+1}^m$, $\tilde{f}_{K+2}^m$ in Eqn.(3.64)-(3.67).

Obtain $\tilde{f}_\theta^m(\theta')$ and $\tilde{f}_\phi^m(\theta')$ by employing Eqn.(3.42)-(3.43).

Call the subroutine written for the fourth step of fast scalar spherical filter separately for $\tilde{f}_\theta^m(\theta')$ and $\tilde{f}_\phi^m(\theta')$.

Multiply the outputs by $1/\sin\theta'$ and obtain the spherical data pertaining to the correction terms.

Step 3

Comment [Calculate the contributions stemming from the fast scalar spherical filter to the spherical data $\tilde{F}_\theta^D(\theta_1',\phi_1'),\tilde{F}_\theta^D(\theta_1',\phi_2'),...,\tilde{F}_\theta^D(\theta_P',\phi_Q')$, $\tilde{F}_\phi^D(\theta_1',\phi_1'),\tilde{F}_\phi^D(\theta_1',\phi_2'),...,\ ...,\tilde{F}_\phi^D(\theta_P',\phi_Q')$.]

Multiply the Fourier coefficients $f_\theta^m(\theta)$ and $f_\phi^m(\theta)$ with $\sin\theta$.

Call the subroutines written for the second, third and fourth steps of fast scalar spherical filter separately for $f_\theta^m(\theta)$ and $f_\phi^m(\theta)$.

Multiply the outputs by $1/\sin\theta'$ and obtain the spherical data by directly employing the fast scalar spherical filter.

Step 4

Comment [Obtain the filtered vector field values $\tilde{F}_\theta(\theta_1',\phi_1'),\tilde{F}_\theta(\theta_1',\phi_2'),...,\tilde{F}_\theta(\theta_P',\phi_Q'),\ \tilde{F}_\phi(\theta_1',\phi_1'),\tilde{F}_\phi(\theta_1',\phi_2'),...,\ \tilde{F}_\phi(\theta_P',\phi_Q')$.]

Combine the contributions from correction terms $\tilde{F}_\theta^C(\theta',\phi')$, $\tilde{F}_\phi^C(\theta',\phi')$ and direct filtering $\tilde{F}_\theta^D(\theta',\phi')$, $\tilde{F}_\phi^D(\theta',\phi')$ one by one and obtain the final values $\tilde{F}_\theta(\theta',\phi')$, $\tilde{F}_\phi(\theta',\phi')$.

End of Algorithm

CHAPTER 4
MLFMA SELF-TUNING LIBRARY

4.1 Introduction

The MLFMA requires $O(N \log N)$ computational complexity and memory requirement. However, there also exists a multiplicative factor inherited in computational complexity estimate. The multiplicative factor heavily depends on the choice of some key parameters (mentioned above), namely the truncation number L, the number of interpolation points p, and the over-sampling ratio s. $L-$parameter is used to truncate the infinite series expansion of free-space Green's function. $p-$parameter is employed to determine the number of points in the local interpolation of the far-field signatures. $s-$parameter is used to (over-) sample the far-field signatures. These three parameters are of paramount importance not only in the accuracy but also in the execution time of the MLFMA. Therefore, these parameters should be optimally selected before executing the MLFMA.

In this chapter, a library comprising robust algorithms is introduced to set these parameters automatically. The library is executed as a preprocessor to the actual call of MLFMA. It requires negligible time and allows significant MLFMA savings in terms of time and memory. The library consists of three main algorithms: (i) truncation number L estimator algorithm, (ii) the number of interpolation points p estimator algorithm, (iii) the over-sampling ratio s estimator algorithm.

This chapter consists of two sections. In the first subsection, the truncation number estimator algorithm is discussed. After the nature of addition theorem is

discussed in detail, a three-stage algorithm to locate requiring truncation number L is presented. Test results of the $L-$parameter estimator algorithm are also provided. In the second subsection, estimator algorithms for interpolation parameters, $p-$parameter estimator algorithm and $s-$parameter estimator algorithm, are explained in detail and numerical test results are provided.

4.2 Truncation Number Estimator Algorithm

4.2.1 Overview

In the numerical implementation of FMM, the numerical errors arise from the numerical integration, the processes of aggregation and disaggregation, and the truncation of the infinite series. By using the quadrature rules addressed in Section 2.3, the numerical integration can be performed exactly. Therefore, the round-off error and the error due to translation operator are expected at numerical integration stage [95]. The interpolation and filtering schemes (without FMM accelerator) described in previous chapter are exact and global operations for the aggregation and the disaggregation processes. For that reason, errors inherited in these operations come from the numerical errors due to the arithmetic operations. In case the local interpolation and anterpolation schemes are used in these processes, the error can be fully controllable due to the band-limited nature of radiating fields. Algorithms for effectively controlling error rates at local interpolation and anterpolation schemes will be presented in next section. The error due to the truncation of infinite series (or addition theorem) is hard to control compared to the other error sources in FMM algorithm. In this section, the error control of addition theorem is discussed in detail.

In order to control the error inherited in the addition theorem, the infinite series in Eqn.(2.16) must be carefully truncated. This is because excessive error due to divergent nature of the spherical Hankel function and early error due to the late convergence nature of the spherical Bessel function emerge at different truncation number values and those should be avoided. It is known that as the order of the spherical Hankel function increases and becomes much larger than the argument ($L \gg kX$), the values of spherical Hankel

function grow exponentially and start to oscillate dramatically. This oscillation results in numerical instabilities in floating-point arithmetic [96]. On the other hand, truncating the infinite series with a small L causes a poor approximation and an additional error due to the late convergence of the spherical Bessel function. The spherical Bessel function starts to converge after $L > kd$. Therefore, an optimum truncation number should be selected in the interval $kX \gg L > kd$ due to desired error level.

Previously, the truncation number has been determined by semi-empirical formula [8, 12, 47, 49, 97]. Later on, it was proved that the semi-empirical formula doesn't yield accurate results for many cases and a new formula called "excessive bandwidth" was derived [50, 98]. The excess bandwidth formula provided good error estimate when the spacing between source and observer clusters (or the buffer size) is large enough. However, when the buffer size is small, the excess bandwidth formula lacks of locating the true truncation number. In order to fix this problem, a novel approach was proposed in [99]. The approach hinges upon the classification of the error regions and the employment of the excess bandwidth formula along with the devised formula presented in [100] in the regions where divergent nature of addition theorem occurs.

In this study, an algorithmic approach to determine the optimum truncation number for the desired level of accuracy in FMM is presented. By this approach, the error due to truncation number can be fully controllable and the optimum truncation number due to desired error can be determined regardless of the buffer size. The approach does not require any classification of the error regions and different approaches for different buffer sizes; it can be readily used for any source-observer configuration. Before proceeding the detailed explanation of the algorithm, the nature of the diagonal addition theorem is discussed first in next subsection.

4.2.2 The Nature of the Diagonal Addition Theorem

The error analysis for the truncation of the diagonal addition theorem has been studied by many researchers [47, 95-98, 101]. The diagonal addition theorem given in Chapter 2 can be rewritten as

$$\frac{e^{ik|\mathbf{X}+\mathbf{d}|}}{|\mathbf{X}+\mathbf{d}|} = ik\sum_{l=0}^{\infty}(-1)^l(2l+1)\,j_l(kd)\,h_l^1(kX)\,P_l\left(\hat{\mathbf{d}}\cdot\hat{\mathbf{X}}\right) \approx ik\sum_{l=0}^{L}(-1)^l(2l+1)\,j_l(kd)\,h_l^1(kX)\,P_l\left(\hat{\mathbf{d}}\cdot\hat{\mathbf{X}}\right), \tag{4.1}$$

where $\mathbf{X}=\mathbf{r}_O-\mathbf{r}_S$ is the vector between cluster centers, $\mathbf{d}=\mathbf{r}-\mathbf{r}_O+\mathbf{r}_S-\mathbf{r}'$ is the summation of the local vectors between the centers of groups and the source and observer points ($\mathbf{r}'$ and $\mathbf{r}$). The condition of validity for the theorem is $d<X$. The semi-empirical formula for truncation is given as [8]

$$L\approx kd+\beta\log(kd+\pi), \tag{4.2}$$

where β is the precision constant and d is the diameter of source/observer sphere. When $\beta=1$, the expected accuracy is 10^{-1}. The number of significant digits when $\beta=5$ and $\beta=10$ are six and eleven respectively. The excess bandwidth formula is [98]

$$L\approx kd+1.8\alpha^{2/3}(kd)^{1/3}, \tag{4.3}$$

where $\alpha=\log_{10}(1/\varepsilon)$ and ε is the desired accuracy. It has been reported in [50, 98] that the difference between the estimated L by this formula and the true L varies between -1 and 2 for $1<kd<500$ and $10^{-1}<\varepsilon<10^{-10}$. It should be noted that limits of kX within which the formula is valid were not mentioned. Derivation of this formula gives a brief analysis of the truncation error compared to the other analyses in [47, 101] and can be found in [96, 98]. Moreover, here, the prominent steps in the derivation are summarized to grasp the nature of the addition:

(i) the truncation error given by

$$\frac{e^{ik|\mathbf{X}+\mathbf{d}|}}{ik|\mathbf{X}+\mathbf{d}|}-\sum_{l=0}^{L}(-1)^l(2l+1)\,j_l(kd)\,h_l^1(kX)\,P_l\left(\hat{\mathbf{d}}\cdot\hat{\mathbf{X}}\right)= \sum_{l=L+1}^{\infty}(-1)^l(2l+1)\,j_l(kd)\,h_l^1(kX)\,P_l\left(\hat{\mathbf{d}}\cdot\hat{\mathbf{X}}\right) \tag{4.4}$$

can be approximated by only taking the leading $L+1$ terms into the account.

(ii) The relative error due to truncation at the $(L+1)^{th}$ term,

$$\varepsilon = \frac{\left|(-1)^{L+1}\left(2(L+1)+1\right) j_{L+1}(kd)\, h_{L+1}^{1}(kX)\, P_{L+1}\left(\hat{\mathbf{d}}\cdot\hat{\mathbf{X}}\right)\right|}{\left|\dfrac{e^{ik|\mathbf{X}+\mathbf{d}|}}{ik|\mathbf{X}+\mathbf{d}|}\right|}, \tag{4.5}$$

can be reduced to

$$\varepsilon \approx j_{L+1}(kd)(2L+3) \tag{4.6}$$

by the asymptotic form of spherical Hankel function with the assumption $kd < L+1 < kX$ [54, 55, 96].

(iii) By using the asymptotic form of spherical Hankel function (for large order and argument [96, 98]) and performing the proper approximations, one can get the excess bandwidth formula (Eqn.(4.3)).

It should be noted that the relative error is related to the convergence rate of spherical Bessel function and its argument kd. After $L+1 > kd$, the spherical Bessel function starts to converge so does the relative error of the addition theorem. However, the distance between clusters kX has also effect on the accuracy of the addition theorem because of the ignored spherical Hankel function in the second step. The simplification in the second step is done with the assumption $kd < L+1 < kX$. Nonetheless, after $L+1 > kX$, the Hankel function starts to grow exponentially, becomes a significant factor in error analysis and can't be ignored like done in the second step.

Due to the rapid convergence rate of the spherical Bessel function, the convergent nature of the spherical Bessel function does not meet with the divergent nature of the spherical Hankel function before it hits to the machine precision if the buffer size is large as in Figure 4-1 . Hence, the excess bandwidth formula is valid for large buffer size case. On the other hand, in case the buffer size is small as depicted in Figure 4-2, before the convergent nature of the spherical Bessel function reaches to the machine precision, it interferes with the divergent nature of the spherical Hankel function and the overall error starts to grow exponentially. In this case, the formula is not valid after a certain error rate

beyond its applicable range $kd < L < kX$. The maximum relative error rate and the estimated relative error rate due to varying L at this case are plotted in Figure 4-3.

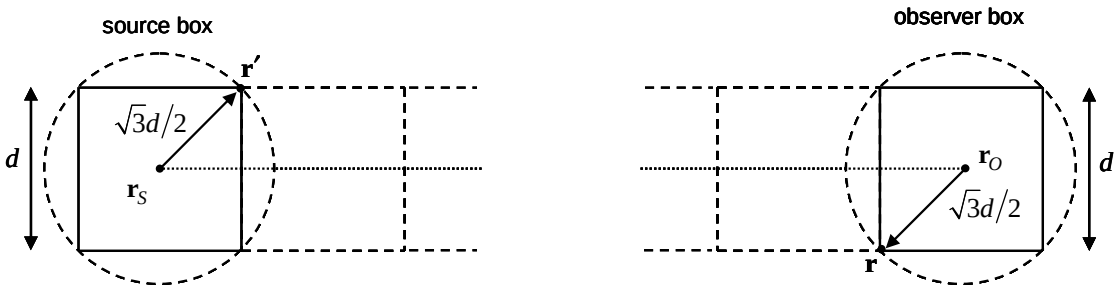

Figure 4-1: Large buffer case is depicted for the worst case analysis in FMM (ten box buffer). Two fictitious spheres, source sphere and observer sphere, enclose the boxes.

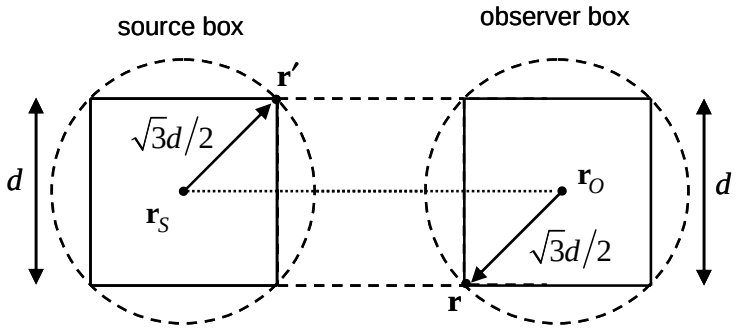

Figure 4-2: Small buffer case is depicted for the worst case analysis in FMM (one box buffer). Two fictitious spheres, source sphere and observer sphere, enclose the boxes.

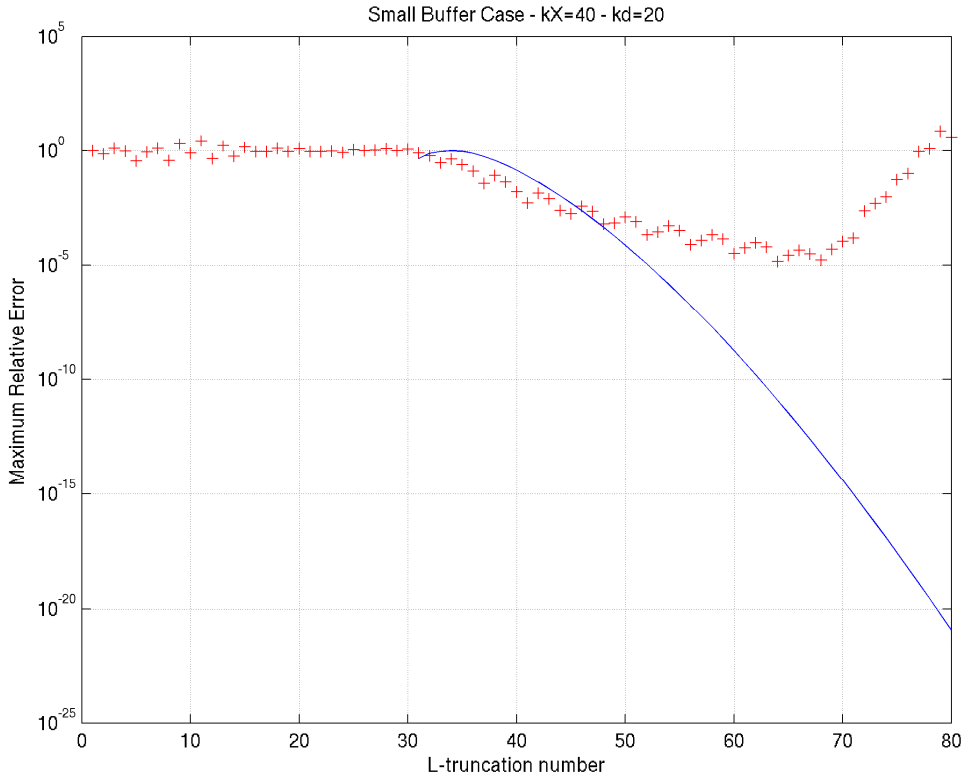

Figure 4-3: The relative error of the addition theorem for a source-observer configuration is plotted with the crosses. Here, kd=20 and kX=40 (one box buffer case).

Other than the buffer size and the cluster size, one additional parameter affects the nature of the diagonal addition theorem. This parameter is related to the relative positions of the source and the observer points in the clusters. In order to investigate the effects of relative positions of the source/observer points on the error rate of the diagonal addition theorem, different source/observer distributions can be tested for the fixed cluster size and buffer size. At the test setup, the source and observer points are located in the directions of the opposite corners of the boxes (See Figure 4-4).

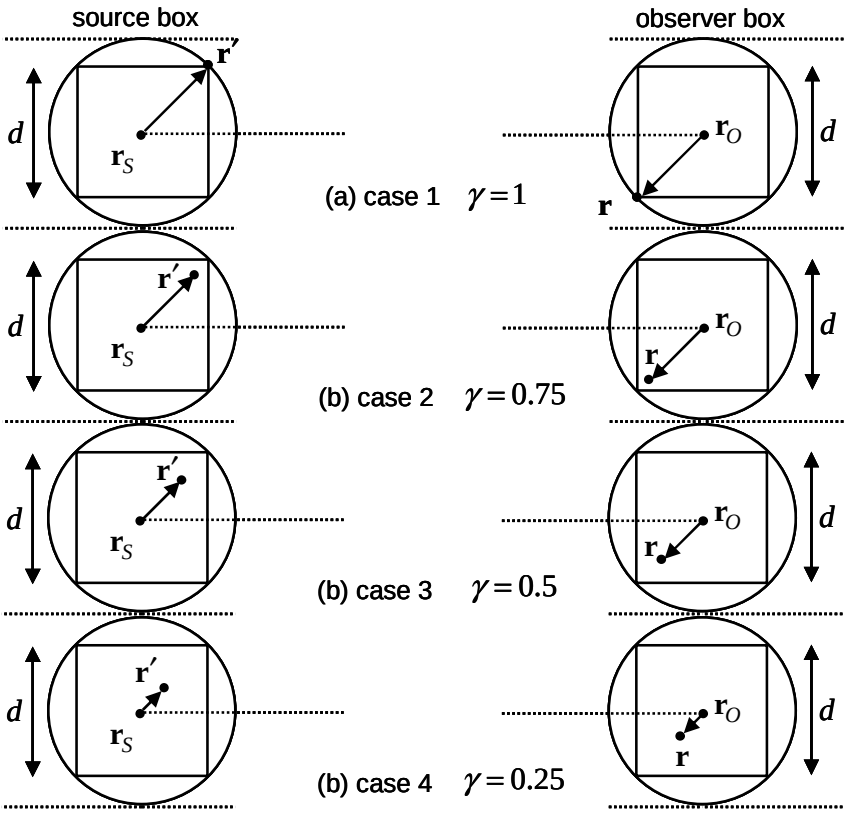

Figure 4-4: Different source/observer distributions are presented to investigate the relative error.

For these four different types of interactions, relative errors due to diagonal addition theorem are examined for two different cases of buffer size: small buffer case and large buffer case. It has been reported in [95] that the maximum relative error occurs when the points are located in the opposite directions and farthest away from the cluster centers. Consequently, the maximum relative error is expected at the case 1 (See Figure 4-4). This prediction is confirmed by the relative error plots in Figure 4-5 and Figure 4-6.

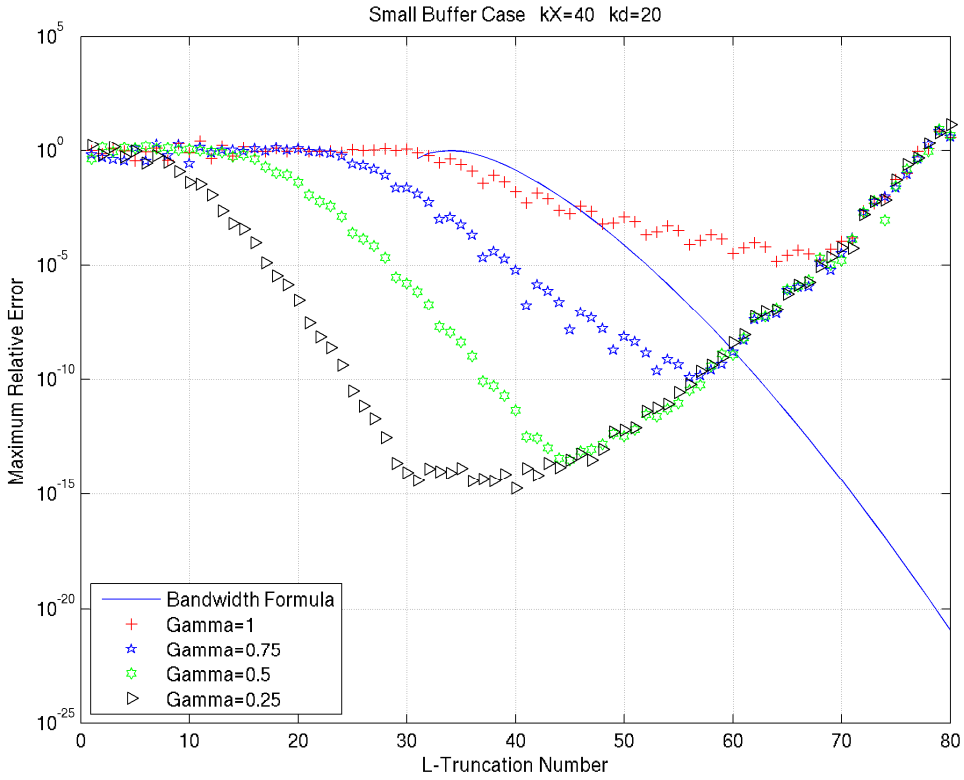

Figure 4-5: Relative errors of different distributions for one buffer case.

Figure 4-5 shows the relative errors of each distribution at test setup (case 1-4) for small-buffer case. Markers indicate the actual errors and solid line shows the estimate by excess bandwidth formula. It is clear from the graph that the Bessel's convergent nature emerge at different truncation number values due to relative positions of the source/observer points. After $L = kX = 40$, divergent nature appears and surpasses the convergent nature. This phenomenon can be clearly observed at the last case ($\gamma = 0.25$). The maximum relative error occurs at the first case ($\gamma = 1$) and it can be considered as the worst case (or the error upper bound) and forms basis in the algorithmic approach presented in next subsection. In the algorithmic approach, the truncation number is selected at the region where the error is decaying.

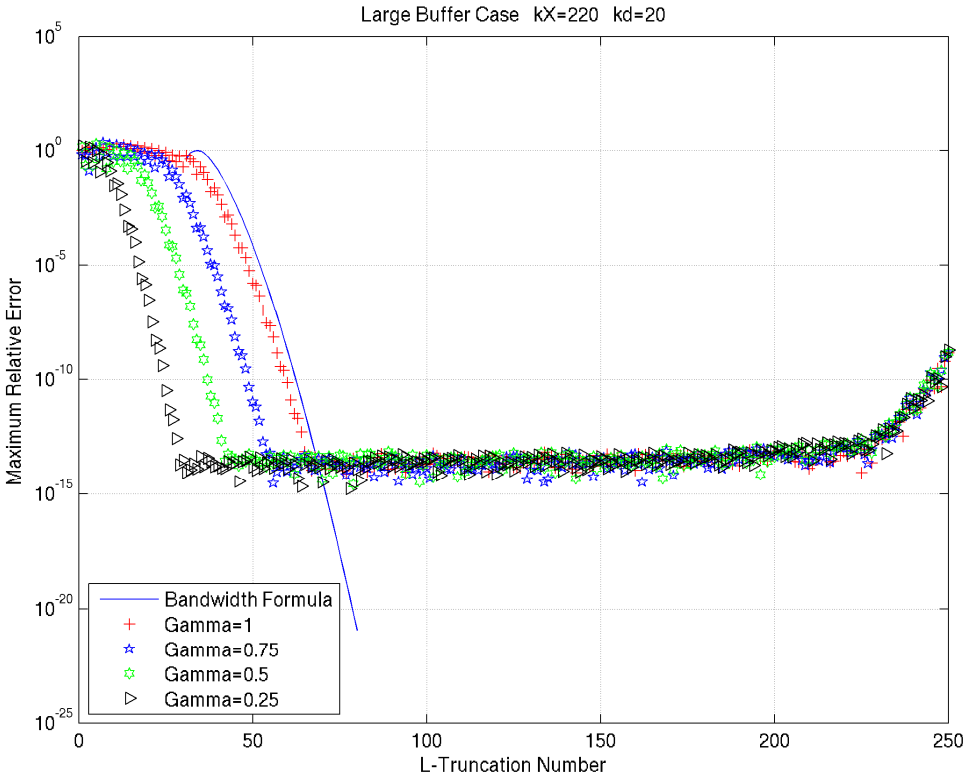

Figure 4-6: Relative errors of different distributions for large buffer case.

Figure 4-6 shows the relative error of each case in the numerical test setup for the large-buffer case. Again, markers show the actual relative errors and the solid line indicates the estimated error due to formula. It's apparent that the convergent nature emerges after $L = kd$, reaches to the machine precision and stays there until $L = kX$. Optimum truncation number should be selected before the convergent nature saturates at the machine precision level. Upper error bound can again be determined via the case 1.

In next subsection, a three-stage algorithm for finding the truncation number L due to given error rate will be described. The algorithm estimates the truncation number regardless of the buffer size.

4.2.3 Algorithm

The truncation number estimator algorithm relies on the utilization of a three-stage algorithm for any source-observer configuration. At the initialization of the algorithm, cluster sizes and buffer size are requested from the user. For the provided

configuration, the worst-case scenario ($\gamma = 1$) is maintained by locating the first source point and the first observer point on the opposite poles. In addition, randomly selected ten points are added over the surface of each sphere. Iteration starts with placing one additional point over clusters. At each iteration, one randomly selected point is added to the distribution over the surface of each cluster and a three-stage algorithm is called to get the estimated L due to given error-rate for the current source-observer distribution. In case the estimated truncation numbers are the same at five consecutive iterations, then the algorithm stops execution and outputs the estimated L.

As seen in previous subsection, error curve for diagonal addition theorem has a bowl-like behavior. Although curve is not smooth and is noisy, the truncation number L due to a desired error level can be estimated at the convergent region of the curve by a robust three-stage algorithm. The first stage in the algorithm is bracketing the global minima of the curve. The second stage is locating the global minimum by golden section search algorithm. The first two stages are performed to determine the limits of the convergent region, $kd < L < \text{global minimum}$. The third one is finding the estimated truncation number by the bisection algorithm in the convergence region.

It should be noted that the convergent part approaches to the machine precision and saturates at this level for the large-buffer case. Because of the noisy nature of the saturation region, where locating the true global minimum is nearly impossible, global minimum at large-buffer case can be considered as the point where the curve hits the machine precision. At small-buffer case, the convergent nature vanishes before reaching to the machine precision and yields a bowl-like curve which has a specific global minimum. At the next two sub-subsection, the bracketing and golden section search algorithms to obtain the limits of convergent region are described in detail. After tools for determining the bounds of convergent region are provided, the bisection root-finding algorithm is summarized for locating the true L corresponding to the desired level of accuracy in this region.

4.2.3.1 The Bracketing Algorithm

Before proceeding to find the global minimum of a unimodal function (See Figure 4-7), the minimum of the function should be bracketed by a triplet (a, b, c). In the

interval bracketed by the triplet, the conditions $a < b < c$, $f(b) < f(a)$ and $f(b) < f(c)$ are to be satisfied.

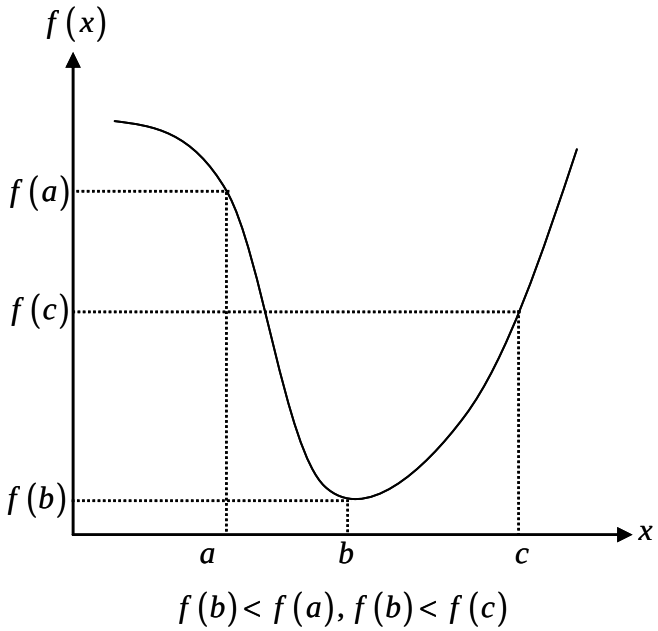

Figure 4-7: A unimodal function whose minimum is bracketed by the triplet (a, b, c).

In order to find the valid triplet that brackets the minima, seeking procedure can be started from the peak of left hill by an initial guess b and proceeded by stepping downhill. Each step size can be increased by a constant factor (defined by golden ratio, as explained below) or by the result of parabolic extrapolation of preceding points. At each step, the condition $f(c) > f(b)$ is checked to find out whether the new interval brackets the minimum or not. If the condition is satisfied, then the triplet (a, b, c) is to be assigned as bracketing triplet.

Parabolic extrapolation is used to expedite the stepping procedure in the downhill direction. Parabolic extrapolation relies on the extrapolation of a point on the curve by fitting a parabola through the triplet (a, b, c). The extrapolated point corresponds to the abscissa of the minimum of the fitted parabola. If the function is nicely fitted by parabola and the conditions $f(x) < f(c)$ or $f(x) > f(b)$ $(\text{while } f(x) > f(c))$ is satisfied, then triplets bracketing the minima can be obtained with a single leap. Otherwise, the minimum can be searched by magnifying the bracket with golden ratios.

Assume a parabola passes through the points (a,b,c) and its algebraic expression is

$$f(x) = A + Bx + Cx^2 \quad , \quad (x = a,b,c). \tag{4.7}$$

Equation provides three linearly independent equations which allow to determine the coefficients $A, B,$ and C. After the coefficients are determined, the abscissa of the minimum of the parabola can be obtained by setting its derivative to zero as

$$\frac{df(x)}{dx} = 0 = 2Cx + B, \tag{4.8}$$

which yields

$$x = \frac{-B}{2C}. \tag{4.9}$$

For finding the expressions of the coefficients B and C in terms of function values and the triplet, one can shift the origin to the point b. By calculating the relative distances of other points with respect to the point b, the linear equations can be written as

$$\begin{aligned} f(a) &= A + Ba' + Ca'^2 \\ f(b) &= A \\ f(c) &= A + Bc' + Cc'^2 \end{aligned} \quad , \tag{4.10}$$

where

$$a' = a - b, \quad c' = c - b. \tag{4.11}$$

By using the elimination method, the coefficients can be obtained as

$$B = \frac{\big(f(a) - f(b)\big)c'^2 - \big(f(c) - f(b)\big)a'^2}{a'c'^2 - c'a'^2}, \tag{4.12}$$

and

$$C = \frac{\big(f(a) - f(b)\big)c' - \big(f(c) - f(b)\big)a'}{c'a'^2 - a'c'^2}. \tag{4.13}$$

Hence, the abscissa of the minimum of the parabola is found at

$$x = \frac{-B}{2C} = \frac{1}{2}\frac{\big(f(a)-f(b)\big)c'^2 - \big(f(c)-f(b)\big)a'^2}{\big(f(a)-f(b)\big)c' - \big(f(c)-f(b)\big)a'}. \tag{4.14}$$

A routine for obtaining the triplet bracketing the minima which uses the default magnification (golden ratio) and the parabolic extrapolation can follow the procedure given below.

Algorithm 1 The bracketing algorithm

Inputs and Outputs.

Inputs: a, initial point, integer.

b, initial guess for the second point in the bracket, integer.

$f(x)$, external real function.

Outputs: a, lower bound of the interval bracketing the minimum, integer.

b, interior point of the interval bracketing the minimum, integer.

c, upper bound of the interval bracketing the minimum, integer.

Pre-definitions.

$Golden = \left(\sqrt{5}+1\right)/2$ (golden ratio by which successive intervals are expanded)

$Limit = 100$ (maximum allowed limit for parabolic fitting)

Core.

Assign $c = b + Golden(b-a)$,

Retrieve the function values $f(a)$, $f(b)$, $f(c)$.

while $\left(f(b) < f(c)\right)$

Compute the abscissa of the minima of the parabola x,

Determine an alternative limit for parabolic fitting by $x_{\lim} = b + Limit(c-b)$.

! Cases start.

if $(b < x < c)$ then

if $\left(f(x) < f(c)\right)$ then

The triplet bracketing the minima is (b, x, c),

Stop execution and output.

else if $\left(f(x) > f(b)\right)$ then

The triplet bracketing the minima is (b, c, x),

Stop execution and output.

end

Assign $x_{new} = c + Golden(c - b)$! no parabolic fit (use golden).

else if $(c < x < x_{\lim})$ then

if $(f(x) < f(c))$ then

Assign $x_{new} = c + Golden(c - b)$,

The triplet is (c, x, x_{new}).

end if

else if $(x > x_{\lim})$ then

! limit the abscissa of minima to the allowed value

Assign $x = x_{\lim}$.

else

! use directly golden ratio, don't use the parabolic approximation

Assign $x_{new} = c + Golden(c - b)$.

end if

the triplet is (b, c, x_{new}).

end while

End of Algorithm

4.2.3.2 The Golden Section Search Algorithm

The golden section search is a widely used one-dimensional minimum/maximum searching technique for finding global minimum/maximum. It was introduced by Jack Kiefer in 1953 [102]. The name of the technique comes from the fact that the algorithm narrows the interval bracketing the minimum with a golden ratio. An approach like bisection algorithm which reduces the length of the interval bracketing the root by the factor of two requires the selection of the mid-point of the interval and the elimination of the one half of the interval at each iteration. However, golden section which reduces the length of interval bracketing the minimum by the factor of three (approximately) requires the selection of two points in the interval and the systematical minimization of the bracketing interval until a tolerance provided by user.

Suppose that the global minimum of a function is bracketed by the algorithm prescribed at previous sub-subsection. The interval bracketing the minima is bounded by the points a and c. Two new points (b and d) are introduced within the interval (See Figure 4-8).

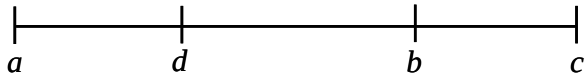

Figure 4-8: Locations of the points in the interval bracketing the global minimum.

Symmetric choice of the interior points leads to

$$b-a=c-d \tag{4.15}$$

The distances can be related with a ratio Γ $(0<\Gamma<1)$ as

$$b-a=\Gamma(c-a) \quad and \quad c-d=\Gamma(c-a). \tag{4.16}$$

This ratio should also be hold for the intervals (a,b) and (d,c) as

$$d-a=\Gamma(b-a) \quad and \quad c-b=\Gamma(c-d). \tag{4.17}$$

Using the first set of equations in the second set yields

$$d-a=\Gamma^2(c-a) \quad \textit{and} \quad c-b=\Gamma^2(c-a). \tag{4.18}$$

Furthermore,

$$b=\Gamma^2 a+\left(1-\Gamma^2\right)c \quad \textit{and} \quad d=\Gamma^2 c+\left(1-\Gamma^2\right)a. \tag{4.19}$$

The first set of equations can be arranged as

$$b=(1-\Gamma)a+\Gamma c \quad \textit{and} \quad d=(1-\Gamma)c+\Gamma a. \tag{4.20}$$

The consistency of the last two equation sets can only be maintained by the condition

$$1-\Gamma^2=\Gamma, \tag{4.21}$$

which yields

$$\Gamma=\frac{\sqrt{5}\pm 1}{2}. \tag{4.22}$$

Since the root $\Gamma=1.618$ violates the condition of being a fraction, the other root $\Gamma=0.618$ should be selected as the solution. If the relative locations of the points b and d are selected according to ratios $\Gamma=0.618$ and $1-\Gamma=0.382$, then, regardless of which triplet is selected ((a,d,b) or (d,b,c)) for the next iteration, relative position of the interior point would remain the same at that bracket. This case is a direct consequence of Eqn.(4.17). By using the retained interior point and introducing a new point in the selected interval, the procedure can be repeated until the length of the interval reaches to a tolerance value provided by the user. This tolerance value couldn't be smaller than the square root of machine precision.

Because of the unimodal property of the function, the selection of correct interval at each iteration can be simply done by comparing the function values. If $f(d)<f(b)$ then the minimum must lie within the interval characterized by the triplet (a,d,b). Otherwise, the minima exists in the interval marked by the triplet (d,b,c). The algorithm for determining the valid interval at each iteration and locating the minimum of the function can be prescribed by the following procedure.

Algorithm 2 The golden section search algorithm

Inputs and Output.

Inputs: a, lower bound of the interval bracketing the minimum, integer.

c, upper bound of the interval bracketing the minimum, integer.

tol, tolerance, integer.

$f(x)$, external real function.

Output: x, the abscissa of the global minimum, integer.

Pre-definitions.

$\Gamma = \left(\sqrt{5}-1\right)/2$ (golden ratio)

Core.

Assign $b = \Gamma(c-a)+a$, $d = (1-\Gamma)(c-a)+a$.

Retrieve the function values $f(b)$ and $f(d)$.

while $abs(c-a) < tol$

if $(f(b) > f(d))$ then

The new bracketing triplet is (a,d,b).

$f(b) = f(d)$.

Assign (a,d,b,c) as $(a, a+(1-\Gamma)(b-a), d, b)$.

Retrieve the new value of $f(d)$.

else

The new bracketing triplet is (d,b,c).

$f(d) = f(b)$.

Assign (a,d,b,c) as $(d, b, d+\Gamma(c-d), c)$.

Retrieve the new value of $f(b)$.

end if

end while

if $\left(f\left(b \right) < f\left(d \right) \right)$ then

The minimum point x is b,

else

The minimum point x is d,

end if

End of Algorithm

4.2.3.3 The Bisection Root Finding Algorithm

Let $f(x)$ be a continuous function in an interval $[a,b]$, such that

$$f(a)\,f(b)<0, \tag{4.23}$$

then there exists at least one zero of the function $f(x)$ on the interval because the sign of the function changes. Suppose that the function has only one root on the interval as depicted in Figure 4-9.

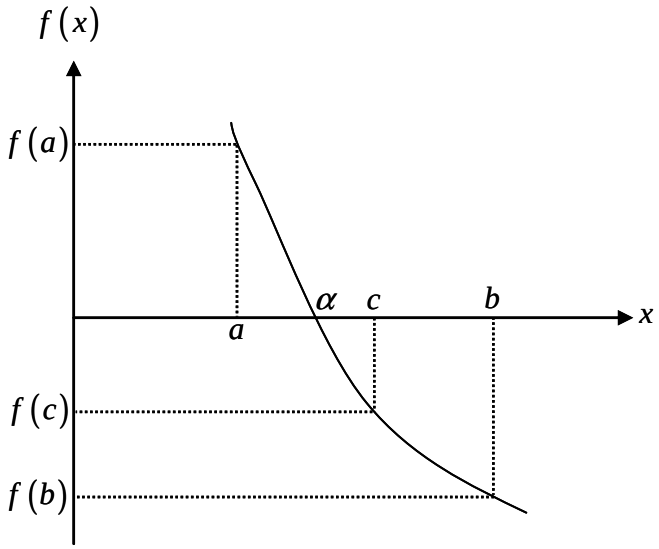

Figure 4-9: A smooth continuous function has a root on the interval *[a,b]* . *c* is the midpoint of the interval.

This root can be located by successively halving the interval until a smaller interval provided by user in which α must lie. The procedure is initiated by defining the midpoint of the interval as $c=(a+b)/2$ and checking the sign of the product $f(c)\,f(b)$. In case the sign of the product is minus, then the root resides in the interval $[c,b]$. Otherwise, the root is in the interval $[a,c]$. Hence, a new interval containing the root α is determined. This procedure is repeated until the new root is located with a desired precision ε, that is

$$|a_n-b_n|<\varepsilon \tag{4.24}$$

where the endpoints of the *nth* interval are a_n and b_n. Although the function is assumed to be smoothly decreasing, the function has a different characteristic and can have a horizontal tangent near to its root. For this kind of case, inclusion of the stopping criteria

$$\left| f\left(a_n\right) \right| < \varepsilon \tag{4.25}$$

expedites the procedure. The first stopping criteria (Eqn.(4.24)) guarantees the convergence to the root with n number of iterations,

$$n = \log_2 \frac{\varepsilon_0}{\varepsilon} \tag{4.26}$$

where ε_0 is the size of the initial bracket. The bisection method described up to this point is slow compared to its alternatives, but it promises the success at many cases at which the convergence cannot be guaranteed by others. In L estimator algorithm, the bisection method is employed after the limits of the convergence region are drawn by the algorithms described in Section 4.2.3.1 and Section 4.2.3.2. Within these limits, estimated L can be located by shifting the threshold to the desired accuracy as if the root sits at a point in the level $f\left(x\right) = \varepsilon_{desired}$ instead of the level $f\left(x\right) = 0$. The outline of the bisection root-finding algorithm can be provided as follows

Algorithm 3 The bisection root finding algorithm

Inputs and Output.

Inputs: a, lower bound of the interval, integer.

b, upper bound of the interval, integer.

tol, tolerance, integer.

$\varepsilon_{desired}$, desired error level, real

$f(x)$, external real function.

Output: x, the abscissa of the root (L corresponding to desired accuracy), integer.

Pre-definitions.

$J_{\max}$, maximum allowed number of iterations, integer.

Core.

do $j = 1, ..., J_{\max}$

Assign the midpoint $c = (b-a)/2$ and retrieve $f(c)$.

if $\left(\left(f(a) - \varepsilon_{desired} \right) \left(f(c) - \varepsilon_{desired} \right) < 0 \right)$ then

The new interval is (a, c).

Assign $b = c$.

else

The new interval is (c, b).

Assign $a = c$.

end if

if $\left(|b-a| < tol \text{ or } f(a) - \varepsilon_{desired} < 0 \right)$ then exit the loop

end do

Assign $x = a$.

End of Algorithm

4.2.3.4 The Accelerators

The algorithms summarized until this point require negligible time compared to the overall execution time of the MLFMA algorithm. Timings of algorithm at different numerical cases are presented in Section 4.2.4. Besides that, the time spent on these routines can be further reduced by utilizing the accelerators. Two different kinds of accelerators can be implemented into existing routines. The first one can be developed to avoid excess steps in bracketing and golden section search algorithms. The second one can be used to expedite the calculation of translation operator by spherical harmonics transforms.

The first kind of accelerator can be implemented with "if cases". Consider the large-buffer case discussed in Section 4.2.2. For this case, locating the minimum point of the curve is actually hard due to noisy nature of the function at machine precision and needless. In addition, locating the minimum is an insignificant process for many cases because the FMM algorithm is often executed with the accuracy varying 10^{-1} through 10^{-11} and the minimum of the curve would remain well below the desired error level. The execution of the bracketing algorithm can be stopped after the retrieved values of the function become smaller than the desired error level during stepping downhill procedure. Stopping execution with "if cases" also avoids to step in the saturation level which appears at level 10^{-13} and below due to the distribution of source/observer points. After the execution is stopped, the bisection algorithm can be invoked to obtain the true truncation number. This accelerator could yield significant time savings in the L estimator algorithm due to the wanted error rate.

The second kind of accelerator based on spherical harmonics transforms can be used for rapid calculation of the translation operator. As mentioned earlier, L estimator algorithm utilizes aforementioned 3-stage algorithm to locate the truncation number for source/observer distribution at each iteration. During each iteration, the routine which calculates the relative error is called many times with different truncation numbers. The relative error computation is performed by calculating the relative differences between the field values obtained by Green's function and the field values computed with FMM

approximation. The field values due to Green's function can be computed and stored at the beginning of each iteration. However, the field values due to FMM approximation should be calculated for different truncation numbers at each iteration. Since the number of directions change due to truncation number, the outgoing/incoming wave expansion matrices and the translation operator vector are supposed to be computed at each calling of relative error routine with different truncation number. Direct computation of outgoing/incoming wave expansion matrices is necessary. Besides that, the translation operator values can be calculated in a fast way by a local interpolator after its equispaced samples are calculated with a spectral domain technique explained below.

The translation operator values can be generated by a local interpolator like far-field values. Because of the band-limited nature of the translation operator, its values can also be generated via spherical harmonics transforms mentioned at previous chapter. By using the forward spherical harmonics transform, the spherical harmonic coefficients of translation operator can be obtained. And these coefficients can be filtered until a degree K and used in the backward spherical harmonics transform to obtain new translation operator values at polynomial degree K less than L. New polynomial degree L should be smaller than previous degree K because the spherical harmonics coefficients, spectral domain representation of a function at polynomial degree L, are not sufficient to recover the spherical data at polynomial degree K ($K > L$). Therefore, the first step in spectral domain technique is calculating the spectral domain spherical harmonic coefficients due to translation operator values at a very high polynomial order, say $L = 1500$. These coefficients are kept in memory, filtered and used in backward spherical harmonic transforms for generating the translation operator values due to different truncation degrees. It can be claimed that only the zeroth order spherical harmonics coefficients are needed to characterize the translation operator in spectral domain. This can be directly proven by the following derivation.

The translation operator given as

$$T_L\left(\cos(\theta)\right) = \sum_{l=0}^{L} i^l \left(2l+1\right) h_l^1\left(kX\right) P_l\left(\cos(\theta)\right), \tag{4.27}$$

can be substituted with $f\left(\theta,\phi\right)$ in the expression of forward spectral transform as

$$f_n^m = \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} \int_0^{\pi} P_n^m\left(\cos(\theta)\right) e^{-im\phi} T_L\left(\cos(\theta)\right) \sin(\theta)\, d\theta d\phi$$
$$= \frac{1}{\sqrt{2\pi}} \int_0^{\pi} P_n^m\left(\cos(\theta)\right) T_L\left(\cos(\theta)\right) \sin(\theta)\, d\theta \int_0^{2\pi} e^{-im\phi} d\phi \quad , \tag{4.28}$$

where the integration along ϕ can be analytically evaluated as

$$\int_0^{2\pi} e^{-im\phi} d\phi = \frac{1 - e^{-i2\pi m}}{im} = \begin{cases} 2\pi, & m = 0 \\ 0, & m \neq 0 \end{cases}, \tag{4.29}$$

which shows that the zeroth order coefficients constitute the spectral data. The forward spectral transform can be further proceeded by explicitly writing the translation operator in forward Legendre transform as

$$f_n^0 = \sqrt{2\pi} \sum_{l=0}^{L} i^l (2l+1) h_l^1(kX) \int_0^{\pi} P_n\left(\cos(\theta)\right) P_l\left(\cos(\theta)\right) \sin(\theta)\, d\theta \,, \tag{4.30}$$

where the integration along θ can be analytically evaluated as

$$\int_0^{\pi} P_n\left(\cos(\theta)\right) P_l\left(\cos(\theta)\right) \sin(\theta)\, d\theta = \begin{cases} \frac{2}{2l+1} & n = l \\ 0 & n \neq l \end{cases}. \tag{4.31}$$

Hence, the spherical harmonics coefficients can be analytically calculated without FFT and Gaussian quadrature. The final expression for spherical harmonics coefficients is

$$f_n^0 = 2\sqrt{2\pi} \sum_{l=0}^{L} i^l h_l^1(kX) \,. \tag{4.32}$$

Once the spherical harmonics coefficients are obtained and stored in the memory, the rest is filtering those coefficients for desired number of truncation number and using those for generating the translation operator values. Needless to say, the backward spectral transform relates with the operations at zeroth order of associated Legendre polynomials and backward Fourier transform. More explicitly, the backward spherical transform can be simplified due to zeroth order coefficients as

$$T_K\left(\cos(\theta)\right) = \frac{1}{\sqrt{2\pi}} \sum_{n=0}^{K} P_n^0\left(\cos(\theta)\right) f_n^0 \ . \tag{4.33}$$

The local interpolator described in Section 2.6 can be used after equispaced samples of translation operator are generated with Eqn.(4.33).

4.2.4 Numerical Results

Three-stage truncation number estimator algorithm has been implemented with accelerators and tested for many cases. Some of test results are provided here.

In the worst-case analysis of the large and small buffer cases, the source and observer points are positioned on the opposite corners of the source and observer boxes. Moreover, in case the source and observer points are distributed over the sphere enclosing the boxes, this yields a better "worst case analysis", provides a safe margin between actual error and estimated error and lifts up the upper bound of estimated error. For that reason, in the test of $L-$parameter estimator algorithm, the source and observer points are scattered over the surfaces of the spheres, namely over the grids on the surfaces. The grid points are defined by outer product of equispaced $N+1$ points in the elevation direction and equispaced $2N+1$ points in the azimuth direction. In the following tests, N is selected as 7 and totally 120 points are chosen over the surface of each sphere.

In the first test, the parameters are selected as $kd = 20$ and $kX = 40$ where d denotes the edge length of the boxes located in the spheres again. This configuration has been shown in Section 4.2.2. Truncation error due to $L-$parameter is plotted in Figure 4-10. Obtained truncation number values for given error rates ε are tabulated in Table 4-1 with the computational time consumed in three-stage routine. Obviously, time spent in the routine decreases with the help of accelerators while the desired error level is increased. The routine stops execution for the desired error rates smaller than 10^{-5} after it decides that the desired error rate is not reachable.

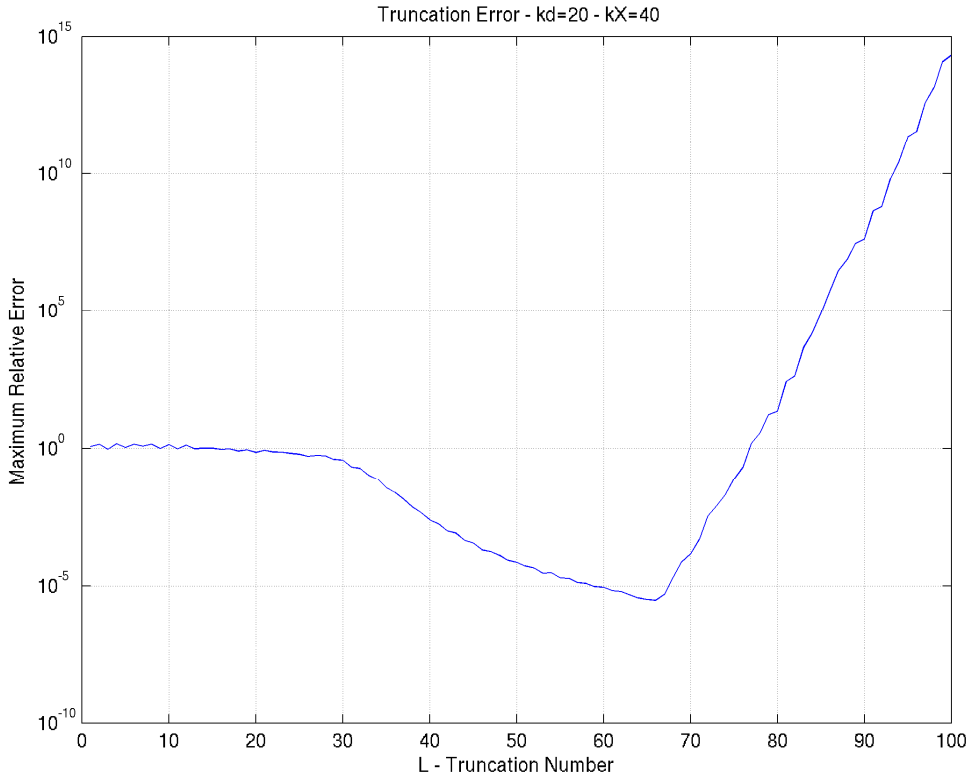

Figure 4-10: Truncation error plot for *kd*=20 and *kX*=40.

ε	L	Time(sec)	ε	L	Time(sec)
10^{-2}	38	0.4389	10^{-7}	N/A	3.7064
10^{-3}	42	0.7038	10^{-8}	N/A	3.7174
10^{-4}	49	1.1568	10^{-9}	N/A	3.7194
10^{-5}	59	1.7627	10^{-10}	N/A	3.7054
10^{-6}	N/A	3.7024	10^{-11}	N/A	3.7064

Table 4-1: Truncation number values corresponding to desired error level and computational time spent in the estimator routine for the configuration *kd*=20, *kX*=40.

In the second test, the parameters involving with the cluster size and spacing are chosen as $kd = 20$ and $kX = 220$. This configuration is nothing more than the large-buffer size configuration discussed in Section 4.2.2. Truncation error for this configuration is demonstrated in Figure 4-11. Retrieved truncation numbers from the routine and execution timings for different error rates are tabulated in Table 4-2. It can be

seen from the table that the routine successfully locates the optimum L – parameter given the desired error level.

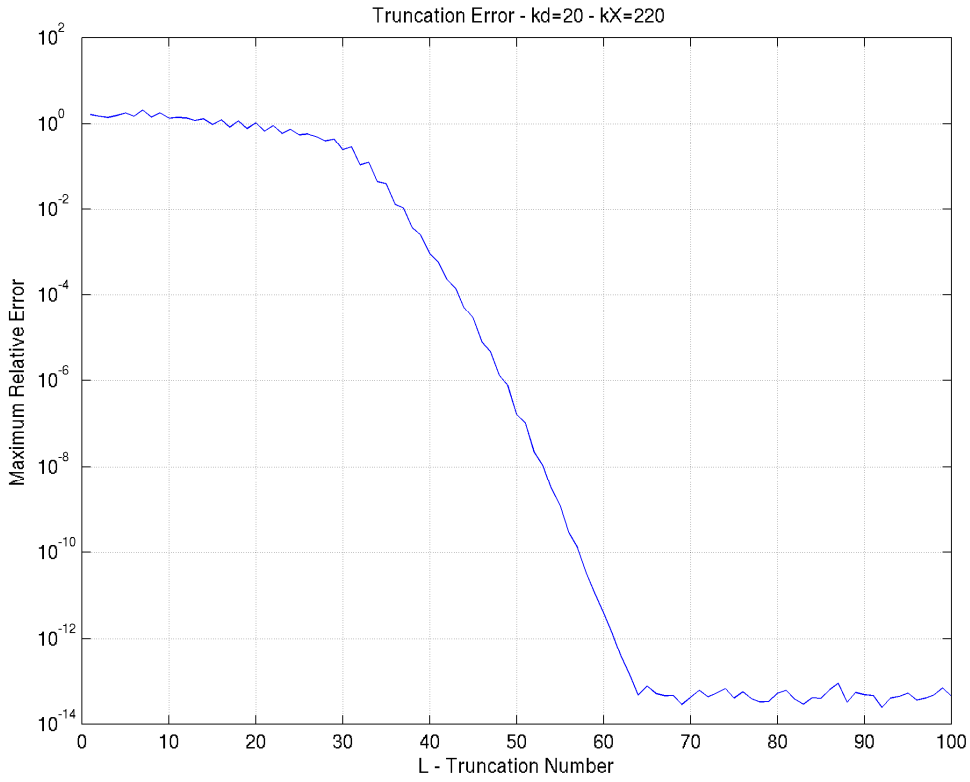

Figure 4-11: Truncation error plot for *kd*=20 and *kX*=220.

ε	L	Time(sec)	ε	L	Time(sec)
10^{-2}	38	0.4399	10^{-7}	52	1.6527
10^{-3}	40	0.7178	10^{-8}	54	1.6987
10^{-4}	44	1.1188	10^{-9}	56	1.6967
10^{-5}	46	1.1268	10^{-10}	58	1.7537
10^{-6}	49	1.1488	10^{-11}	60	1.7727

Table 4-2: Truncation number values corresponding to desired error level and computational time spent in the estimator routine for the configuration *kd*=20, *kX*=220.

As the last example, another one box buffer configuration is tested. Its specifications are $kd = 40$ and $kX = 80$. As seen from the Figure 4-12, the error curve

exhibits a very steep incline around the error level 10^{-7}. Even in this region, the algorithm exactly locates the optimum truncation number for the error rate 10^{-7} and shows the robustness of the proposed method. The results in next chapter also validate the robustness of the three-stage algorithm.

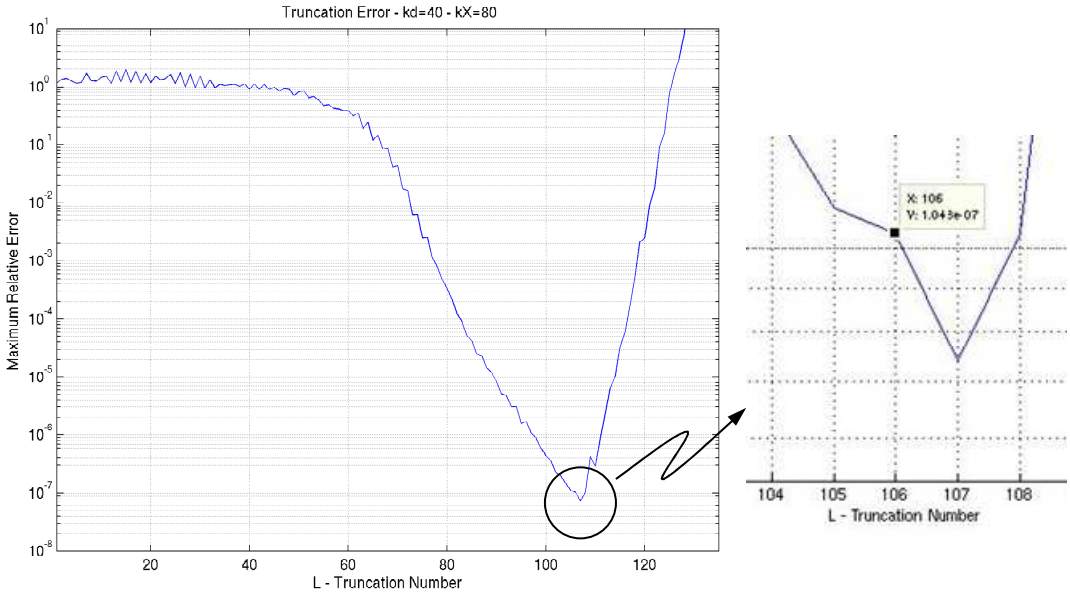

Figure 4-12: Truncation error plot for *kd*=40 and *kX*=80.

ε	L	Time(sec)	ε	L	Time(sec)
10^{-2}	73	2.6236	10^{-7}	107	11.2412
10^{-3}	78	3.4264	10^{-8}	N/A	12.0181
10^{-4}	83	4.7842	10^{-9}	N/A	12.7290
10^{-5}	90	5.0492	10^{-10}	N/A	12.6650
10^{-6}	98	5.3151	10^{-11}	N/A	12.1251

Table 4-3: Truncation number values corresponding to desired error level and computational time spent in the estimator routine for the configuration *kd*=40, *kX*=80.

4.3 The Local Interpolation Parameters Estimator Algorithms

4.3.1 Overview

In the implementation of MLFMA, the local interpolation procedure is frequently invoked while interpolating/anterpolating field signatures and translating the outgoing fields into incoming fields. Since the most of the computational time during MLFMA execution is spent in these operations, the local interpolator and its parameters are to be optimally chosen. The parameters of the local interpolator, the number of interpolation points p and the over-sampling ratio s, affect the CPU time and accuracy of MLFMA.

The literature involving with the optimum parameter selection for the local interpolator is in short supply, especially for the approximate prolate spheroidal (APS) function described in Chapter 2. Due to the error analysis of the APS function given in [63, 64], Ohnuki and Chew proposed formulas for the optimum selection of the interpolation parameters [103]. However, the formulas given there do not provide precise results as in this study. Another study was conducted by Ergul and Gurel [104]. They tested several cluster sizes and proposed optimum (p, s) pairs due to several error rates ε. Nonetheless, the method proposed here is a generic procedure and suitable to any randomly selected cluster size and error rate without conducting any test. The algorithm provide either $p-$parameter for given s, L, and ε or $s-$parameter due to given p, L and ε. In the rest of this subsection, first, the algorithm for estimating $p-$parameter is discussed. Then the algorithm for finding $s-$parameter is presented.

Before proceeding further, the APS formulation is reviewed first. The APS function is given by [63]

$$\tilde{f}(\theta) = \sum_{m=m_0-p+1}^{m_0+p} f(m\Delta\theta) S_N(\theta - m\Delta\theta, \theta_0) D_M(\theta - m\Delta\theta), \tag{4.34}$$

where $D_M(\theta)$ denotes the periodic sinc function or Dirichlet kernel while $S_N(\theta, \theta_0)$ is the windowing function. Windowing function is defined as

$$S_N(\theta, \theta_0) = \frac{R_N(\theta, \theta_0)}{R_N(0, \theta_0)}, \tag{4.35}$$

$$R_N(\theta,\theta_0)=\frac{\sinh\left[(2N+1)\sinh^{-1}\sqrt{\sin^2(\theta_0/2)-\sin^2(\theta/2)}\right]}{\sqrt{\sin^2(\theta_0/2)-\sin^2(\theta/2)}}. \tag{4.36}$$

And the Dirichlet kernel is expressed as

$$D_M(\theta)=\frac{\sin\left[(2M+1)\theta/2\right]}{(2M+1)\sin(\theta/2)}. \tag{4.37}$$

The terms given in the expressions can be more explicitly described as follows: L is the truncation number of addition theorem. $M=sL$ is the total number of sampling points, where s is the over-sampling ratio. The sample spacing is $\Delta\theta=(2\pi)/(2M+1)$. $N=M-L=(s-1)L$ is the number of over-sampling points. $m_0=\mathrm{Int}[\theta/\Delta\theta]$ shows the index of the nearest sampling point to the interpolating point. $\theta_0=p\Delta\theta$ is the width of the window. $p=n/2$ is the number of the interpolation points.

4.3.2 The Number of Interpolation Points (p) Estimator Algorithm

4.3.2.1 Algorithm

In literature, the formula for estimating the optimum $p-$parameter is given as [103]

$$p=\frac{cs}{\pi(s-1)} \tag{4.38}$$

which is the direct consequence of the formula for the upper bound of the error given as [63]

$$10^{-d_0}<\frac{1}{\sinh(c)} \tag{4.39}$$

where

$$c=\pi\left(1-\frac{1}{s}\right)p\ \ (>1). \tag{4.40}$$

It was proven that the Eqn.(4.38) does not guarantee precise results and doesn't take the argument L into account which has also effect on the error rate, albeit not so much (see Section 4.3.2.2). In order to have a rigorous expression for selecting the optimum $p-$parameter, consider the windowing function (Eqn.(4.35)) which is called convergence factor in [64]. Tuning the parameters of the windowing function properly allows finding the upper and lower bounds of the approximation error. The parameters of the windowing function are N, θ_0, and θ. Since the parameter N is related with the polynomial order L of the function whose values are generated and over-sampling ratio s, it can't be changed for the error analysis. The approximation error (or convergence factor) heavily depends on the proper selection of window width θ_0. The width of window can be at least $\Delta\theta$ which means that the local interpolation is performed by only two points in the neighborhood of the interpolating point, or at most π which means that whole points on the circumference are taken into account, i.e. global interpolation $(\Delta\theta < \theta < \pi)$. The parameter θ can be at least 0 and at most θ_0 because of the limits of the summation $(0 < \theta < \theta_0)$. When $\theta = 0$, the convergence factor becomes

$$S_N(0,\theta_0) = \frac{R_N(0,\theta_0)}{R_N(0,\theta_0)} = 1 \tag{4.41}$$

which constitutes the upper bound of the convergence factor (or the top of main lobe). When $\theta = \theta_0$,

$$S_N(\theta_0,\theta_0) = \frac{R_N(\theta_0,\theta_0)}{R_N(0,\theta_0)}, \tag{4.42}$$

and

$$R_N(\theta_0,\theta_0) = \frac{\sinh\left[(2N+1)\sinh^{-1}\sqrt{\sin^2(\theta_0/2)-\sin^2(\theta_0/2)}\right]}{\sqrt{\sin^2(\theta_0/2)-\sin^2(\theta_0/2)}}. \tag{4.43}$$

The Eqn (4.43) is in indeterminate form $(0/0)$. At this point, the L'Hôpital's rule can be invoked. Taking the limit of expression when $\theta \to \theta_0$ yields

$$R_N(\theta_0,\theta_0)=2N+1. \tag{4.44}$$

Hence,

$$S_N(\theta_0,\theta_0)=\frac{2N+1}{R_N(0,\theta_0)} \tag{4.45}$$

that gives the lower bound of the convergence factor (or the side lobe level). By properly tuning Eqn.(4.45), the desired level of accuracy can be obtained in interpolation with APS interpolator.

The algorithm to obtain the optimum $p-$parameter corresponding to given s, L, and ε can be summarized in three steps:

(i) The routine checks the maximum and minimum convergence rates by setting $\theta_0=\Delta\theta$ and $\theta_0=\pi$. If the desired error is out of the bounds of the convergence $\left[S_N(\Delta\theta,\Delta\theta),S_N(\pi,\pi)\right]$, then the routine warns the user and stops execution.

(ii) If the desired error falls into the convergence interval, then the algorithm starts to seek it by bisection algorithm described in Section 4.2.3.3. At the end of the search, it outputs the optimum θ_0 corresponding to given ε.

(iii) By using $p=\theta_0/\Delta\theta$, the optimum $p-$parameter is determined.

4.3.2.2 Numerical Results

The $p-$parameter estimator algorithm has been tested for many numerical cases and five of them which can possibly be encountered at many levels of MLFMA are presented here.

In numerical tests, first, the translation operator values at equispaced sampling points in the interval $[0,2\pi]$ are directly generated. Then, three equally spaced points between two consecutive sampling points are chosen for each two consecutive sampling points a_i and a_{i+1} $(i=1,...,2M+1)$. Since the distance between two successive sampling points is denoted by $\Delta\theta$, the locations of three equispaced points in the interval corresponds to $(a_i+(\Delta\theta/4),a_i+(\Delta\theta/2),a_{i+1}-(\Delta\theta/4))$. The values of translation operator at these interior points are generated by using the interpolator and employing

directly the translation operator routine. The relative differences between interpolated values and directly calculated values (or interpolation errors) are plotted. It's predicted and confirmed that the interpolation error is minimum when the interpolation point is near to sampling point. As the interpolation point is at the midpoint of the interval, the interpolation error is maximum.

For each test, the robustness of the $p-$parameter estimator algorithm is analyzed through setting the error rates to $10^{-1}, 10^{-3}, 10^{-6}$, and 10^{-9}. While the parameters L and s are kept fixed, the optimum $p-$parameter is calculated for each error rate. At the title of each interpolation error plot, the maximum interpolation error encountered at that case is indicated. In addition, the maximum interpolation error is compared with the desired error level.

The first test is conducted on a configuration in which the parameter L is chosen as 40 and that can be encountered in MLFMA at finest level most frequently. The interpolation errors at selected points due to calculated $p-$parameters, $s = 2$, $\mathrm{k}L = 40$ and desired error levels are calculated and demonstrated in Figure 4-13 . Note that only the uppermost portions of the error plots are zoomed in order to see how precise the maximum interpolation errors are obtained. The maximum relative errors at each case except 10^{-1} case are below the desired levels of accuracy. At 10^{-1} case, the maximum relative error is slightly higher than the desired error. Since the procedure described in algorithm section relies on an approximate approach, the error results can deviate one digit of accuracy and this deviation is tolerable.

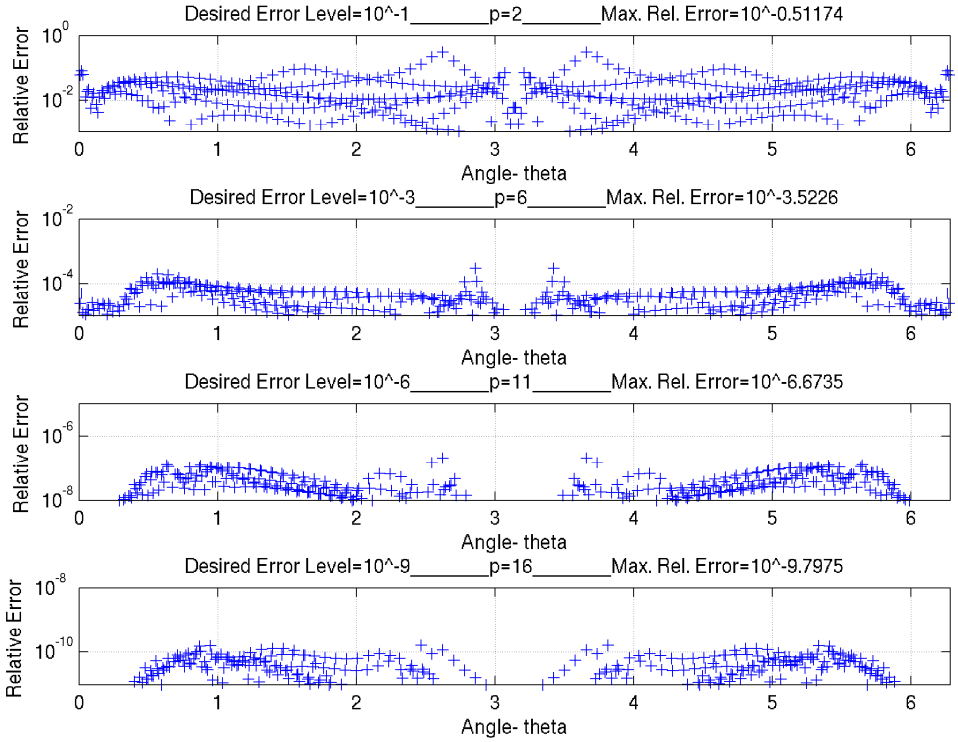

Figure 4-13: Interpolation errors are plotted for different desired error levels with estimated p – parameters, L=40, and s=2.

In the second test, the parameters involving with the interpolation are chosen as $L = 400$, $s = 1.2$. For each error level, the optimum p – parameter is computed and the approximation errors on interpolation points are calculated. The maximum relative errors are also determined. As seen from Figure 4-14, the approximation errors are very near to the desired error rates and lesser than those; this means that the algorithm estimates the true p – parameter very accurately.

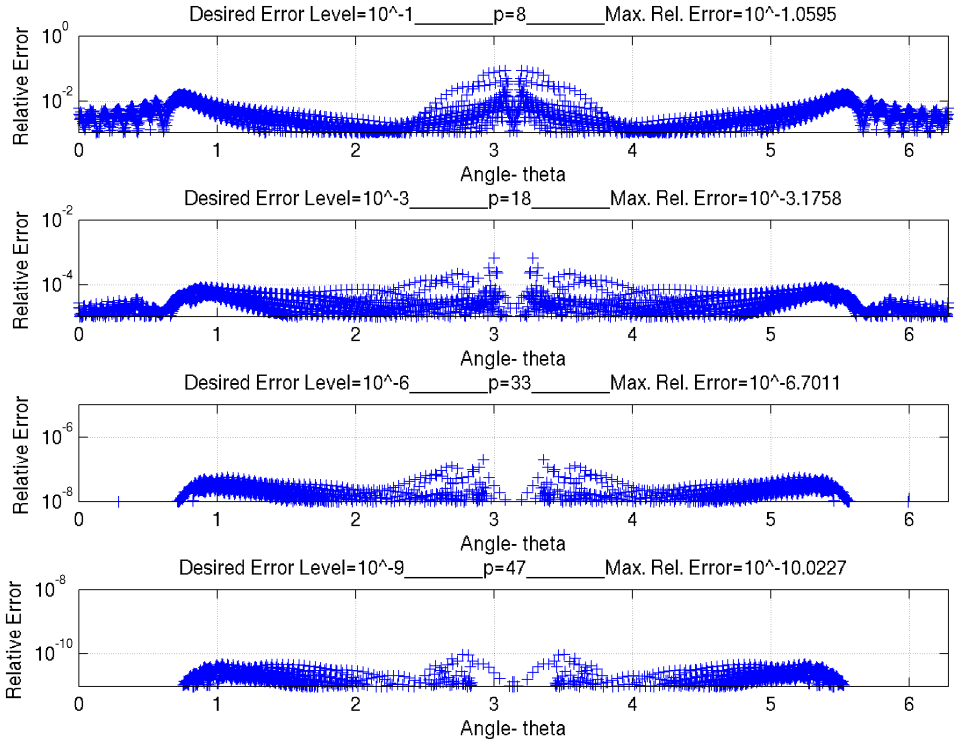

Figure 4-14: Interpolation errors are plotted for different desired error levels with estimated p – parameters, L=400, and s=1.2.

In the third test, the degree of translation operator is increased up to $L = 1000$ .The other parameters are the same as those of previous test. For each error level, the approximation errors on interpolation points are calculated due to fixed parameters s, L and requiring $p-$parameter. Note that the $p-$parameters obtained at previous test are the same as those of previous test, since the interpolation error depends on the parameter s after a certain polynomial degree L. Again, it's apparent from Figure 4-15 that the approximation errors are close to the desired error rates. Also, it's observed in the extensive tests that the interpolation algorithm and the estimator algorithm works until the polynomial degree of 2000 for $s = 1.2$. After this degree, the sinus hyperbolic function in Eqn. (4.36) causes floating overflow. Therefore, the operating limits of the estimator and the interpolator are bounded with the specifications of the machine.

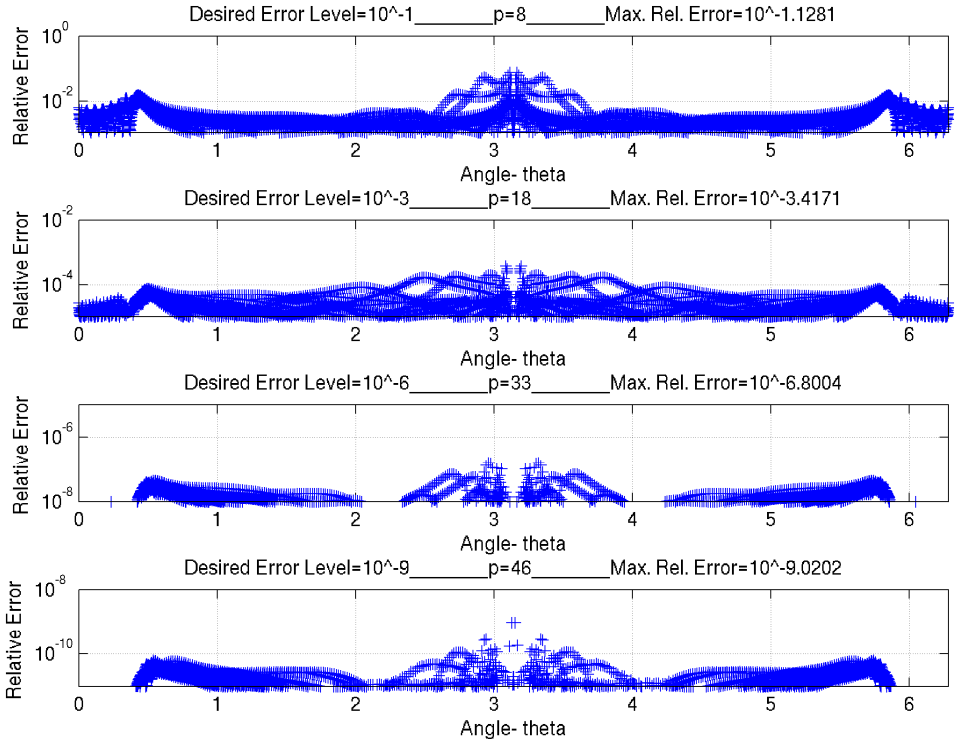

Figure 4-15: Interpolation errors are plotted for different desired error levels with estimated $p-$parameters, L=1000, and s=1.2.

In the fourth and fifth tests, the approximation errors due to estimated $p-$parameters are compared with the approximation errors according to $p-$parameters calculated with the previously used formula given in [103] (Eqn.(4.38)). In the fourth test, the parameters are chosen as $L=100$ and $s=1.2$. At the fifth one, the parameters are selected as $L=100$ and $s=2$. For these specifications, the relative errors due to $p-$parameter estimated with the proposed algorithm are plotted in Figure 4-16 and Figure 4-18. In addition, the approximation errors due to $p-$parameters calculated with Eqn.(4.38) are presented in Figure 4-17 and Figure 4-19. It is apparent from the results that the previous approach yields poor estimations. However, the proposed method results in very precise approximation errors near to the desired error rates.

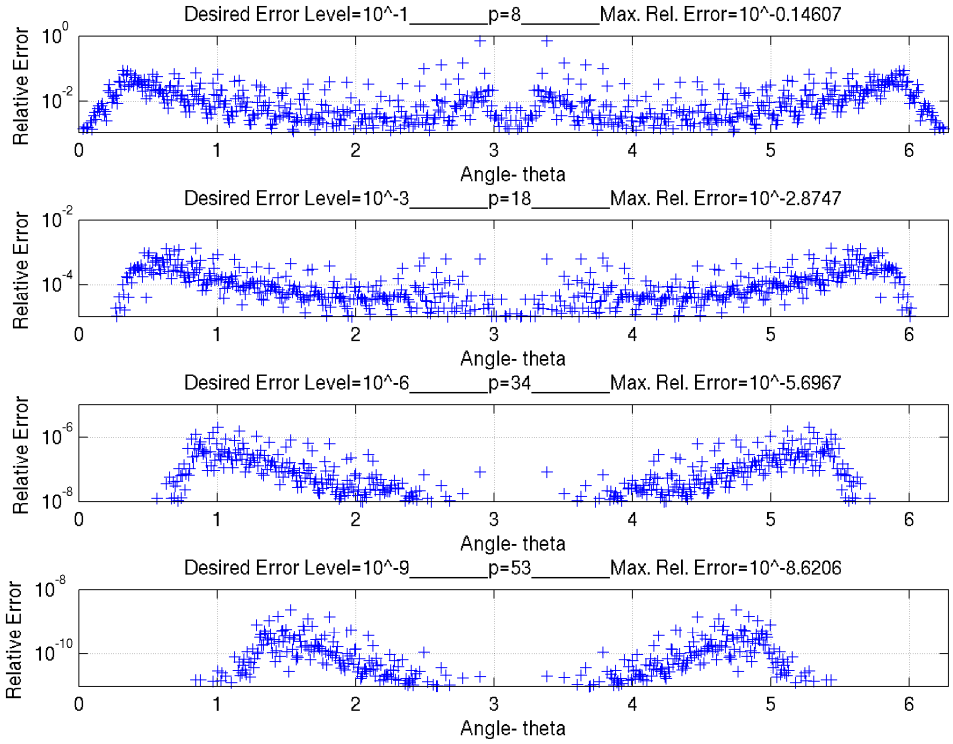

Figure 4-16: Interpolation errors are plotted for different desired error levels with the p – parameters estimated by the proposed algorithm, L=100, and s=1.2.

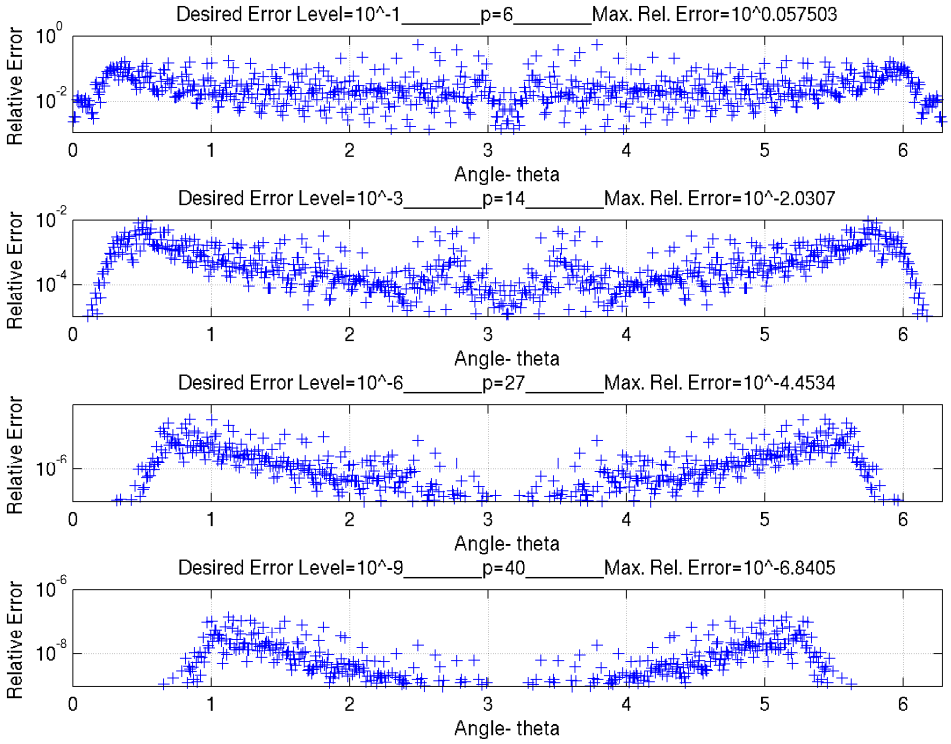

Figure 4-17: Interpolation errors are plotted for different desired error levels with the p – parameters estimated by previously used formula, L=100, and s=1.2.

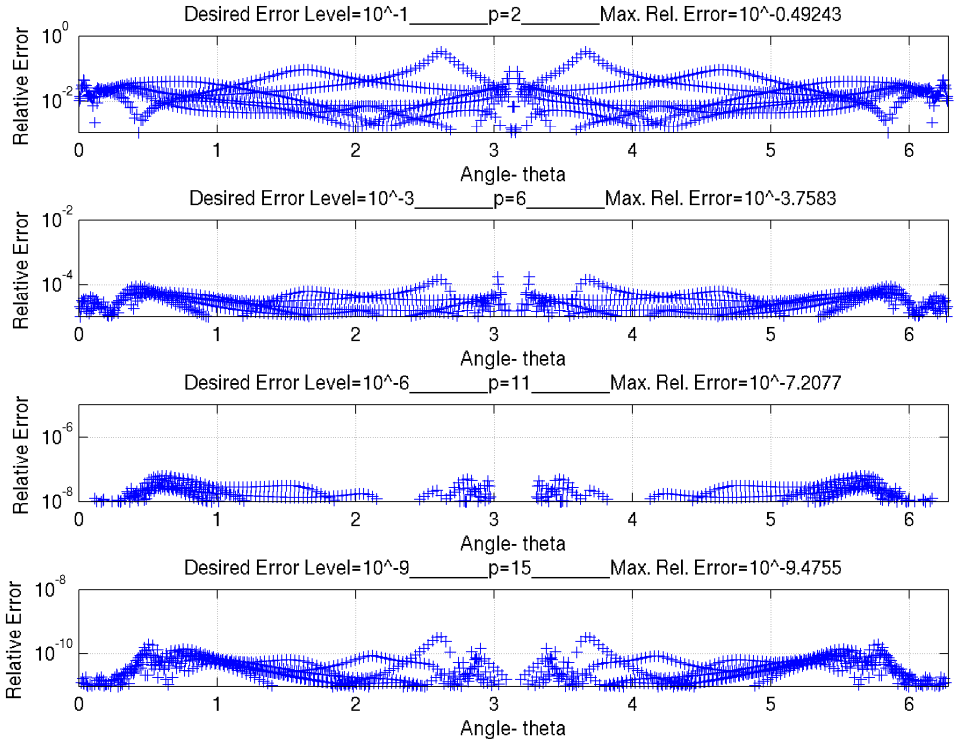

Figure 4-18: Interpolation errors are plotted for different desired error levels with the p – parameters estimated by the proposed algorithm, L=100, and s=2.

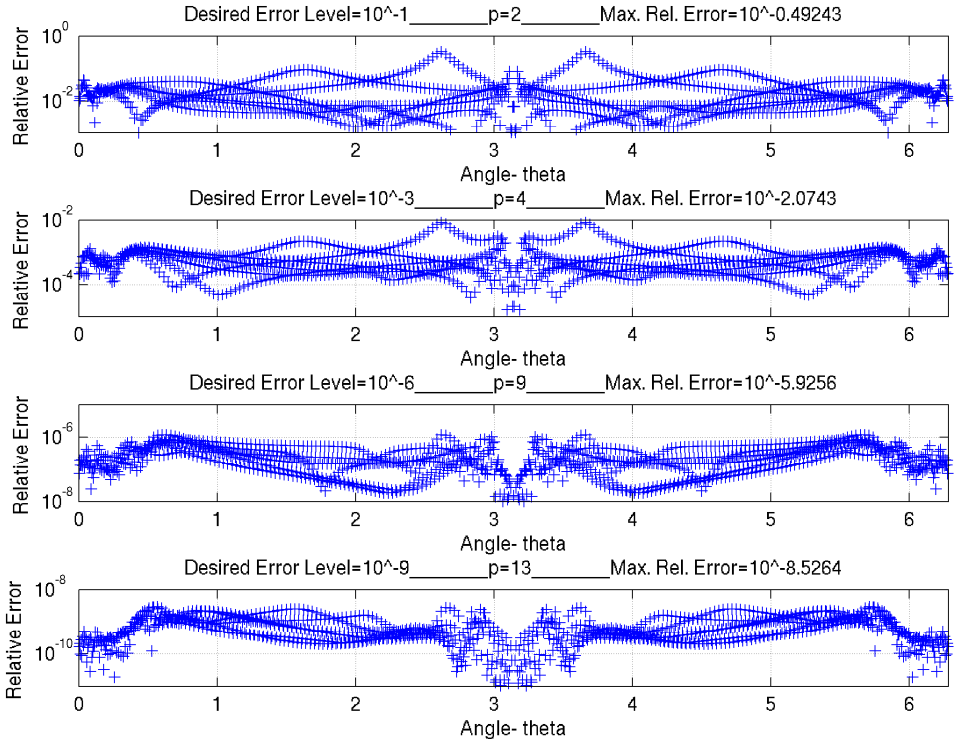

Figure 4-19: Interpolation errors are plotted for different desired error levels with the p – parameters estimated by previously used formula, L=100, and s=2.

4.3.3 The Over-Sampling Ratio (s) Estimator Algorithm

4.3.3.1 Algorithm

In [103], the formula to obtain the optimum $s-$parameter is expressed as

$$s = \frac{\pi p}{\pi p - c}, \tag{4.46}$$

where c is defined as in (4.40). Like $p-$parameter expression in Eqn.(4.38), $s-$parameter formula also doesn't yield accurate error estimation. Therefore, the formulation given for $p-$parameter estimation is arranged to obtain an accurate convergence rate formula for $s-$parameter due to given p and ε. The derivation can be initialized by explicitly writing the θ_0 and N in Eqn.(4.45) as

$$S_N(\theta_0,\theta_0)=\frac{(2(s-1)L+1)\left(\sin(\frac{p\pi}{2sL+1})\right)}{\sinh\left[(2(s-1)L+1)\left(\sinh^{-1}\left(\sin(\frac{p\pi}{2sL+1})\right)\right)\right]}. \quad (4.47)$$

The $s-$parameter can be minimally equal to 1. At this case, the convergence factor becomes

$$S_N(\theta_0,\theta_0)=\frac{\sin\left(\frac{p\pi}{2sL+1}\right)}{\sin\left(\frac{p\pi}{2sL+1}\right)}=1, \quad (4.48)$$

which forms the upper bound of the convergence rate. Similarly, when $s \gg p$,

$$S_N(\theta_0,\theta_0)\approx\frac{p\pi}{\sinh(p\pi)}, \quad (4.49)$$

that is the lower bound of the convergence rate.

The algorithm to obtain the optimum $s-$parameter given p and ε consists of two steps:

(i) The routine checks the lower bound of convergence rate by Eqn (4.49). If the desired error is smaller than this amount, it warns the user and interrupts the execution.

(ii) If the desired error rate is greater than the lower bound of convergence rate, first the algorithm seeks it in the interval [1,10] by bisection algorithm because of the fast decaying nature of error rate. For the remaining part, it employs the Newton-Raphson root-finding algorithm.

4.3.3.2 Numerical Results

The $s-$parameter estimator algorithm has been extensively tested and some of test cases and results are presented here.

The tests of $s-$parameter estimator algorithm are similar to the tests of $p-$parameter estimator algorithm. Basically, the translation operator values are generated by both interpolation routine and translation routine. The relative differences between generated values are calculated and the interpolation errors on specific points are

obtained. The maximum of interpolation errors is compared with the desired error level and the accuracy of the estimator is evaluated through setting the error rates to $10^{-1}, 10^{-3}, 10^{-6}$, and 10^{-9}.In test routines, first the optimum $s-$parameter is calculated due to error rate, L and the parameter p. Then the interpolation routine is executed and resulting approximation errors on interpolation points are plotted.

In the first test, the parameters are selected as $L = 40$ and $p = 10$. It should be noted that the selection of the number of the interpolation points as one forth of the polynomial degree does not yield significant savings in computational time for this test procedure. However, the parameter p should minimally be selected as 8 due to Eqn.(4.49) in order to maintain the minimum error rate 10^{-9}.

The interpolation errors are plotted in Figure 4-20 for above given error rates. Again, the maximum error parts of the error plots are zoomed as in $p-$parameter estimator tests and one digit deviation from the desired error level is acceptable. It's apparent from the graph that the maximum approximation error is well below the threshold for each error case.

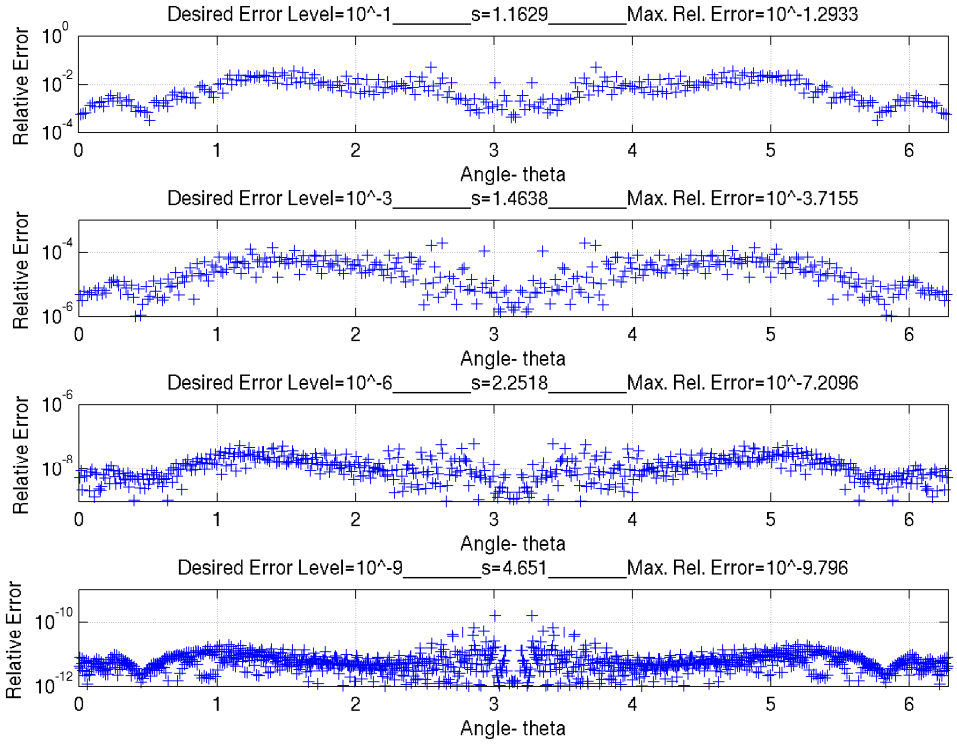

Figure 4-20: Interpolation errors are plotted for different desired error levels with estimated s – parameters, L=40, and p=10.

In the second and third tests, the polynomial degrees are increased up to 400 and 1000 respectively. The $p-$parameter is kept fixed at both tests as 15. It's claimed that optimum $s-$parameters for error cases will be smaller than those of previous test since the $p-$parameter is increased to 15. And the $s-$parameters will not change in both cases due to independency of the error estimate from the polynomial degree after a certain polynomial degree. These expectations are met with the reality in Figure 4-21 and Figure 4-22. Like in previous test, the approximation errors are close to the desired error rates and lesser than those; this means that the algorithm estimates the true $s-$parameter very accurately.

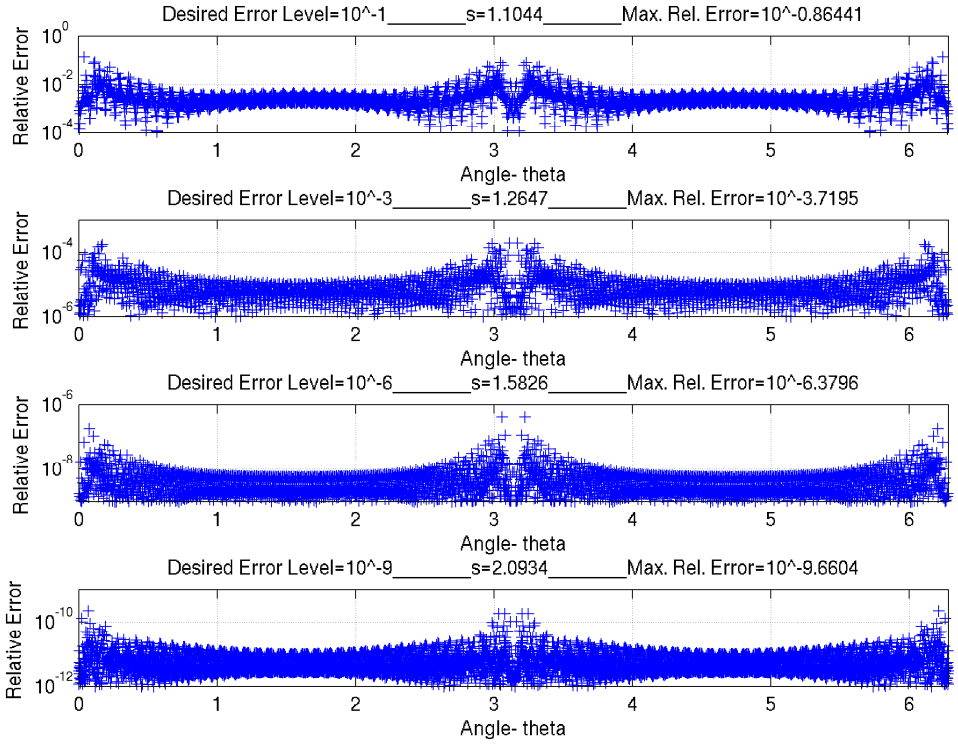

Figure 4-21: Interpolation errors are plotted for different desired error levels with estimated s – parameters, L=400, and p=15.

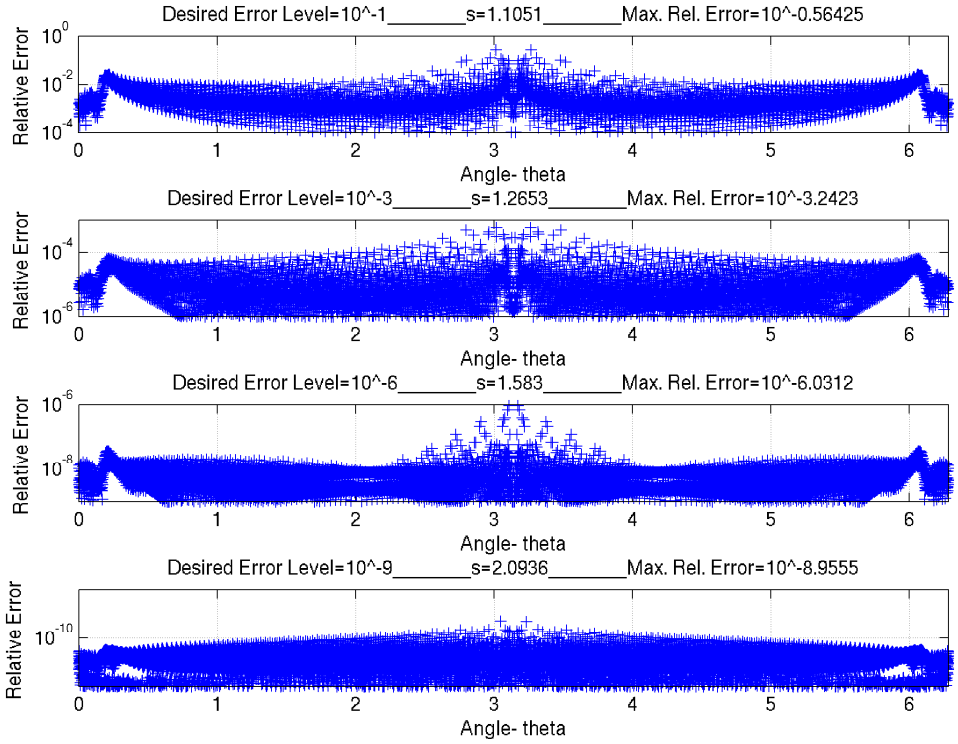

Figure 4-22: Interpolation errors are plotted for different desired error levels with estimated s – parameters, L=1000, and p=15.

In the last two tests, the interpolation errors according to estimated $s-$parameters are compared with the interpolation errors due to $s-$parameters calculated with the formula provided in [103] (Eqn.(4.46)). In the fourth test, the parameters are chosen as $L=100$ and $p=8$. In the fifth test, the parameters are selected as $L=100$ and $p=25$. For these parameters, the interpolation errors due to $s-$parameter estimated with the proposed algorithm are demonstrated in Figure 4-23 and Figure 4-25. In addition, the interpolation errors due to $s-$parameters calculated with Eqn.(4.46) are presented in Figure 4-24 and Figure 4-26. The last case in Figure 4-23 is an exceptional case and shows how the proposed estimator accurately finds the optimum $s-$parameter. The overall results show that the previous approach yields poor results. However, the proposed method estimates the true $s-$parameter very precisely.

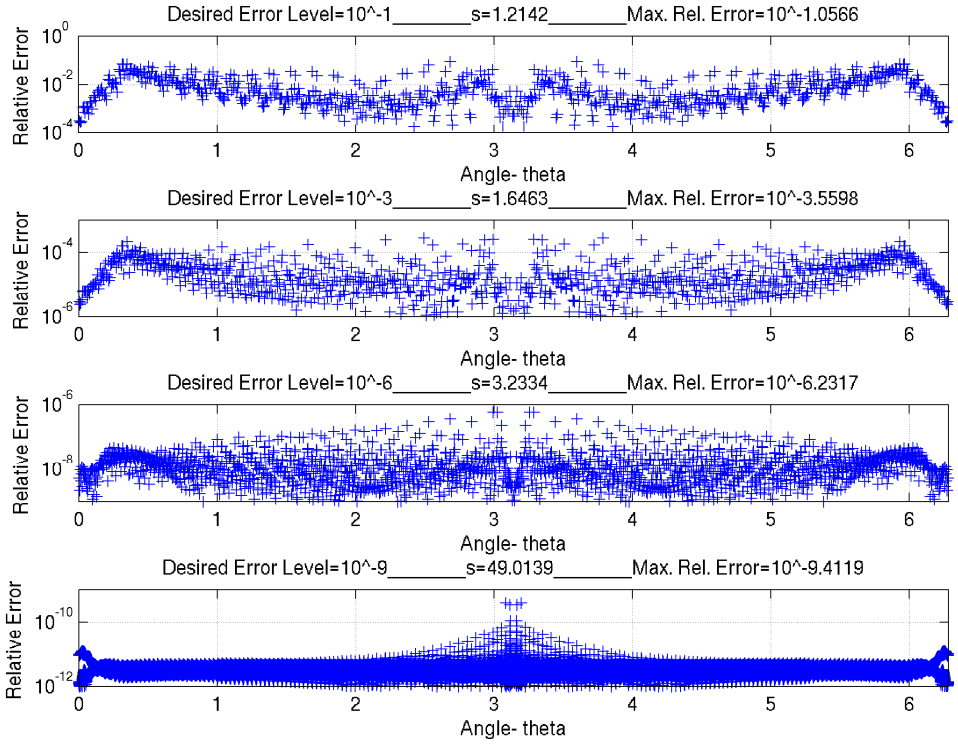

Figure 4-23: Interpolation errors are plotted for different desired error levels with the *s* – parameters estimated by the proposed algorithm, L=100, and p=8.

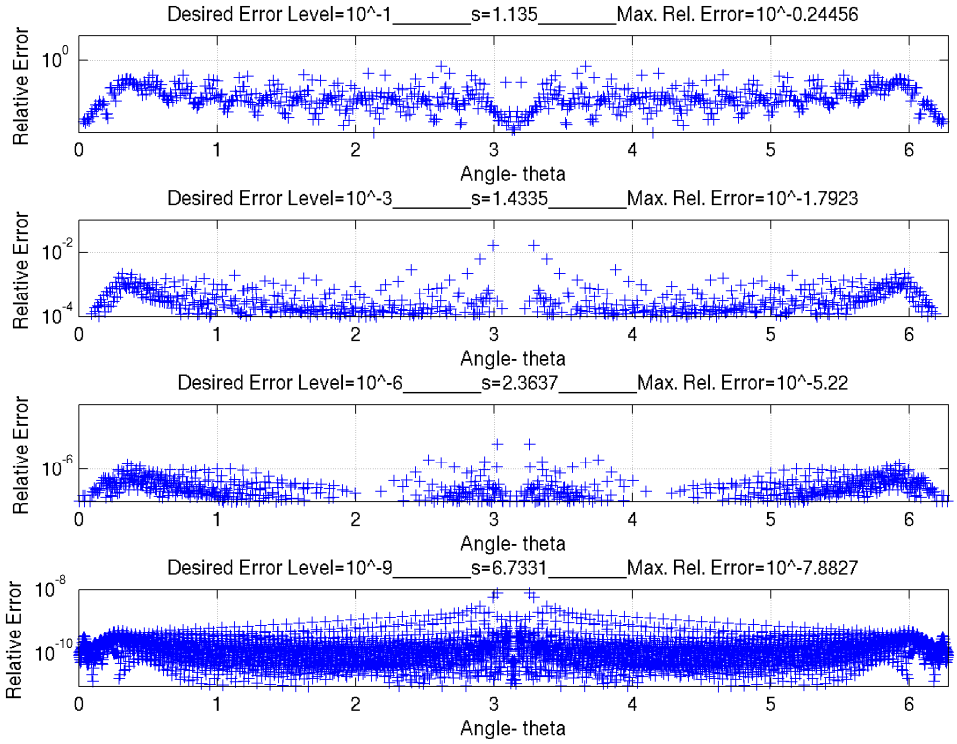

Figure 4-24: Interpolation errors are plotted for different desired error levels with the *s* – parameters estimated by previously used formula, *L*=100, and *p*=8.

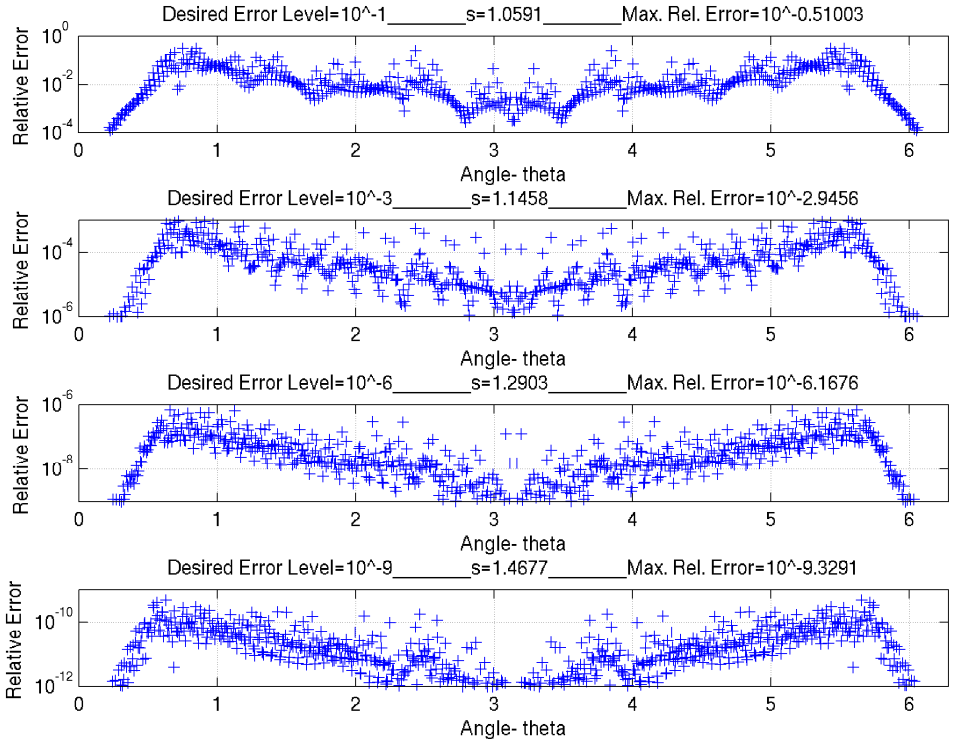

Figure 4-25: Interpolation errors are plotted for different desired error levels with the *s* – parameters estimated by the proposed algorithm, L=100, and p=25.

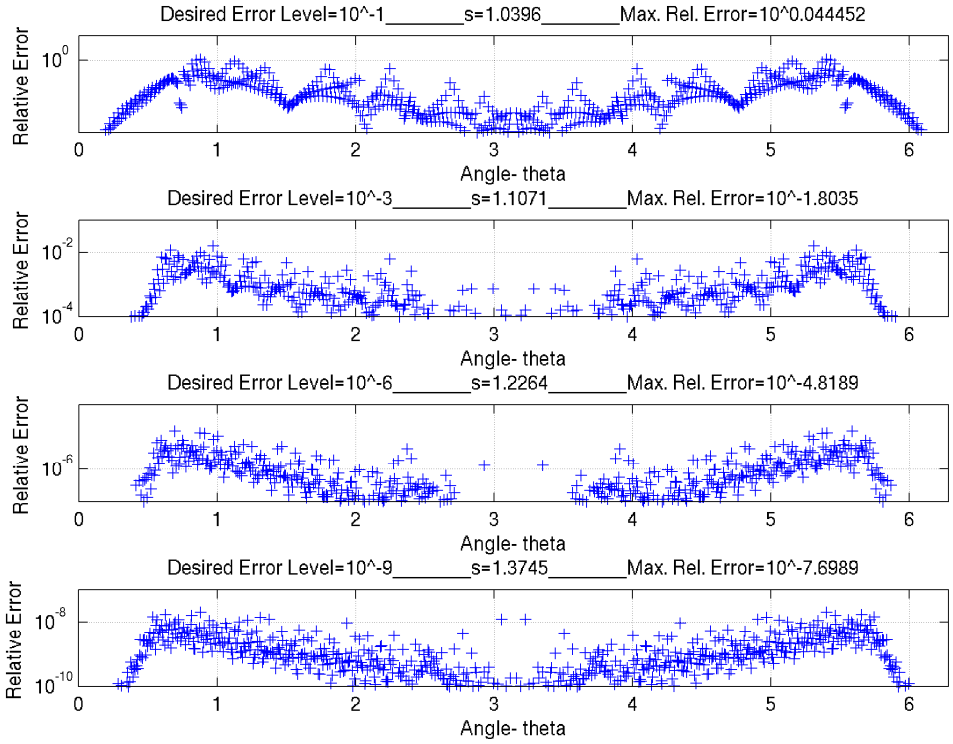

Figure 4-26: Interpolation errors are plotted for different desired error levels with the *s* – parameters estimated by previously used formula, *L*=100, and *p*=25.

CHAPTER 5
NUMERICAL RESULTS

5.1 Introduction

In this chapter, results of extensive tests conducted by Helmholtz MLFMA and Maxwell MLFMA with self-tuning library are presented. In these tests, our first goal is to see how accurate MLFMA results are obtained by using self-tuning library. The second purpose of these tests is to check the limits of the algorithms, i.e. to see how large the cluster size can be used during the executions of the algorithms.

The test setups are constructed for one-level FMM and two-level FMM. At each test setup, the source/observer points for Helmholtz problem or source/observer dipoles for Maxwell problem are selected over the surfaces of clusters. Totally, one hundred twenty points or dipoles are distributed on equally spaced grid over each cluster. The cluster sizes or spacings are changed to examine the limits of algorithms during tests. At each test, four different error rates are examined. These error rates are 10^{-3}, 10^{-5}, 10^{-7}, and 10^{-9}. In addition, relative differences between field values at observer points/dipoles calculated by exact Green's function expression and FMM approximation are plotted.

5.2 One-Level FMM

One-level FMM test configuration is given in Figure 5-1. With this test configuration, Helmholtz MLFMA and Maxwell MLFMA are tested and the results are presented in next subsections. In order to examine small-buffer case in this configuration,

the spacing between cluster centers is set to 16λ at one of tests. All other tests are conducted with the spacing of 1000λ. In all tests below, MLFMA results are obtained with desired levels of accuracy.

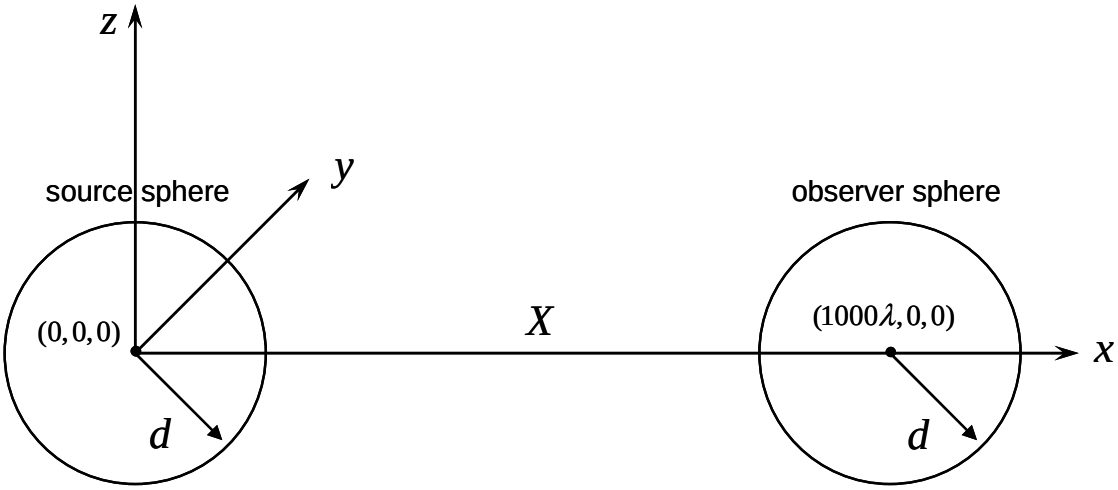

Figure 5-1: One-level FMM test configuration.

5.2.1 Results for Helmholtz MLFMA

In the first test, the cluster radii are selected as 4λ and the spacing between clusters is set to 1000λ. At this configuration, optimum L and p parameters are estimated with self-tuning library. Over-sampling ratio is selected as 1.5. Results are plotted for different desired error rates (See Figure 5-2). It can be seen from the results that the desired error rates are achieved precisely through the self-tuning library.

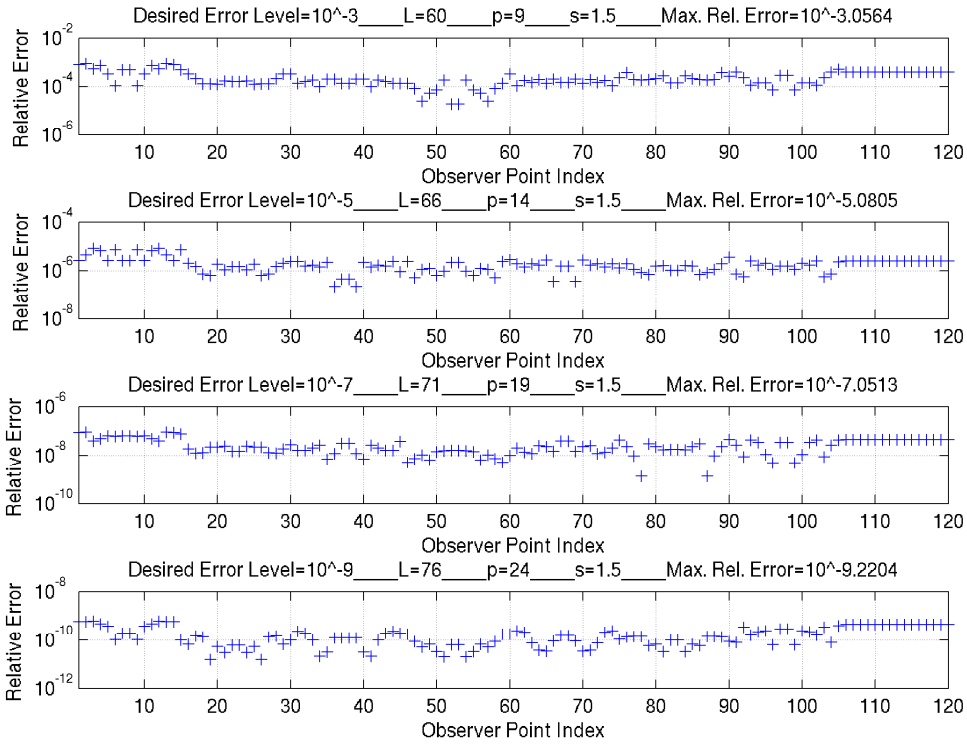

Figure 5-2: Relative errors of field values at observer points at one-level FMM for the configuration $d = 4\lambda$ and $X = 1000\lambda$.

In second test, the cluster radii are kept as in previous test. The observer cluster is placed at $(16\lambda, 0, 0)$ (one-sphere buffer). At this configuration, optimum L and p parameters when $s = 1.5$ are estimated with self-tuning library. Results are plotted for different desired error rates (See Figure 5-3). It can be seen from the results that desired MLFMA error rates are achieved.

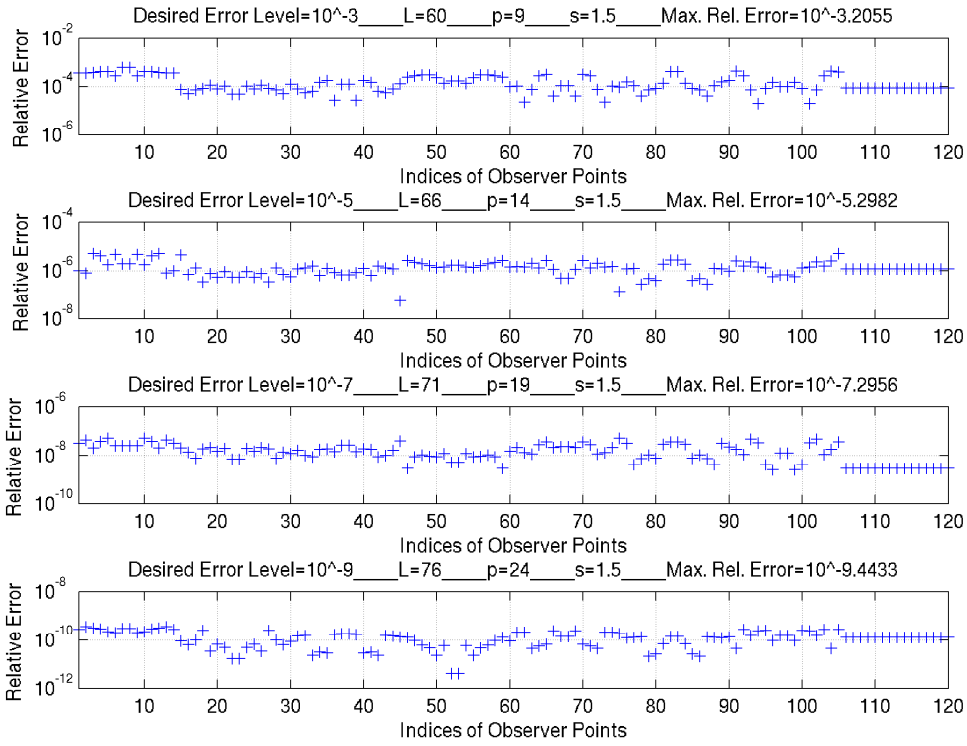

Figure 5-3: Relative errors of field values at observer points at one-level FMM for the configuration $d = 4\lambda$ and $X = 16\lambda$.

In the following two tests, the spacing between two clusters is fixed to 1000λ and various cluster sizes are examined. In the third and fourth tests, cluster radii are selected as 40λ and 150λ respectively. The fourth test also shows the computational limits of the algorithm in our workstation that has 16 GB RAM at one CPU. The results at both tests are satisfactory.

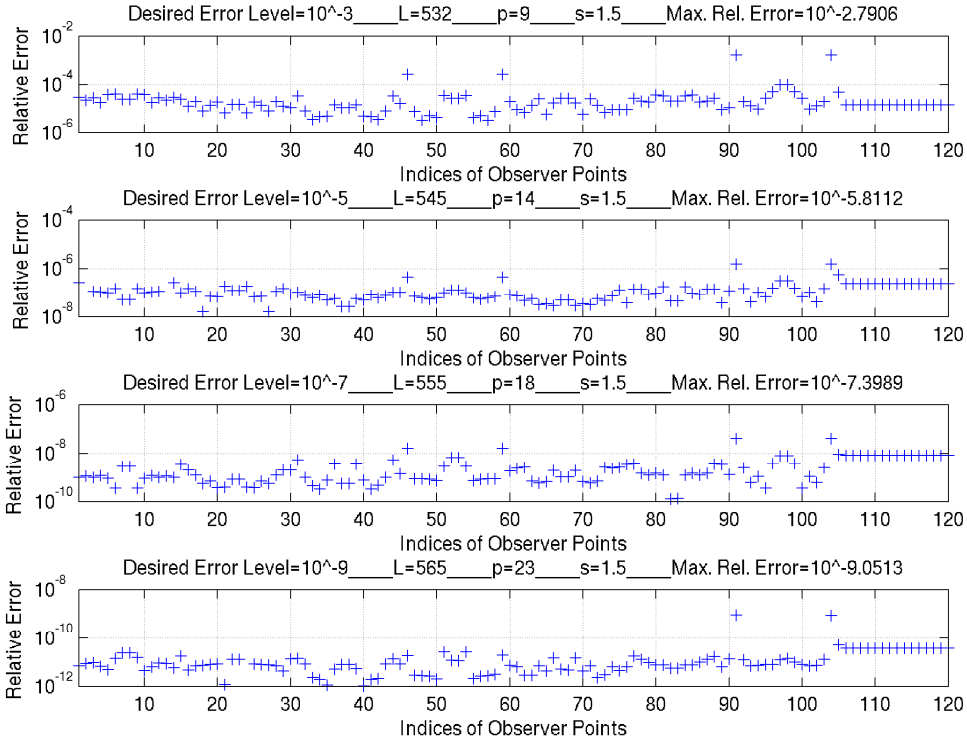

Figure 5-4: Relative errors of field values at observer points at one-level FMM for the configuration $d = 40\lambda$ and $X = 1000\lambda$.

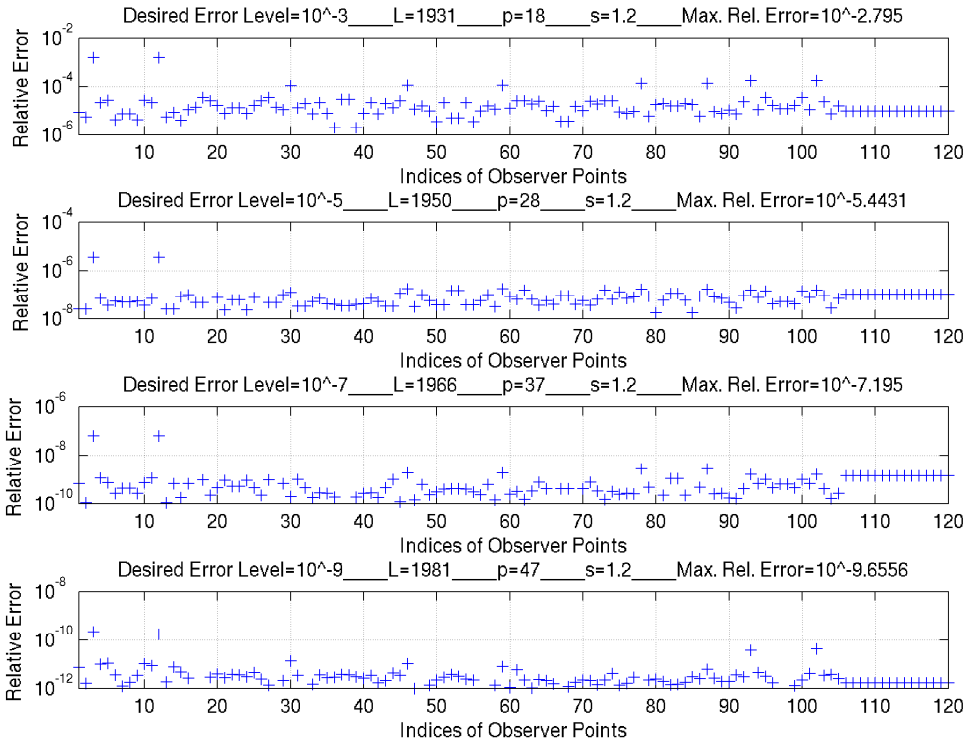

Figure 5-5: Relative errors of field values at observer points at one-level FMM for the configuration $d = 150\lambda$ and $X = 1000\lambda$.

Finally yet importantly, the first test is repeated with the parameters estimated with the formulas presented in literature. It is validated that the proposed method in this thesis yields very accurate results (in Figure 5-2) compared to results in Figure 5-6.

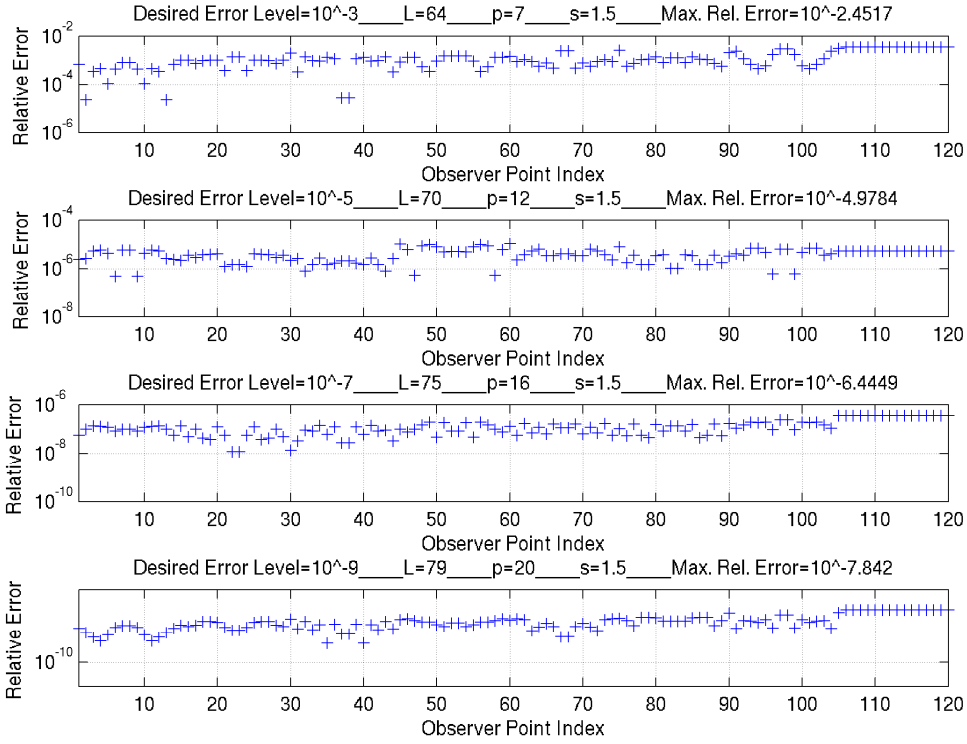

Figure 5-6: Relative errors of field values at observer points at one-level FMM for the configuration $d = 4\lambda$ and $X = 1000\lambda$ (with formulas in literature).

5.2.2 Results for Maxwell MLFMA

The tests for the Helmholtz MLFMA are repeated here for Maxwell MLFMA. In the first test, the cluster radii are assigned as 4λ and $X = 1000\lambda$. At this configuration, optimum L and p parameters are estimated with self-tuning library. It should be stated that the $L-$parameter estimating algorithm is modified for employing source/observer dipoles over the surfaces of clusters instead of source/observer points. Over-sampling ratio is again set to 1.5. Results are plotted in Figure 5-7 for different desired error rates. It can be seen from the results that the desired error rates are achieved precisely through the self-tuning library.

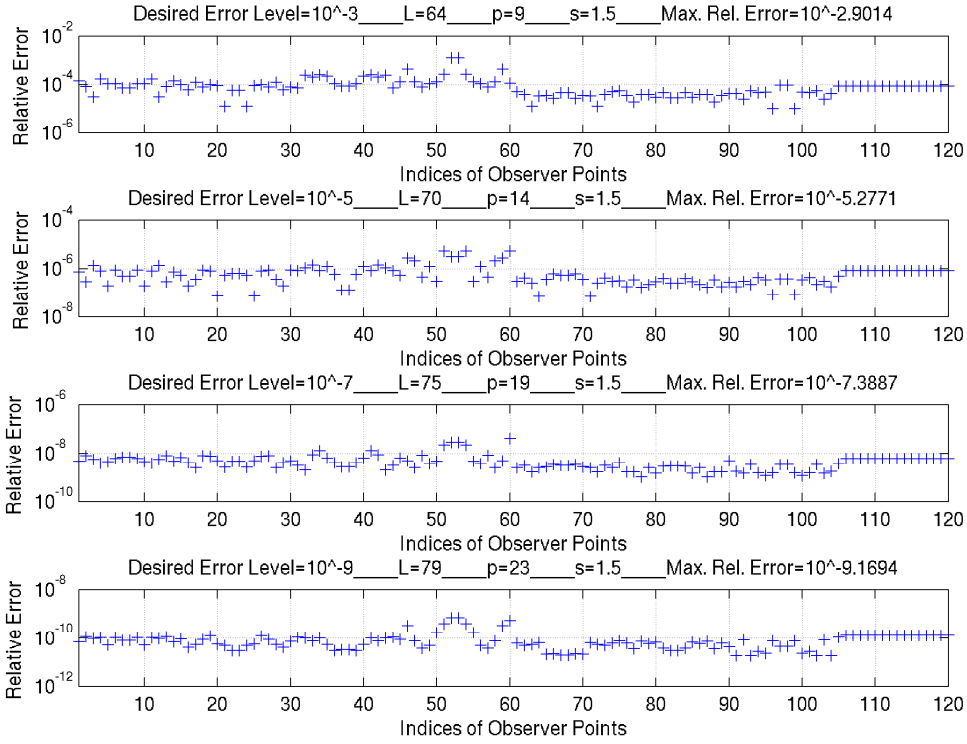

Figure 5-7: Relative errors of field values at observer dipoles at one-level FMM for the configuration $d = 4\lambda$ and $X = 1000\lambda$.

In the second test, the cluster radii are kept as in previous test. The observer cluster is located at $(16\lambda, 0, 0)$ (one-sphere buffer). At this configuration, optimum L and p parameters are estimated with self-tuning library when $s = 1.5$. Results are shown in Figure 5-7. It can be seen from the results that MLFMA error rates are as precise as desired.

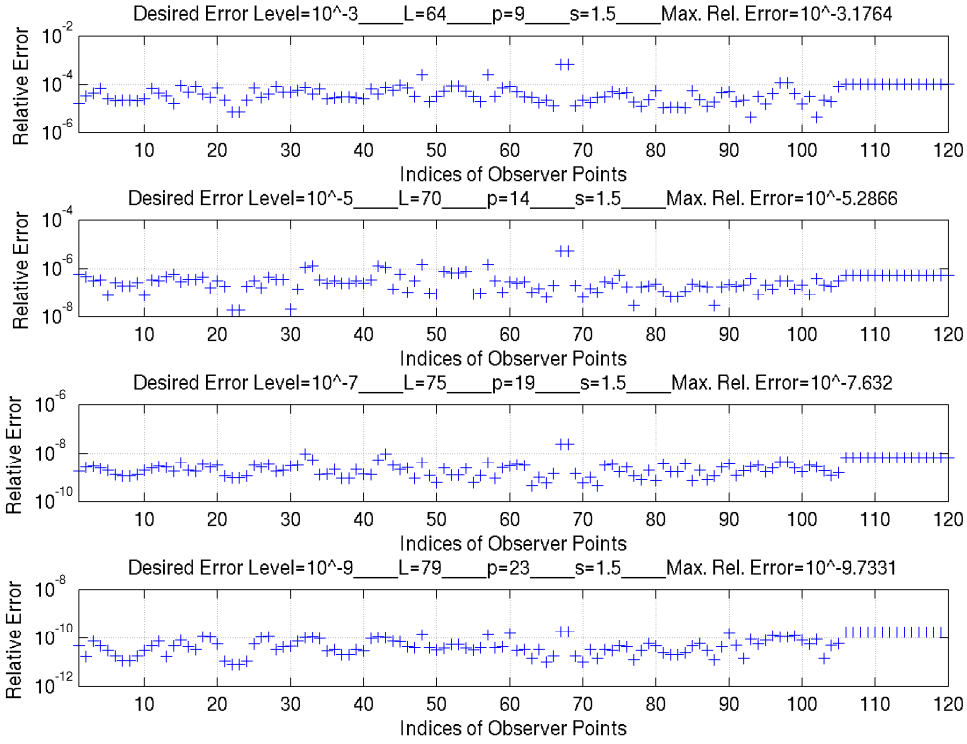

Figure 5-8: Relative errors of field values at observer dipoles at one-level FMM for the configuration $d = 4\lambda$ and $X = 16\lambda$.

In the last two tests, the spacing between cluster centers is set to 1000λ. In the third test, radii of clusters are selected as 40λ. In the fourth test, the radii of clusters are assigned as 65λ which is the upper limit for the computation in our workstation for Maxwell MLFMA implementation. In both tests, the results are as precise as expected before.

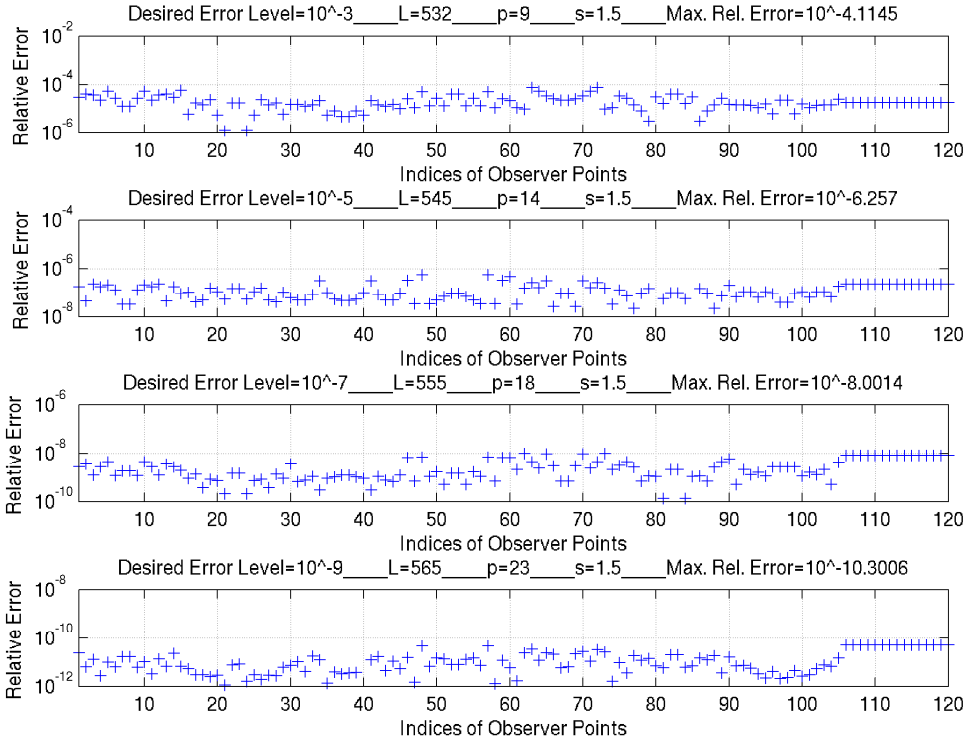

Figure 5-9: Relative errors of field values at observer dipoles at one-level FMM for the configuration $d = 40\lambda$ and $X = 1000\lambda$.

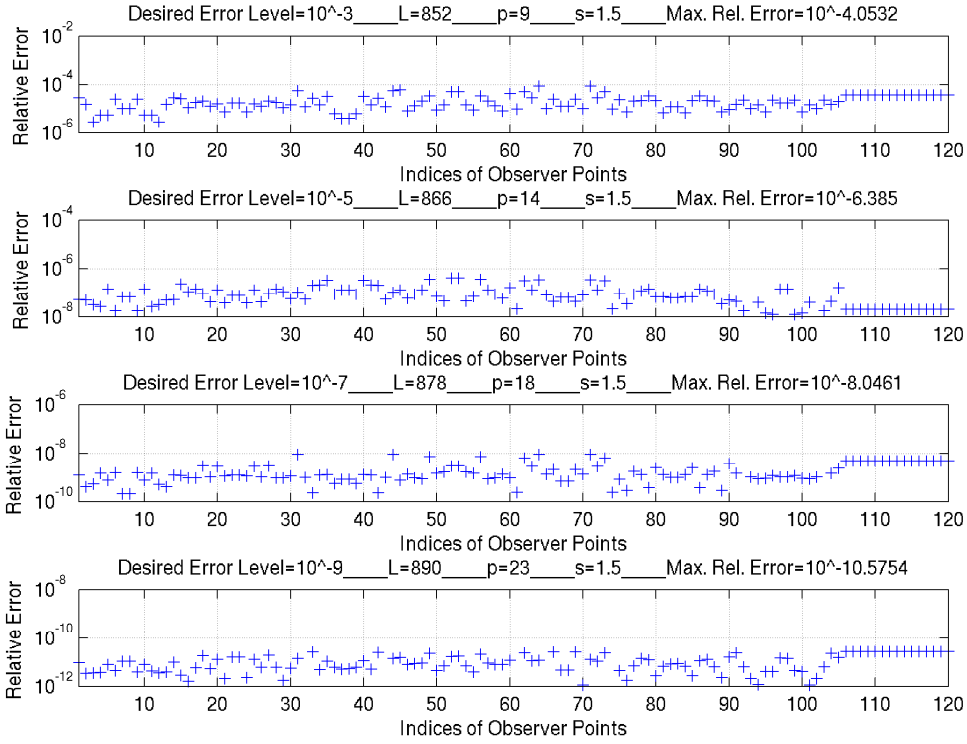

Figure 5-10: Relative errors of field values at observer dipoles at one-level FMM for the configuration $d = 65\lambda$ and $X = 1000\lambda$.

Like in Helmholtz tests, the first test is repeated with the parameters obtained with the formulas in literature. In addition, the accurateness of the method proposed in this study is confirmed again.

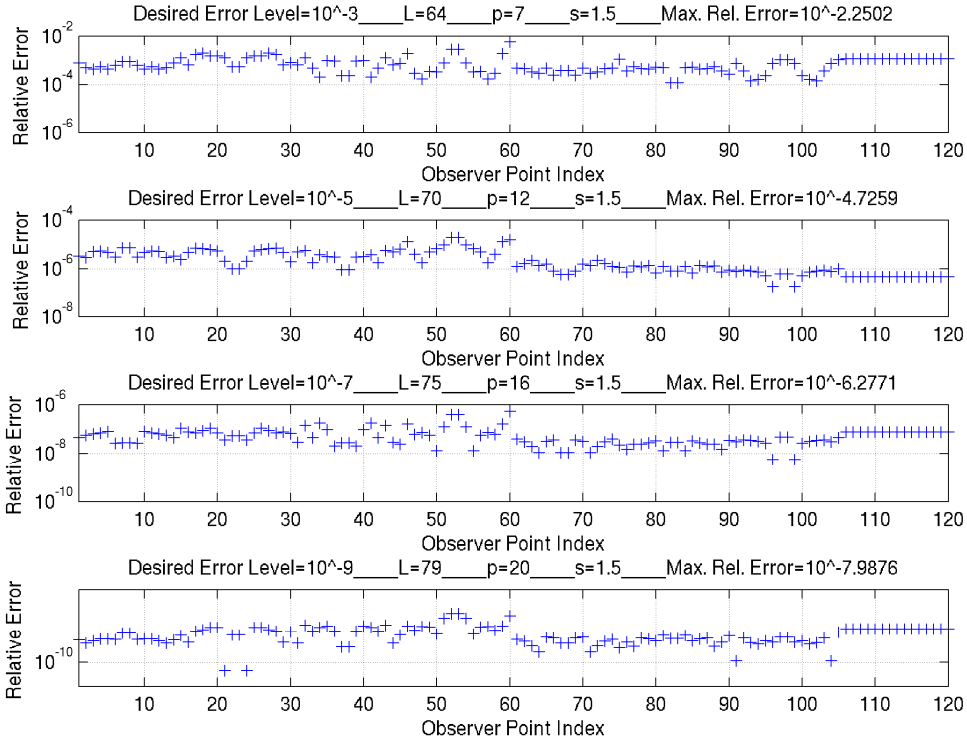

Figure 5-11: Relative errors of field values at observer dipoles at one-level FMM for the configuration $d = 4\lambda$ and $X = 1000\lambda$ (with formulas in literature).

5.3 Two-Level FMM

Two-level FMM test configuration is depicted in Figure 5-12. With this test configuration, Helmholtz MLFMA and Maxwell MLFMA are examined and the results are presented in next subsections. In this configuration, all source spheres enclosed by the sphere located at the origin interact with the observer spheres enclosed by the sphere located at $(1000\lambda, 0, 0)$. However, the fields at only one sphere that resides at the upper-left portion of the big observer cluster are compared. In order to examine small-buffer case in this configuration, the spacing between big cluster centers is set to 38.62λ at one of tests. All other tests are conducted with the spacing of 1000λ. In all tests below, MLFMA results are obtained with desired levels of accuracy.

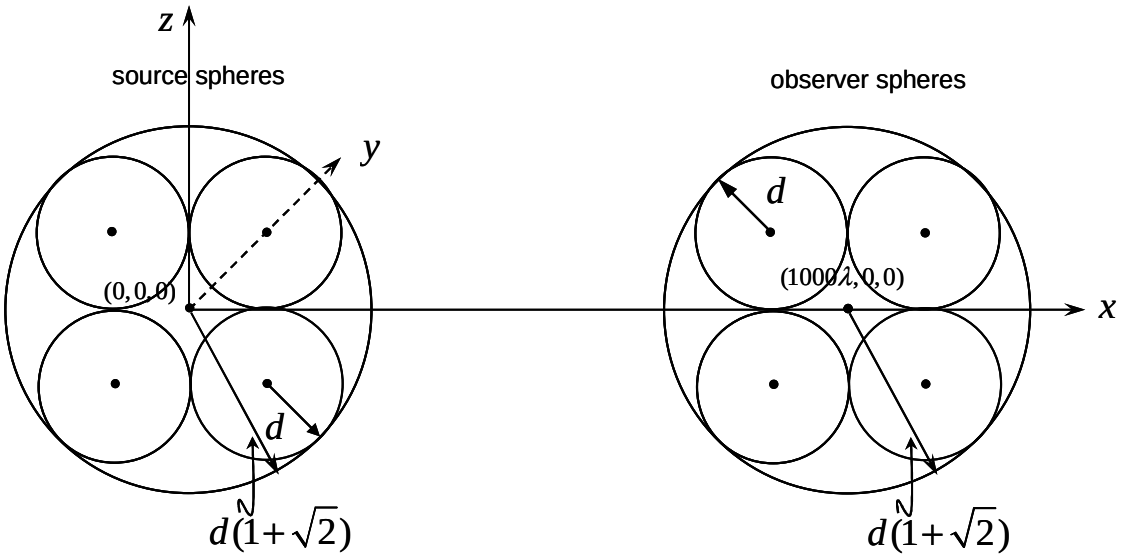

Figure 5-12: Two-level FMM test configuration.

5.3.1 Results for Helmholtz MLFMA

In the first test, the small cluster radii d are selected as 4λ and the spacing between clusters is 1000λ. In this configuration, optimum L and p parameters are estimated with self-tuning library. Over-sampling ratio is set to 1.5. Approximation error at the field points located over the surface of upper-left sphere in the big observer cluster is plotted for different desired error rates (See Figure 5-13). Desired results are obtained through the self-tuning library and those are below the desired error rates. It should be also noted that the approximation errors are lesser than those of the one-level FMM. This is because the estimation calculations are performed at the worst case in which it is assumed that the source/observer points sit over the surfaces. However, source/observer points reside within the big clusters in two-level FMM. Therefore, the results are expected well below the desired level.

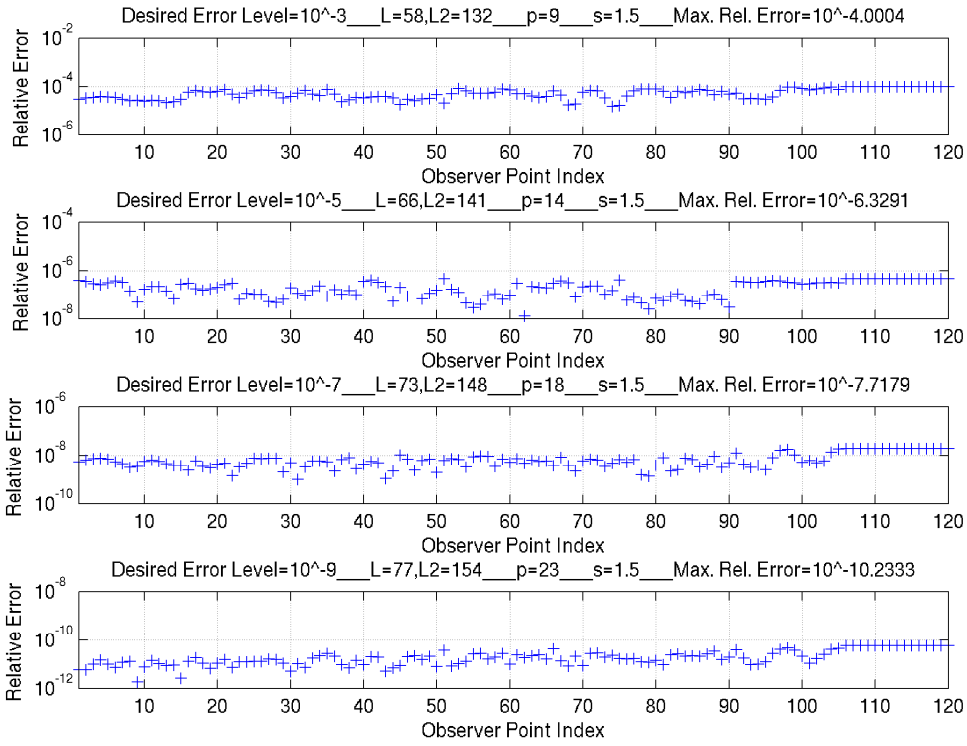

Figure 5-13: Relative errors of field values at observer points at two-level FMM for the configuration $d = 4\lambda$ and $X = 1000\lambda$.

In the second test of this part, the spacing between cluster centers is set to 38.62λ (one big cluster buffer case). In addition, the cluster radii are kept as same as in previous test. The approximation errors on observer points are demonstrated for different error rates (See Figure 5-14). The results are satisfactory

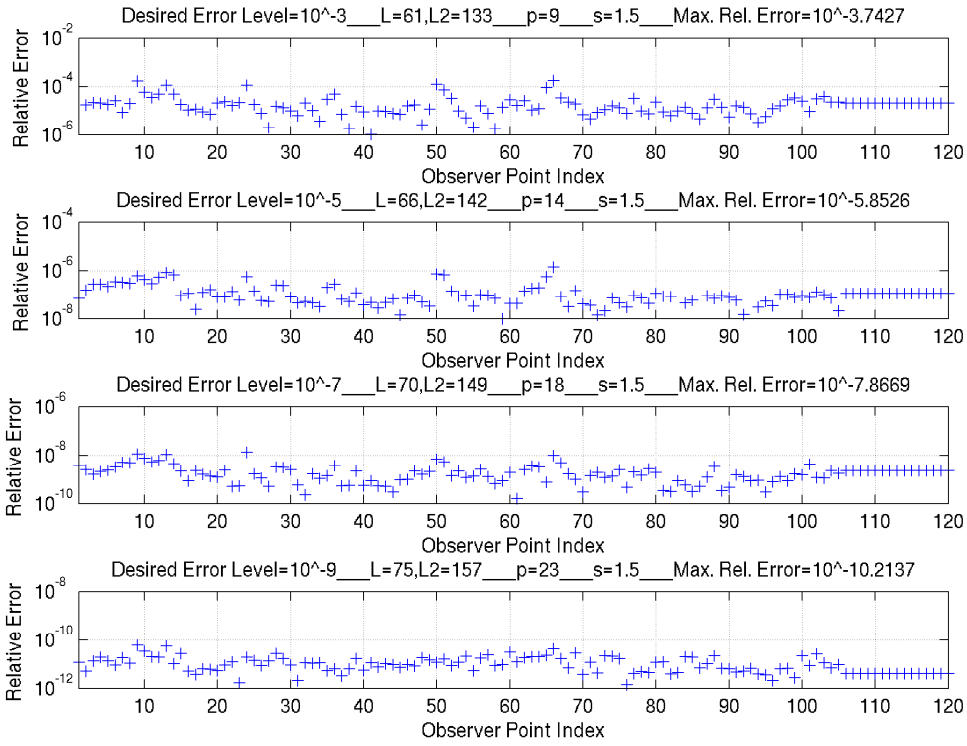

Figure 5-14: Relative errors of field values at observer points at two-level FMM for the configuration $d = 4\lambda$ and $X = 38.62\lambda$.

In the last two tests for Helmholtz MLFMA, the cluster spacing is set to 1000λ. The radii of small clusters are assigned as 40λ for the third test and 100λ for the fourth test. At the configurations, approximation errors for different desired error rates are calculated and plotted in Figure 5-15 and Figure 5-16. In the last test, the translation operation is performed at very high degrees such as 3147. This shows that the routines in MLFMA implementations generate precise results at very high degrees.

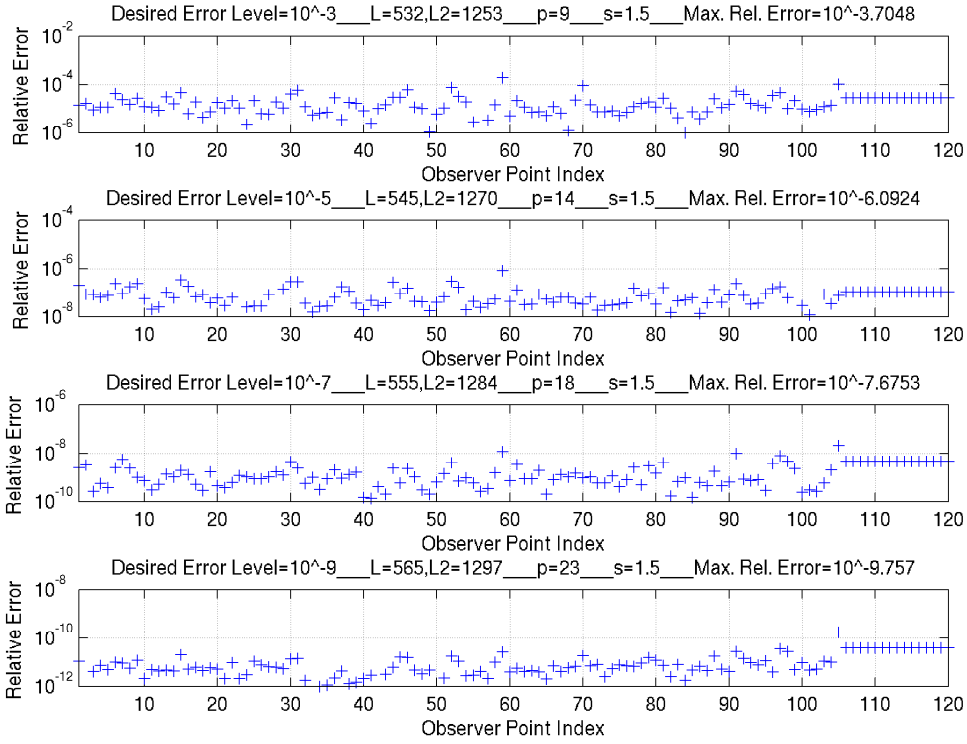

Figure 5-15: Relative errors of field values at observer points at two-level FMM for the configuration $d = 40\lambda$ and $X = 1000\lambda$.

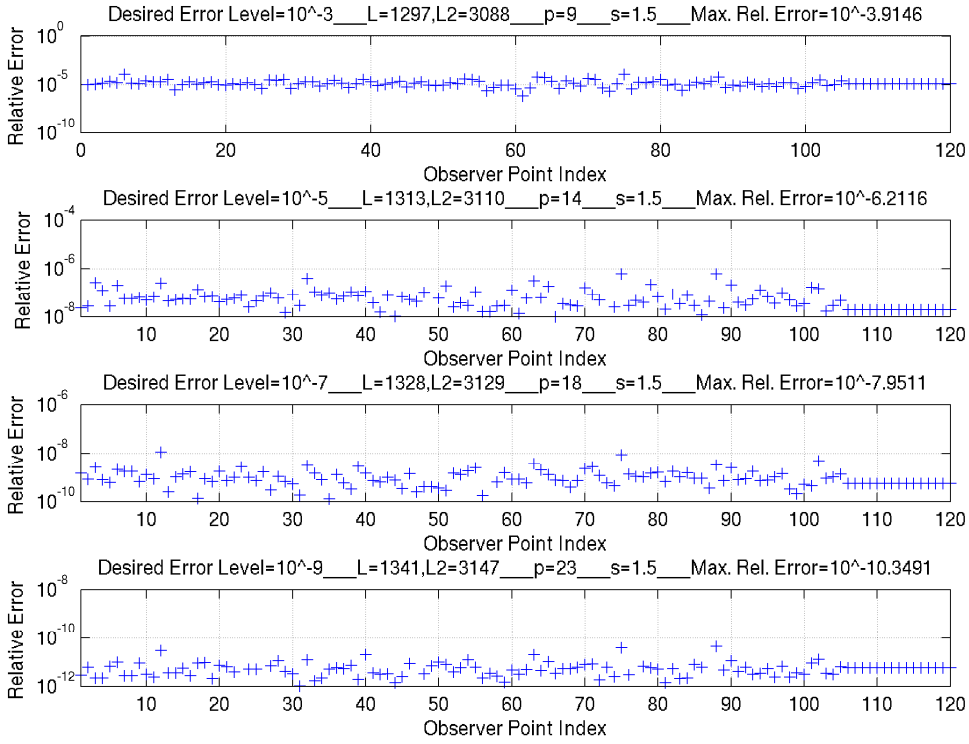

Figure 5-16: Relative errors of field values at observer points at two-level FMM for the configuration $d = 100\lambda$ and $X = 1000\lambda$.

5.3.2 Results for Maxwell MLFMA

The tests for the Helmholtz MLFMA for two-level are repeated here for Maxwell MLFMA. In the first test, the cluster small radii are assigned as 4λ and $X = 1000\lambda$. In the second test, the spacing between big cluster centers is set to 38.62λ and small cluster radius is 4λ. In the third and fourth tests, $X = 1000\lambda$ and the small cluster radii are 40λ and 60λ respectively. The results of these tests are shown in Figure 5-17, Figure 5-18, Figure 5-19, and Figure 5-20. The last test shows the computational limit of algorithm in existing workstation. All test results are satisfactory as before.

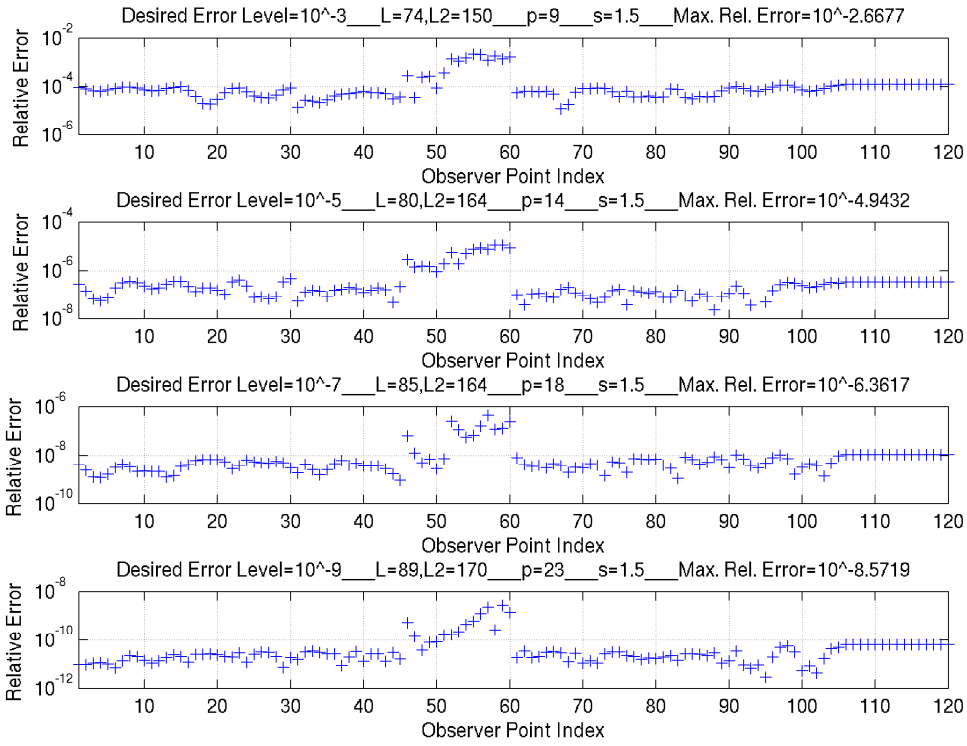

Figure 5-17: Relative errors of field values at observer dipoles at two-level FMM for the configuration $d = 4\lambda$ and $X = 1000\lambda$.

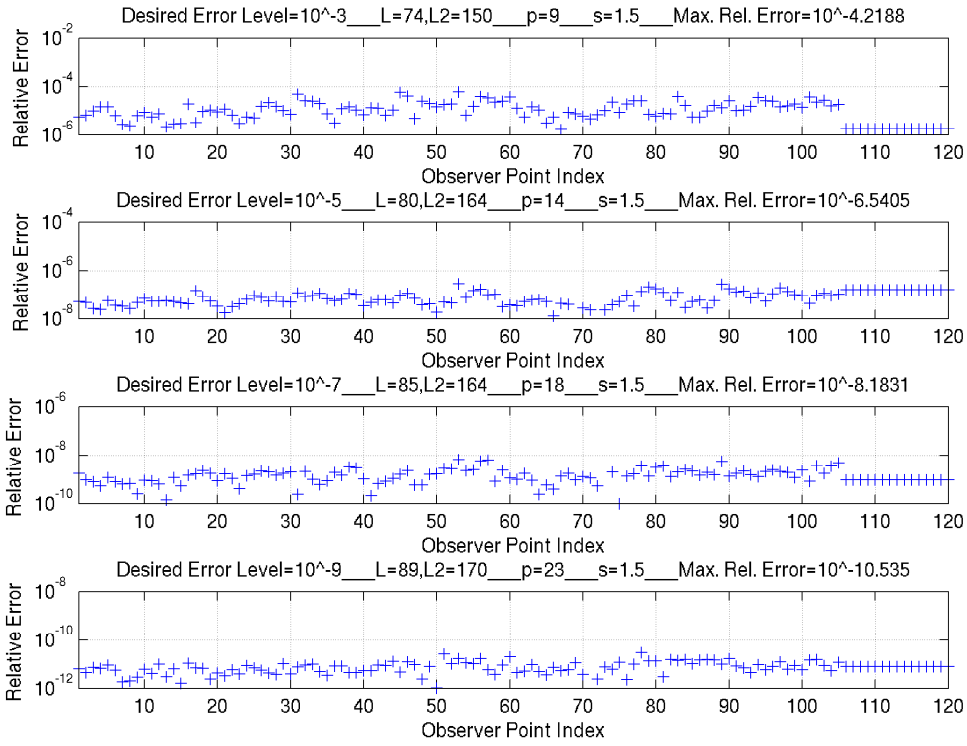

Figure 5-18: Relative errors of field values at observer dipoles at two-level FMM for the configuration $d = 4\lambda$ and $X = 38.62\lambda$.

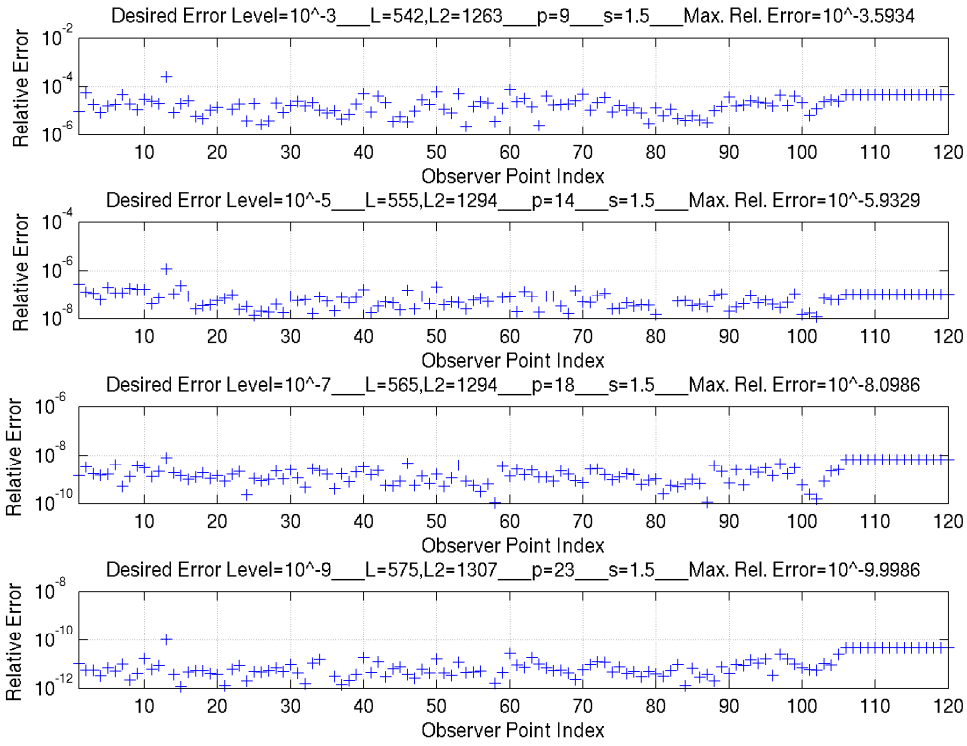

Figure 5-19: Relative errors of field values at observer dipoles at two-level FMM for the configuration $d = 40\lambda$ and $X = 1000\lambda$.

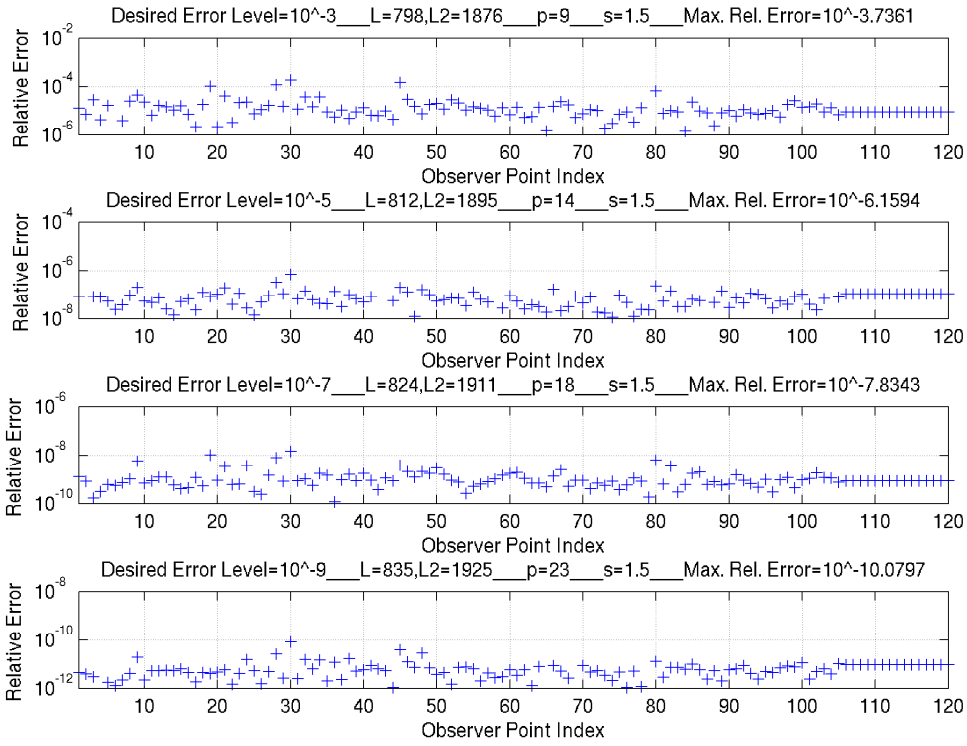

Figure 5-20: Relative errors of field values at observer dipoles at two-level FMM for the configuration $d = 60\lambda$ and $X = 1000\lambda$.

SUMMARY AND FUTURE WORK

In this thesis, efficient and accurate numerical techniques for determining the key parameters in MLFMA are presented. The scalar and vector spherical filters, which significantly affect the accuracy and performance of MLFMA, are extensively discussed. In addition, numerical implementations of FMM and MLFMA to electromagnetic surface scattering problem are provided.

Prior to this study, key parameters in MLFMA, the number of multipoles L, the number of interpolation points p, and the over-sampling ratio s, were being determined through some heuristic formulas, which don't either work in all possible cases or yield precise results. The well-known excess bandwidth formula [98] to determine the number of multipoles fails in the cases where the spacing between interacting clusters isn't large enough. And the formulas to determine the local interpolation parameters (p and s) in [103] give rise to poor estimations. Estimated parameters by these formulas yield the maximum error rates very below or above the desired error rates. Consequently, who uses existing formulas in literature to determine these key parameters can encounter unexpected overall error rates at the end of MLFMA executions. By using the self-tuning library presented in this study, these key parameters can be accurately estimated and MLFMA can be fully error controllable. The results presented in Chapter 4 and Chapter 5 show the power of estimator algorithms and the accurateness of the expected overall error rates at the end of MLFMA execution.

Scalar and vector spherical filters used in upward and downward traversal in Helmholtz MLFMA and Maxwell MLFMA are discussed extensively in Chapter 3. After the standard (slow) scalar spherical filtering procedure is described, the fast spherical

filtering procedure introduced in literature is provided with its complexity analysis. Although the literature for scalar spherical filter is abundant, that for vector spherical filter is scarce. Existing formulation for the fast vector spherical filter is incorrect and explanation of its implementation is hard to comprehend. Here, the correct derivation of vector spherical filter is provided by first reviewing the vector spherical harmonics bases and the standard vector filtering procedure. Then, the implementation details of the fast vector spherical filter along with the fast scalar spherical filter are provided to ease the workload of who would like to implement fast vector spherical filter.

In Chapter 2, implementations of FMM and MLFMA to CFIE to solve electromagnetic surface scattering problem are explained comprehensively. After the integral equations are reviewed, the core of the FMM approach, approximating the free-space Green's function, is discussed. Once the approximate expression is derived, its utilization in CFIE is discussed. Multilevel implementation of FMM for CFIE is also explained. At the end of the chapter, an optimal local interpolation scheme for locally interpolating the translation operator values in FMM and MLFMA is elucidated.

By the accomplished work in this study, Helmholtz and Maxwell problems can be solved with a desired precision level. MLFMA can be utilized for integral equations based on free-space kernels. Present code can be parallelized for solving large-scale problems [105].

REFERENCES

[1] J. Dongarra and F. Sullivan, "Guest Editors Introduction to the top 10 algorithms," *Computing in Science & Engineering*, vol. 2, pp. 22-23, 2000.

[2] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *Journal of Computational Physics*, vol. 73, pp. 325-348, 1987.

[3] H. Cheng, L. Greengard, and V. Rokhlin, "A fast adaptive multipole algorithm in three dimensions," *J. Comput. Phys.*, vol. 155, pp. 468-498, 1999.

[4] L. Greengard and J. Lee, "A direct adaptive Poisson solver of arbitrary order accuracy," *J. Comput. Phys.*, vol. 125, pp. 415-424, 1996.

[5] L. Greengard and V. Rokhlin, "A new version of the fast multipole method for the laplace equation in three dimensions," *Acta Numer*, vol. 6, pp. 229-269, 1997.

[6] A. McKenney, L. Greengard, and A. Mayo, "A fast Poisson solver for complex geometries," *J. Comput. Phys.*, vol. 118, pp. 348-355, 1995.

[7] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White, "Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory," *SIAM J. Sci. Comput.*, vol. 15, pp. 713-735, 1994.

[8] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: a pedestrian prescription," *Antennas and Propagation Magazine, IEEE*, vol. 35, pp. 7-12, 1993.

[9] V. Rokhlin, "Diagonal Forms of Translation Operators for the Helmholtz Equation in Three Dimensions," *Applied and Computational Harmonic Analysis*, vol. 1, pp. 82-93, 1993.

[10] L. R. Hamilton, P. A. Macdonald, M. A. Stalzer, R. S. Turley, J. L. Visher, and S. M. Wandzura, "3D method of moments scattering computations using the fast multipole method," in *Antennas and Propagation Society International Symposium, 1994. AP-S. Digest*, 1994, pp. 435-438 vol.1.

[11] J. M. Song and W. C. Chew, "Fast Multipole Method Solution of Combined Field Integral Equation," in *11th Annual Review of Progress in Applied Computational Electromagnetics*. vol. 1 Monterey, California, March 1995, pp. 629-636.

[12] J. M. Song, C. C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *Antennas and Propagation, IEEE Transactions on,* vol. 45, pp. 1488-1493, 1997.

[13] G. Biros, L. Ying, and D. Zorin, "A fast solver for the Stokes equations with distributed forces in complex geometries," *Journal of Computational Physics,* vol. 193, pp. 317-348, 2004.

[14] L. Greengard, M. C. Kropinski, and A. Mayo, "Integral Equation Methods for Stokes Flow and Isotropic Elasticity in the Plane," *Journal of Computational Physics,* vol. 125, pp. 403-414, 1996.

[15] L. Ying, G. Biros, and D. Zorin, "A kernel-independent adaptive fast multipole algorithm in two and three dimensions," *J. Comput. Phys.,* vol. 196, pp. 591-626, 2004.

[16] W. Dehnen, "A Hierarchical (N) Force Calculation Algorithm," *Journal of Computational Physics,* vol. 179, pp. 27-42, 2002.

[17] A. H. Boschitsch, M. O. Fenley, and W. K. Olson, "A Fast Adaptive Multipole Algorithm for Calculating Screened Coulomb (Yukawa) Interactions," *Journal of Computational Physics,* vol. 151, pp. 212-241, 1999.

[18] L. F. Greengard and J. Huang, "A new version of the fast multipole method for screened Coulomb interactions in three dimensions," *J. Comput. Phys.,* vol. 180, pp. 642-658, 2002.

[19] V. Rokhlin, "Rapid solution of integral equations of scattering theory in two dimensions," *J. Comput. Phys.,* vol. 86, pp. 414-439, 1990.

[20] C. C. Lu and W. C. Chew, "Fast algorithm for solving hybrid integral equations [EM wave scattering]," *Microwaves, Antennas and Propagation, IEE Proceedings H,* vol. 140, pp. 455-460, 1993.

[21] H.-Y. Chao, J.-S. Zhao, and W. C. Chew, "Application of curvilinear basis functions and MLFMA for radiation and scattering problems involving curved PEC structures," *Antennas and Propagation, IEEE Transactions on,* vol. 51, pp. 331-336, 2003.

[22] W. C. Chew, C. Tie Jun, and J. M. Song, "A FAFFA-MLFMA algorithm for electromagnetic scattering," *Antennas and Propagation, IEEE Transactions on,* vol. 50, pp. 1641-1649, 2002.

[23] Y.-H. Chu and W. C. Chew, "A multilevel fast multipole algorithm for electrically small composite structures," *Microwave and Optical Technology Letters,* vol. 43, pp. 202-207, 2004.

[24] T. J. Cui, W. C. Chew, G. Chen, and J. Song, "Efficient MLFMA, RPFMA, and FAFFA algorithms for EM scattering by very large structures," *Antennas and Propagation, IEEE Transactions on,* vol. 52, pp. 759-770, 2004.

[25] K. C. Donepudi, J. Jian-Ming, S. Velamparambil, J. Song, and W. C. Chew, "A higher order parallelized multilevel fast multipole algorithm for 3-D scattering," *Antennas and Propagation, IEEE Transactions on,* vol. 49, pp. 1069-1078, 2001.

[26] K. C. Donepudi, J. Song, J. M. Jin, G. Kang, and W. C. Chew, "A novel implementation of multilevel fast multipole algorithm for higher order Galerkin's method," *Antennas and Propagation, IEEE Transactions on,* vol. 48, pp. 1192-1197, 2000.

[27] L. J. Jiang and W. C. Chew, "A mixed-form fast multipole algorithm," *Antennas and Propagation, IEEE Transactions on,* vol. 53, pp. 4145-4156, 2005.

[28] S. Koc and W. C. Chew, "Multilevel Fast Multipole Algorithm for the Discrete Dipole Approximation," *Journal of Electromagnetic Waves and Applications,* vol. 15, pp. 1447-1468, 2001.

[29] C.-C. Lu and W. C. Chew, "A multilevel algorithm for solving a boundary integral equation of wave scattering," *Microwave and Optical Technology Letters,* vol. 7, pp. 466-470, 1994.

[30] Y. C. Pan and W. C. Chew, "A hierarchical fast-multipole method for stratified media," *Microwave and Optical Technology Letters,* vol. 27, pp. 13-17, 2000.

[31] J. Song and W. C. Chew, "The Fast Illinois Solver Code: requirements and scaling properties," *Computational Science & Engineering, IEEE,* vol. 5, pp. 19-23, 1998.

[32] J. M. Song and W. C. Chew, "Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering," *Microwave and Optical Technology Letters,* vol. 10, pp. 14-19, 1995.

[33] J. M. Song, C. C. Lu, W. C. Chew, and S. W. Lee, "Fast Illinois solver code (FISC)," *Antennas and Propagation Magazine, IEEE,* vol. 40, pp. 27-34, 1998.

[34] S. Velamparambil, W. C. Chew, and J. Song, "10 million unknowns: is it that big? [computational electromagnetics]," *Antennas and Propagation Magazine, IEEE,* vol. 45, pp. 43-58, 2003.

[35] R. L. Wagner and W. C. Chew, "A ray-propagation fast multipole algorithm," *Microwave and Optical Technology Letters,* vol. 7, pp. 435-438, 1994.

[36] J.-S. Zhao and W. C. Chew, "Three-dimensional multilevel fast multipole algorithm from static to electrodynamic," *Microwave and Optical Technology Letters,* vol. 26, pp. 43-48, 2000.

[37] A. A. Ergin, B. Shanker, and E. Michielssen, "Fast evaluation of three-dimensional transient wave fields using diagonal translation operators," *J. Comput. Phys.,* vol. 146, pp. 157-180, 1998.

[38] B. Shanker, A. A. Ergin, M. Lu, and E. Michielssen, "Fast analysis of transient electromagnetic scattering phenomena using the multilevel plane wave time domain algorithm," *Antennas and Propagation, IEEE Transactions on,* vol. 51, pp. 628-641, 2003.

[39] K. Aygun, B. Shanker, A. A. Ergin, and E. Michielssen, "A two-level plane wave time-domain algorithm for fast analysis of EMC/EMI problems," *Electromagnetic Compatibility, IEEE Transactions on,* vol. 44, pp. 152-164, 2002.

[40] A. A. Ergin, B. Shanker, and E. Michielssen, "The plane-wave time-domain algorithm for the past analysis of transient wave phenomena," *IEEE Antennas and Propagation Magazine,* vol. 41, pp. 39-52, 1999.

[41] M. Y. Lu, M. Lv, A. A. Ergin, B. Shanker, and E. Michielssen, "Multilevel plane wave time domain-based global boundary kernels for two-dimensional finite difference time domain simulations," *Radio Science,* vol. 39, 2004.

[42] B. Shanker, A. A. Ergin, K. Aygun, and E. Michielssen, "Analysis of transient electromagnetic scattering phenomena using a two-level plane wave time-domain algorithm," *IEEE Transactions on Antennas and Propagation,* vol. 48, pp. 510-523, 2000.

[43] B. Shanker, A. A. Ergin, and E. Michielssen, "Plane-wave-time-domain-enhanced marching-on-in-time scheme for analyzing scattering from homogeneous dielectric structures," *Journal of the Optical Society of America a-Optics Image Science and Vision,* vol. 19, pp. 716-726, 2002.

[44] M. S. Warren and J. K. Salmon, "A parallel hashed Oct-Tree N-body algorithm," in *Proceedings of the 1993 ACM/IEEE conference on Supercomputing* Portland, Oregon, United States: ACM, 1993.

[45] W. C. Chew, S. Koc, J. M. Song, C. C. Lu, and E. Michielssen, "A succinct way to diagonalize the translation matrix in three dimensions," *Microwave and Optical Technology Letters,* vol. 15, pp. 144-147, 1997.

[46] M. A. Epton and B. Dembart, "Multipole translation theory for the three-dimensional Laplace and Helmholtz equations," *SIAM J. Sci. Comput.,* vol. 16, pp. 865-897, 1995.

[47] B. Dembart and E. Yip, "The accuracy of fast multipole methods for Maxwell's equations," *Computational Science & Engineering, IEEE,* vol. 5, pp. 48-56, 1998.

[48] B. Dembart and E. Yip, "A 3D fast multipole method for electromagnetics with multiple levels," in *11th Annual Review of Progress in Applied Computational Electromagnetics*. vol. 1995 Monterey, California, March 1995, pp. 621-628.

[49] M. Gyure, F. and M. Stalzer, A. , "A Prescription for the Multilevel Helmholtz FMM," *IEEE Comput. Sci. Eng.,* vol. 5, pp. 39-47, 1998.

[50] W. C. Chew, J. M. Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*: Artech House, 2001.

[51] J. R. Mautz and R. F. Harrington, "H-field, E-field and combined-field solutions for conducting bodies of revolution," *Arch. Elektron. Übertragungstechn. (Electron. Commun.),* vol. 32, pp. 157-164, 1978.

[52] G. B. Antilla and N. G. Alexopoulos, "Scattering from complex three-dimensional geometries by a curvilinear hybrid finite-element-integral equation approach," *J. Opt. Soc. Am.,* vol. 11, pp. 1513-1527, 1994.

[53] S. Rao, D. Wilton, and A. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *Antennas and Propagation, IEEE Transactions on,* vol. 30, pp. 409-418, 1982.

[54] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions*: Dover Publications., 1964.

[55] G. B. Arfken and H. J. Weber, *Mathematical Methods For Physicists*, 6th ed.: Academic Press, 2005.

[56] J. A. Stratton, *Electromagnetic Theory*, First ed. New York: McGraw Hill, 1941.

[57] A. D. McLaren, "Optimal Numerical Integration on a Sphere," *Mathematics of Computation,* vol. 17, pp. 361-383, 1963.

[58] M. S. Warren and J. K. Salmon, "A portable parallel particle program," *Computer Physics Communications,* vol. 87, pp. 266-290, 1995.

[59] E. Darve, "The Fast Multipole Method: Numerical Implementation," *Journal of Computational Physics,* vol. 160, pp. 195-240, 2000.

[60] J. Song and W. C. Chew, "Interpolation of translation matrix in MLFMA," *Microwave and Optical Technology Letters,* vol. 30, pp. 109-114, 2001.

[61] B. K. Alpert and V. Rokhlin, "A fast algorithm for the evaluation of Legendre expansions," *SIAM J. Sci. Stat. Comput.,* vol. 12, pp. 158-179, 1991.

[62] S. Velamparambil and W. C. Chew, "A fast polynomial representation for the translation operators of an MLFMA," *Microwave and Optical Technology Letters,* vol. 28, pp. 298-303, 2001.

[63] O. M. Bucci, C. Gennarelli, and C. Savarese, "Optimal interpolation of radiated fields over a sphere," *Antennas and Propagation, IEEE Transactions on,* vol. 39, pp. 1633-1643, 1991.

[64] J. Knab, "Interpolation of band-limited functions using the approximate prolate series (Corresp.)," *Information Theory, IEEE Transactions on,* vol. 25, pp. 717-720, 1979.

[65] O. Ergul and L. Gurel, "Enhancing the Accuracy of the Interpolations and Anterpolations in MLFMA," *IEEE Antennas Wireless Propagat. Lett.,* vol. 5, pp. 467-470, 2006.

[66] M. Böhme and D. Potts, "A fast algorithm for filtering and wavelet decomposition on the sphere.," *Electron. Trans. Numer. Anal.,* pp. 70 - 92, 2003.

[67] D. M. Healy, D. N. Rockmore, P. J. Kostelec, and S. Moore, "FFTs for the 2-Sphere-Improvements and Variations," *Journal of Fourier Analysis and Applications,* vol. 9, pp. 341-385, 2003.

[68] R. Jakob-Chien and B. K. Alpert, "A Fast Spherical Filter with Uniform Resolution," *Journal of Computational Physics,* vol. 136, pp. 580-584, 1997.

[69] S. Kunis and D. Potts, "Fast spherical Fourier algorithms," *J. Comput. Appl. Math.,* vol. 161, pp. 75-98, 2003.

[70] M. J. Mohlenkamp, "A Fast Transform for Spherical Harmonics." vol. PhD dissertation: Yale University, 1997.

[71] M. J. Mohlenkamp, "A fast transform for spherical harmonics," *Journal of Fourier Analysis and Applications*, vol. 5, pp. 159-184, 1999.

[72] Y. Norman and R. Vladimir, "A generalized one-dimensional fast multipole method with application to filtering of spherical harmonics," *J. Comput. Phys.*, vol. 147, pp. 594-609, 1998.

[73] S. A. Orszag, "Fast eigenfunction transforms," in *Science and Computers* G. C. Rota, Ed. New York: Academic Press, 1986, pp. 23 - 30.

[74] V. Rokhlin and M. Tygert, "Fast Algorithms for Spherical Harmonic Expansions," *SIAM J. Sci. Comput.*, vol. 27, pp. 1903-1928, 2006.

[75] W. F. Spotz and P. N. Swarztrauber, "A performance comparison of associated Legendre projections," *J. Comput. Phys.*, vol. 168, pp. 339-355, 2001.

[76] R. Suda and S. Kuriyama, "Another preprocessing algorithm for generalized one-dimensional fast multipole method," *J. Comput. Phys.*, vol. 195, pp. 790-803, 2004.

[77] R. Suda and M. Takami, "A fast spherical harmonics transform algorithm," *Math. Comput.*, vol. 71, pp. 703-715, 2002.

[78] P. N. Swarztrauber and W. F. Spotz, "Generalized discrete spherical harmonic transforms," *J. Comput. Phys.*, vol. 159, pp. 213-230, 2000.

[79] M. Tygert, "Fast Algorithms for Spherical Harmonic Expansions II," Program in applied mathematics, Yale University, New Haven, CT, technical report 1381, 2007.

[80] J. D. Jackson, *Classical Electrodynamics*, Third ed.: Wiley, 1998.

[81] M. J. Mohlenkamp, "A User's Guide to Spherical Harmonics," Ohio University - Mathematics, Athens, OH 2007.

[82] S. A. Holmes and W. E. Featherstone, "A unified approach to the Clenshaw summation and the recursive computation of very high degree and order normalised associated Legendre functions," *Journal of Geodesy*, vol. 76, p. 279, 2002.

[83] D. W. Lozier and J. M. Smith, "Algorithm 567: Extended-Range Arithmetic and Normalized Legendre Polynomials [A1], [C1]," *ACM Trans. Math. Softw.*, vol. 7, pp. 141-146, 1981.

[84] F. W. J. Olver and J. M. Smith, "Associated Legendre Functions on the Cut," *Journal of Computational Physics*, vol. 51, p. 502, 1983.

[85] J. M. Smith, F. W. J. Olver, and D. W. Lozier, "Extended-Range Arithmetic and Normalized Legendre Polynomials," *ACM Trans. Math. Softw.*, vol. 7, pp. 93-105, 1981.

[86] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in Fortran*: Cambridge University Press, 1992.

[87] M. Wieczorek, "SHTOOLS," http://www.ipgp.jussieu.fr/~wieczor/SHTOOLS/www/legendre.html.

[88] T. N. Krishnamurti, V. M. Hardiker, H. S. Bedi, and L. Ramaswamy, "Multilevel Global Spectral Model," in *An Introduction to Global Spectral Modeling*, 2006, p. 112.

[89] M. Frigo and S. G. Johnson., "FFTW, C subroutine library," http://www.fftw.org.

[90] T. Sauer, *Numerical Analysis*: Addison Wesley, 2007.

[91] A. Dutt, M. Gu, and V. Rokhlin, "Fast algorithms for polynomial interpolation, integration, and differentiation," *SIAM J. Numer. Anal.*, vol. 33, pp. 1689-1711, 1996.

[92] T. N. Krishnamurti, V. M. Hardiker, H. S. Bedi, and L. Ramaswamy, "Mathematical Aspects of Spectral Models," in *An Introduction to Global Spectral Modeling*, 2006, pp. 76-111.

[93] E. W. Hobson, The Theory of Spherical and Ellipsoidal Harmonics: Cambridge Press, 1931.

[94] R. G. Barrera, G. A. Estevez, and J. Giraldo., "Vector spherical harmonics and their application to magnetostatics," *European Journal of Physics*, vol. 6, pp. 287-294, 1985.

[95] S. Koc, S. Jiming, and W. C. Chew, "Error Analysis for the Numerical Evaluation of the Diagonal Forms of the Scalar Spherical Addition Theorem," *SIAM J. Numer. Anal.*, vol. 36, pp. 906-921, 1999.

[96] M. Nilsson, "Stability of the High Frequency Fast Multipole Method for Helmholtz' Equation in Three Dimensions," *BIT Numerical Mathematics*, vol. 44, pp. 773-791, 2004.

[97] V. Rokhlin, "Sparse Diagonal Forms for Translation Operations for the Helmholtz Equation in Two Dimensions," Department of Computer Science, Yale University, New Haven, CT, 1995.

[98] J. Song and W. C. Chew, "Error analysis for the truncation of multipole expansion of vector Green's functions [EM scattering]," *Microwave and Wireless Components Letters, IEEE*, vol. 11, pp. 311-313, 2001.

[99] S. Ohnuki and W. C. Chew, "Truncation Error Analysis of Multipole Expansion," *SIAM J. Sci. Comput.*, vol. 25, pp. 1293-1306, 2003.

[100] M. L. Hastriter, S. Ohnuki, and W. C. Chew, "Error control of the translation operator in 3D MLFMA," *Microwave and Optical Technology Letters*, vol. 37, pp. 184-188, 2003.

[101] E. Darve, "The Fast Multipole Method I: Error Analysis and Asymptotic Complexity," *SIAM J. Numer. Anal.*, vol. 38, pp. 98-128, 2000.

[102] J. Kiefer, "Sequential minimax search for a maximum," *Proceedings of the American Mathematical Society*, vol. 4, pp. 502–506, 1953.

[103] S. Ohnuki and W. C. Chew, "Numerical analysis of local interpolation error for 2D-MLFMA," *Microwave and Optical Technology Letters*, vol. 36, pp. 8-12, 2003.

[104] O. Ergul and L. Gurel, "Optimal Interpolation of Translation Operator in Multilevel Fast Multipole Algorithm," *IEEE Trans. Antennas Propagat.*, vol. 54, pp. 3822-3826, 2006.

[105] A. C. Yucel, F. P. Andriulli, and E. Michielssen, "Parallel self-tuning MLFMA library," presented at URSI XXIX General Assembly, 2008