IMPLEMENTATION OF A BROADBAND MULTILEVEL FAST MULTIPOLE ALGORITHM FOR MULTISCALE ELECTROMAGNETICS PROBLEMS

A DISSERTATION SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

By

Manouchehr Takrimi

June, 2016

Implementation of a Broadband Multilevel Fast Multipole Algorithm for Multiscale Electromagnetics Problems

By Manouchehr Takrimi

June, 2016

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Vakur Behçet Ertürk (Advisor)

Özgür Ergül (Co-advisor)

Ayhan Altıntaş

Lale Alatan

Orhan Arıkan

Sencer Koç

Approved for the Graduate School of Engineering and Science:

Levent Onural
Director of the Graduate School

ABSTRACT

IMPLEMENTATION OF A BROADBAND MULTILEVEL FAST MULTIPOLE ALGORITHM FOR MULTISCALE ELECTROMAGNETICS PROBLEMS

Manouchehr Takrimi
Ph.D. in Electrical and Electronics Engineering
Advisor: Vakur Behçet Ertürk
June, 2016

Fast multipole method (FMM) in computational physics and its multilevel version, i.e., multilevel fast multipole algorithm (MLFMA) in computational electromagnetics are among the best known methods to solve integral equations (IEs) in the frequency-domain. MLFMA is well-accepted in the computational electromagnetic (CEM) society since it provides a full-wave solution regarding Helmholtz-type electromagnetics problems. This is done by discretizing proper integral equations based on a predetermined formulation and solving them numerically with $\mathcal{O}(N \log N)$ complexity, where N is the number of unknowns. In this dissertation, we present two broadband and efficient methods in the context of MLFMA, one for surface integral equations (SIEs) and another for volume integral equations (VIEs), both of which are capable of handling large multiscale electromagnetics problems with a wide dynamic range of mesh sizes. By invoking a novel concept of incomplete-leaf tree structures, where only the overcrowded boxes are divided into smaller ones for a given population threshold, a versatile method for both types of IEs has been achieved. Regarding SIEs, for geometries containing highly overmeshed local regions, the proposed method is always more efficient than the conventional MLFMA for the same accuracy, while it is always more accurate if the efficiency is comparable. Regarding VIEs, for inhomogeneous dielectric objects possessing variable mesh sizes due to different electrical properties, in addition to obtaining similar results from the proposed method, a reduction in the storage is also achieved. Several canonical and also real-life examples are provided to demonstrate the superior efficiency and accuracy of the proposed algorithm in comparison to the conventional MLFMA.

Keywords: Multilevel Fast Multipole Algorithm, Integral equations, Incomplete-leaf tree.

ÖZET

GENİŞ BANTLI ÇOK-SEVİYELİ HIZLI ÇOKKUTUP YÖNTEMİNİN ÇOK-ÖLÇEKLİ ELEKTROMANYETİK PROBLEMLER İÇİN GERÇEKLEŞTİRİLMESİ

Manouchehr Takrimi
Elektrik ve Elektronik Mühendisliği, Doktora
Tez Danışmanı: Vakur Behçet Ertürk
Haziran, 2016

Hesaplamalı fizikte kullanılan hızlı çokkutup yöntemi (HÇY) ile bu yöntemin çok-seviyeli biçimi olan çok-seviyeli hızlı çokkutup yöntemi (ÇSHÇY), frekans uzayında integral denklemlerinin (İD) çözümü için bilinen en iyi metotlardandır. ÇSHÇY, Helmholtz tipi elektromanyetik problemlere tam-dalga çözümleri sunabilmesi sayesinde, hesaplamalı elektromanyetik (HEM) camiası içinde büyük kabul görmektedir. Bu yöntem, uygun integral denkleminin önceden belirlenmiş formülasyonlara dayanarak ayrıklaştırılabilmesi ve N sayıda bilinmeyen için $\mathcal{O}(N \log N)$ karmaşıklığı ile sayısal olarak çözülebilmesi ile gerçekleştirilmektedir. Bu doktora tezinde, geniş bir aralıkta örgü boylarıyla ayrıklaştırılan çok ölçekli elektromanyetik problemlerinde kullanılabilecek, biri yüzey integral denklemleri (YİD), diğeri ise hacim integral denklemleri (HİD) için, ÇSHÇY kapsamında geliştirilmiş iki geniş bantlı ve verimli yöntem sunmaktayız. Orjinal bir kavram olan ve sadece önceden belirlenen bir popülasyon eşiğinden daha kalabalık kutuların küçük kutulara bölündüğü tamamlanmamış-yaprak ağaç yapılarından yararlanılarak, iki tipteki İD için de kullanılabilen çok yönlü bir yöntem geliştirilmiştir. Önerilen yöntem aynı doğruluk için, YİD'lerde fazla ağla örgülenmiş yerel bölgeler içeren geometrilerde her zaman geleneksel ÇSHÇY'den daha verimli olmakta, verimliliğin kıyaslanabilir olduğu durumlarda ise her zaman daha doğru sonuçlar vermektedir. HİD'lerde ise homojen olmayan, farklı elektriksel özelliklerden dolayı çok değişken ağla örgülenmiş dielektrik nesneler için, önerilen yöntemden YİD'lerdekine benzer sonuçlar elde edilmesine ek olarak, bellek gereksiniminde büyük bir azalma elde edilmiştir. Önerilen yöntemin geleneksel ÇSHÇY'ye kıyasla yüksek verimlilik ve doğruluğunun gösterilmesi amacıyla, kuralsal örneklerin yanı sıra gerçek hayattan da örnekler sunulmaktadır.

Anahtar sözcükler: Çok-Seviyeli Hızlı Çokkutup Yöntemi, Integral denklemleri, Tamamlanmamış-yaprak ağaç.

Acknowledgement

The journey of pursuing a PhD degree in a foreign country is challenging. On this special journey, I am indebted to many people for their generous help and support.

First of all, I am very grateful to my advisor Dr. Vakur Behçet Ertürk. He has been quite supportive not only by providing me a research assistantship over the past one and half years, but also guiding me through the whole process of research academically.

Furthermore, I am deeply thankful to Dr. Özgür Ergül as my second advisor through more than three years of using his CEM lecture notes, papers, his invaluable book, and especially his Fortran and recently, MATLAB codes, which gradually build up my coding skills as well as my knowledge in CEM.

Thank to both of them, I had the opportunity to contribute to the development of the MLFMA as one of the best frequency-domain methods well-known for CEM society. It was quite challenging and also very instructive.

Finally but most importantly, I would like to thank my whole family, my father and mother for their unconditional love and constant support, and especially my beloved wife and my dear daughter for being strong, patient and supportive. Without them, I could not accomplish this task.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

These days, the area of computational electromagnetics (CEM) plays a critical role in the analysis of electromagnetic (EM) phenomena and also in design and implementation of high-frequency EM systems. Using modern computer technology and recent numerical algorithms, many practical and real-life problems can be efficiently modeled and simulated within acceptable range of accuracy [1].

A powerful computational algorithm, known as the method of moments (MoM) [2], has been verified to be one of the most versatile and accurate techniques for solving radiation and scattering problems. Apart from many good features, this method has some serious disadvantages. The major one is due to its high computational cost regarding memory consumption and CPU time, both of which are great burdens even for today's modern computers. Hence, fast and accurate solvers have been developed based on iterative solutions of matrix equations along with specialized acceleration techniques for matrix-vector multiplications (MVMs) to empower this method. These accelerations are generally performed in two ways [3] described as follows:

- To use some sets of directional basis and testing functions radiating narrow beams with quasi-sparse impedance matrices, such as
    1. Impedance Matrix Localization (IML) [4],

2. Complex Multipole Beam Approach (CMBA) [5],
3. Wavelet Expansion [6].

- To use mathematical or physical properties of MoM matrix to approximate MVMs, such as

1. Fast Multipole Method (FMM) [7],
2. Multilevel Fast Multipole Method (Algorithm) (MLFMM=MLFMA) [8],
3. Adaptive Integral Method (AIM) [9],
4. Precorrected Fast Fourier Transform (pFFT) [10],
5. Multilevel Matrix Decomposition Algorithm (MLMDA) [11],
6. QR-Based or SVD-Based Methods [12],
7. Adaptive Cross-Approximation (ACA) [13].

Generally speaking, in many of these and other similar methods, the memory and CPU requirements are reduced from $\mathcal{O}(N^2)$ complexity to $\mathcal{O}(N^\alpha)$ $(1 \le \alpha \le 2)$ for single level implementations, and to $\mathcal{O}(N \log^\alpha N)$ for the corresponding multilevel versions, where N denotes the number of unknowns.

Regardless of different approaches used in the above mentioned methods, the ultimate goal is to increase the efficiency and/or accuracy using a wide variety of analytical or numerical algorithms, matrix manipulation techniques, and diverse parallelization schemes. Because we try to accelerate the MVMs by using the mathematical / physical properties of the MoM matrix in this dissertation, we should first focus on the accuracy, and consider the possible error sources commonly encountered in the MoM, FMM and its multilevel version. These errors can be categorized as (a) MoM-related errors due to surface and operator discretizations and integration over basis and testing functions, (b) FMM-related errors due to truncated summations, computations of special functions over unit sphere, and interpolation/anterpolation operations, (c) other computational errors due to various numerical integrations, compression and decomposition of matrices (if any), and residual errors due to iterative solutions. It should be emphasized that the convergence of an iterative solver is an important issue for the total efficiency of any algorithm, so if the object has a complicated shape and it requires discretization with fine elements that are very small compared to the wavelength, then the convergence of these iterative solvers will be problematic.

For the above-mentioned error sources, there are many primitive or elegant remedies, leading to a wide range of novel and practical methods. However, another serious issue regarding the MoM arises when there are a wide variety of different element sizes over the meshed surface to capture the fine geometric features. Using mixed-size elements to capture sharp geometric features embedded inside or around highly overmeshed local regions, another source of error or a significant inefficiency originates from the fact that almost all of the above-mentioned methods deploy fixed-size boxes at each and every level of the corresponding tree structures. In such multiscale problems, one may use very dense meshes over the entire geometry to possess the fine details, so that accurate results can be obtained using fixed-size boxes. However, such an attempt is a brute-force solution with a huge number of unknowns requiring excess amount of memory in addition to being extremely inefficient even when large numbers of levels are employed.

Note that as a heuristic definition, any sort of concentration of large number of unknowns in an electrically small domain leads to a multiscale problem in frequency, length or both. Most of the existing methods face some complications when applied to this class of multiscale problems. Narrow feeding lines for an array of small printed-circuit antennas over a large ground plane is a good example for the aforementioned class of problems. Such problems deteriorate the condition number of the matrix system and hence, causes iterative solvers to converge slowly or become stagnant.

A widely preferred approach is to use nonuniform meshes to discretize multiscale problems. In this approach, one may use large boxes together with lower number of levels to maintain the desired accuracy. Unfortunately, in this case, an $\mathcal{O}(N^2)$ complexity is inevitable in near-field computations due to boxes enclosing locally overcrowded regions. On the other hand, one can use smaller boxes with some deep levels of MLFMA to avoid the $\mathcal{O}(N^2)$ complexity. However, leaf level boxes at deep levels are too small to hold basis/testing functions. Parts of basis/testing functions may be located in different boxes and some of these boxes may even be in the far-zone of each other based on the one-buffer-box criterion [1] that is frequently used in MLFMA implementations. Hence, critical computational errors contaminate the accuracy. Furthermore, these errors become more significant as the multiscale factor, which is defined as the ratio of the largest edge length to the smallest one over the entire meshed surface, increases.

As a multipurpose solution, i.e., to alleviate the low-frequency breakdown, to exploit the high-frequency performance of MLFMA, and in order to overcome the multiscale limitations, this dissertation proposes a novel broadband incomplete-leaf (IL) MLFMA for multiscale electromagnetics problems by invoking a concept of novel IL tree structures, where only overcrowded boxes are divided into the smaller ones. Therefore, for multiscale problems, in addition to using variable mesh sizes, variable box sizes are used so that both $\mathcal{O}(N^2)$ complexity coming from the near-field calculations of overcrowded boxes and the computational errors arising from the protrusion of basis/testing functions are eliminated.

1.1 Literature Review

In 1985, Rokhlin proposed fast multipole method (FMM) to evaluate particle interactions and also to solve static integral equations [14],[15]. Later on it was extended to solve the scattering problems regarding the acoustic waves and subsequently to solve electromagnetic scattering problems in both two and three dimensions [7],[16]. It has been proved that MLFMA [17] achieves $\mathcal{O}(N\ \log N)$ complexity under the conventional fixed mesh size scenario, which may be defined as $ka = \frac{2\pi}{\lambda}a \approx$ constant, where a is the mesh size and λ is the corresponding wavelength. This complexity brings us the possibility of facing with electrically large EM problems [18],[19]. However, this great performance decreases dramatically for high density multi-size meshes or low-frequency problems where MLFMA is subjected to the “sub-wavelength breakdown” [20],[21]. The main reason behind this deficiency comes from the fact that, the spherical Hankel functions utilized

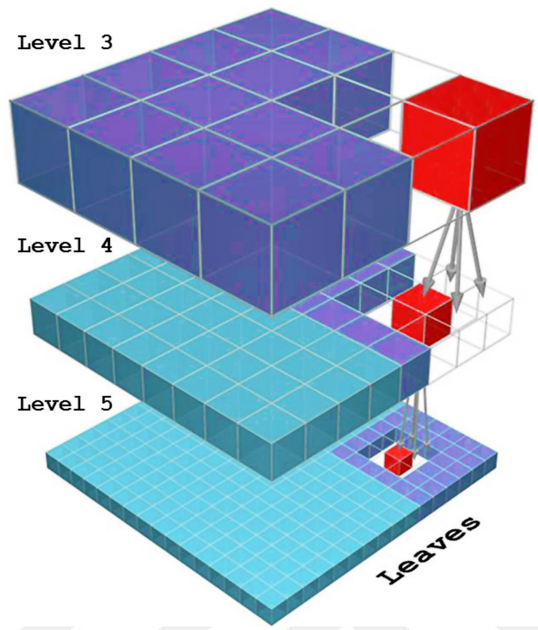


Figure 1.1: Octree structure in FMM. For a given red (black) box on a certain level, only a constant number of interactions take place (dark blue (gray) boxes as far-field and transparent boxes as near-field). The remaining interactions (light blue (gray)) are performed in a different level.

in the derivation of the translation operators are almost singular when the source and observation points are in the close proximity of each other [20]. Consequently, the expansion of the Green's function (using spherical Hankel functions) becomes unstable for multiscale and/or low-frequency EM applications. To alleviate this dilemma, a scaling factor was introduced by Chew [22] to smooth the transition from the high-frequency regime down to the low-frequency Laplace FMM. Although it was a great step forward, later on it has been demonstrated that this approach is not suitable for problems with discretization sizes of multiple orders [23]. To clarify it a bit further, recall that multilevel FMM puts the object in a cubic box and uses a recursive clustering technique to construct an octree structure from nonempty subboxes in consecutive levels (Fig. 1.1), wherein the last level, namely the leaf level (with a minimum box size of $\frac{\lambda}{4}$ to $\frac{\lambda}{16}$), is prone to face with breakdown issues. Vikram *et al.* in [23] introduced Cartesian expansions and claimed that, in order to implement a wideband FMM and overcome the limitation of multiple length/frequency scales, a combination of Cartesian harmonics (for multipoles) and local expansions are exploited for levels beneath the

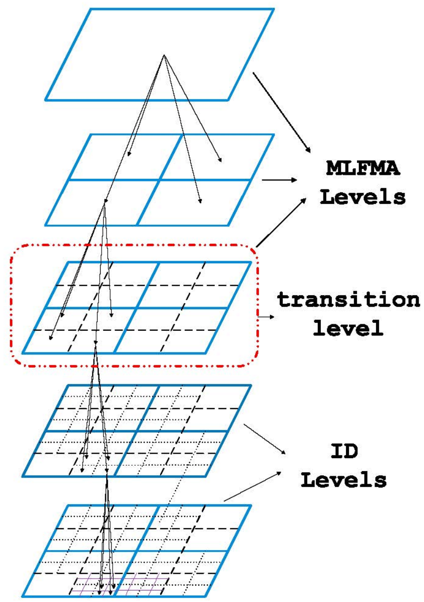


Figure 1.2: Tree structure of the interpolative decomposition MLFMA (ID-MLFMA) [3].

leaf level. Consequently, a mapping procedure has to be performed between accelerated Cartesian expansion (ACE) and fast multipole expansion to accomplish the integration task. Unfortunately, this conversion and additional computational cost at ACE levels for each and every MVM can be considered as a drawback of this method. When the frequency drops, evanescent waves dominate over the propagating waves, hence, an MLFMA implementation using propagating plane-wave expansions encounters a breakdown for translation distances below 0.1λ [24]. In order to simultaneously solve the low-frequency breakdown of MLFMA and high-frequency inefficiency of low-frequency FMM, a mixed-form of FMM has been devised to address a unified broadband FMM [25],[26].

In another method by Rokhlin *et al.* [27], the spectral expansion of the Green's function in terms of evanescent waves and propagating plane-waves was developed. It is applicable for both spatial and frequency scaling but an indefinite integral has to be evaluated in the k-space. The k-space is an extension of the concept of Fourier space which represents the spatial frequency information in two or three dimensions [28]. Another work proposed for wideband FMM [29] redefines the translation procedure within a spherical harmonic space (h-space) and uses complicated matrix rotation operations. Also it requires additional computations to make a bridge between the plane-wave expansion in the high-frequency regime and the partial-wave expansion used in the low-frequency regime.

A slightly different idea to circumvent the difficulties arising from fine meshes was propounded by X.Pen *et al.* by combining the interpolative decomposition (ID) with conventional MLFMA, denoted by ID-MLFMA [3]. As shown in Fig. 1.2, some of the leaf level (i.e., transition level) boxes in terms of the MLFMA, which are chosen based on a predetermined size criterion (e.g., 50 basis elements), are further decomposed as ID levels. Near-field interactions are conducted through the skeleton approximation at ID levels. Far-field interactions in MLFMA levels obey traditional rules, but in ID levels, using interpolative decomposition method [30], very efficient data sparse representation is found to approximate these rank deficient sub-matrices.

Another new method, hierarchical MLFMM (H-MLFMA) was proposed for the solution of multiscale problems. It benefits from two distinct types of basis functions to address two different natures of physics based on electrical size of elements. H-MLFMM employs the so-called skeleton basis for evanescent waves. Some compression techniques along with ACA has been used to compress the near-field matrix and near-to-far mapping matrices [31].

Another novel algorithm, called the nondirective stable plane-wave multilevel fast multipole algorithm (NSPWMLFMA) [32],[33], was presented for low frequencies. The method is based on an analytical expression for a translation operator in the z direction. This translation operator is made numerically stable using a shift of the integration path into the complex plane. A QR-based method is then used to extend the applicability to all the other translation directions. Obviously integration in a complex plane makes the implementation more difficult, and also the computational effort for evaluating the translation operators is increased in comparison to the original MLFMA.

More recently, an approximate method for the diagonalization of the Green's function, which is stable at arbitrary short distances was developed by Ergül *et* *al.* [34]. In this method, scaled spherical functions have been employed for the stabilization of plane-wave expansions, where the value of scaling factor depends on the electrical distance of the translation and shift operations. Simulations based on canonical problems reveal that the proposed method remains stable by increasing the mesh density, which in turn, makes it quite suitable choice for multiscale and low-frequency problems.

By presenting an elective set of diverse methods to get through the bottlenecks of using traditional MLFMA in a class of multiscale and/or wideband real-life problems, we have to choose which strategy is more powerful, yet flexible to proceed on. In this dissertation, we deploy the idea of extending highly populated over-meshed boxes in each level and invoke the approximate method proposed for the diagonalization of the Green's function [34] to present a wideband MLFMA capable of handling multiscale electromagnetics problems.

It should be mentioned that in various advanced computing [36], chemistry [37], biology [38], and physics [39] applications, especially in N-body problems, a modified version of FMM, namely, adaptive FMM has been used. This method employs an adaptive-tree strategy, which visually resembles our proposed IL-tree structures using variable-sized boxes. However, these problems deal with N discrete entities (particles, molecules, masses, etc.) with static (and mostly analytical) kernels. Hence, they use the original FMM with variable-sized boxes with a single level only. Extension of the adaptive FMM to dynamic problems and constructing its multilevel version (for the desired complexity) have not been attempted to the best of our knowledge. In fact, a direct extension of the adaptive FMM would be less accurate compared to the proposed IL-MLFMA, and it would yield problems when parallelization is concerned, as briefly explained in the upcoming chapters. We note that the concept of using a simple hybrid tree structure for a multiscale problem was proposed in [40] with only two different box sizes at the leaf level.

1.2 Some Preliminary Concepts

1.2.1 Surface Integral Equations and Surface Formulations

Surface integral equations (SIE) are used to solve radiation and scattering problems. By defining equivalent electric and/or magnetic current sources on the surface of the metallic or homogeneous dielectric object(s), it is possible to enforce the appropriate boundary conditions (BCs) and formulate the problems. Depending on the BC as well as the testing strategy, four fundamental SIEs may be defined [35]. These are the tangential electric-field integral equation (T-EFIE), the tangential magnetic-field integral equation (T-MFIE), the normal electric-field integral equation (N-EFIE), and the normal magnetic-field integral equation (N-MFIE). In the tangential versions, tangential components of the fields on the surfaces are sampled and tested regarding each BC. However, in the normal versions, the BCs are first rotated via cross-product operations with proper outward normal vectors and then, they are tested accordingly.

Generally speaking, all formulations can be grouped as tangential formulations involving T-EFIE/T-MFIE, normal formulations involving N-EFIE/N-MFIE, and mixed formulations with at least one of each with appropriate coefficients. It is worth to mention that some of these combined formulations are not stable [35] and hence, to solve real-life EM problems a proper combination of SIEs are needed.

1.2.1.1 Perfectly Conducting Objects

For a perfect electric conductor (PEC), all four formulations may be solved independently to carry out the induced electric current on the surface of the object(s). Among them, N-MFIE and T-EFIE are mostly used in the literature [17],[22],[26]. It should be mentioned that for those objects with a closed surface, these formulations suffer from internal resonances. Consequently, it is preferred to use a convex combination of EFIE and MFIE to derive a proper combined-field integral equation (CFIE) formulation [41].

1.2.1.2 Homogeneous Dielectric Objects

For homogeneous dielectric objects, surface formulations may be obtained by testing the BCs on both sides of the surface of the object. This means that, one can linearly combine and simultaneously solve both of the EFIE and MFIE formulations [42] for outer and inner regions to carry out two sets of unknowns, one for the electric surface currents and another one for the magnetic surface currents.

There are different formulations using this approach. Some of the most common ones are as follows [35]. Poggio-Miller-Chang-Harrington-Wu-Tsai (PMCHWT) formulation, normal Müller formulation (NMF), combined tangential formulation (CTF), and combined normal formulation (CNF). Finally, a fully mixed formulation called the electric and magnetic current combined-field integral equation (JMCFIE) can be used by combining all four types of integral equations. It is worth mentioning that the proposed IL-MLFMA may be used for dielectric objects concerning SIEs, but we will not discuss it in this dissertation. However, scattering problems regarding dielectric objects using volume integral equations (VIEs) will be discussed in the last chapter separately.

1.2.2 Discretization

To solve real-life EM problems with complicated shapes and geometries, it is necessary to discretize the SIE (VIE) as well as the surface (volume) of the geometry. This can be accomplished by expanding the equivalent currents over the surface (inside the volume) of the geometry of interest with a set of appropriate basis functions. By invoking the MoM and testing the BCs using a proper formulation, the resultant dense matrix equations can be solved to find the coefficients of those basis functions.

To obtain the elements of the MoM matrix, where each element represents a mutual electromagnetic interaction between a pair of discretized elements, it is very common to use the Galerkin method. By using this method, the same set of functions are used to expand the currents (i.e., basis functions) and also to test the BCs (i.e., testing function) as well. Using such a scheme, tangential, normal, and mixed formulations can be obtained. Two types of these formulations, i.e., the normal and the mixed one contain the identity operator, which is proved to be a well-tested operator. Hence, the resultant MoM matrix that belongs to any formulation containing such an operator is known to be diagonally dominant and well-conditioned [43]. Note that the other formulation type, i.e., the tangential one, does not contain a well-tested identity operator and hence, the resultant MoM matrix may be an ill-conditioned one. This is why it is mostly preferred to combine those formulations from the efficiency point of view.

Unfortunately, existence of a well-tested identity operator may dramatically decrease the accuracy of the solutions. This means that the solution error in some normal formulations in comparison to those obtained from the tangential formulations may be significant [43],[44]. To improve the accuracy of these formulations, it is common to use either higher-order basis functions [45] or some other functions introduced recently like Buffa-Christiansen [46].

1.2.3 Condition Number

Assume that we want to solve a matrix equation, e.g., $\bar{\boldsymbol{A}}\boldsymbol{x} = \boldsymbol{b}$. To find out the possible impact of a small change in $\boldsymbol{b}$ (due to some numerical errors) on the final solution $\boldsymbol{x}$, one can compute and use the condition number of the matrix $\bar{\boldsymbol{A}}$. The condition number $\kappa(\bar{\boldsymbol{A}}) \geq 1$ is defined as

$$\kappa(\bar{\boldsymbol{A}}) = \left\|\bar{\boldsymbol{A}}\right\| \left\|\bar{\boldsymbol{A}}^{-1}\right\| \tag{1.1}$$

where $\|\cdot\|$ denotes a matrix norm. If 2-norm is used to calculate the condition number, then it can be simplified as

$$\kappa(\bar{\boldsymbol{A}}) = \frac{\sigma_{max}(\bar{\boldsymbol{A}})}{\sigma_{min}(\bar{\boldsymbol{A}})}, \tag{1.2}$$

where σ_{max} and σ_{min} are the maximal and minimal singular values of $\bar{\boldsymbol{A}}$. It can be shown that for normal matrices (a matrix is normal if it commutes with its conjugate transpose) [47], the condition number is given by

$$\kappa(\bar{\boldsymbol{A}}) = \left|\frac{\lambda_{max}(\bar{\boldsymbol{A}})}{\lambda_{min}(\bar{\boldsymbol{A}})}\right|, \tag{1.3}$$

where λ_{max} and λ_{min} are the maximum and minimum eigenvalues of $\bar{\boldsymbol{A}}$, respectively. If the condition number is small, the matrix is well conditioned and the solution to the matrix equation converges rapidly via iterative solvers. If the condition number is very large, then the matrix is said to be ill-conditioned. This means that the iterative solver converges slowly, or even it may not converge and become stagnant. A matrix that is not invertible has a condition number equal to infinity.

1.2.4 Multilevel Fast Multipole Algorithm

It is more than 20 years that the fast multipole method (FMM) has been introduced for the efficient solution of EM problems [7]. For such an efficient solution, one needs to choose a proper and convenient integral equation based on one of the above mentioned formulations and use an appropriate set of basis functions to obtain a dense matrix equation, which can be solved iteratively via a Krylov-subspace algorithm. FMM reduced the computational complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N^{1.5})$ using MVM in an iterative manner. Later on, a multilevel version of FMM, namely MLFMA [8], reduced the complexity of MVMs to $\mathcal{O}(N\log N)$ [1]. MLFMA uses a tree structure consisting of $L = \mathcal{O}(\log N)$ levels. The tree is constructed by putting the object inside a cubic box and dividing it recursively into eight equal sub-boxes down to a predetermined level. It should be noted that the smallest box size at the lowest level depends on the desired accuracy. However, when the box size at the lowest level is smaller than $\lambda/10$ where λ is the wavelength, the low-frequency breakdown of MLFMA is expected [1]. After eliminating the empty boxes, the interactions between near (touching) boxes are computed directly. For far boxes, group-by-group interactions between the basis boxes (as radiating ones) and the testing boxes (as receiving ones) are computed based on a diagonalized version of a well-known EM theorem known as addition theorem. Such a group-by-group interactions are carried out in three separate steps called aggregation, translation, and disaggregation. In each and every MVM, these steps (with more details in Section 3.2.2) are performed hierarchically as follows:

Aggregation:

At the lowest level, the radiated fields of individual boxes are obtained by combining the radiation patterns of the basis functions inside each and every box. At higher levels, the radiated fields of a larger box (containing smaller sub-boxes) are computed by combining the radiated fields of its sub-boxes. This strategy continues to be applied for parent boxes upto the topmost level.

Translation:

Outgoing radiated fields, computed and collected at the center of the boxes during the aggregation process, are translated into incoming fields at the center of the receiving boxes. For each box located at any level, there are at most $7^3 - 3^3 = 316$ different translations which are independent of the number of boxes at that level. This is mainly due to the bisecting mechanism of the cubic structure and the strict definition of the far boxes.

Disaggregation;

All of the aggregated and translated incoming fields arriving from different boxes at the same level are collected and combined with those incoming fields arriving from the center of existing higher levels (i.e., parent levels). In the lowest level, testing functions within a testing box, receive a part of this field according to their overall receiving patterns.

In MLFMA outgoing (radiated) and incoming (receiving) fields are sampled on the unit sphere as a function of spherical coordinates θ and ϕ. The number of samples needed for each box is proportional to the electrical size of the box [48]. This means that, to match different sampling rates among the consecutive levels, efficient interpolation and anterpolation techniques are required during aggregation and disaggregation processes, respectively.

In the setup phase of MLFMA, the translation operators are calculated and stored in memory. This is mainly because they are going to be used repeatedly in MVM routines. Another important computation that should be directly calculated and stored in memory for the same reason, concerns with the near-field interactions with a complexity of $\mathcal{O}(N)$. These interactions are either between touching basis and testing boxes or between basis and testing functions within the same box. These calculations are based on a combination of analytic and also numerical methods where the accuracy of the numerical part is controllable and can be increased by sacrificing the efficiency. The analytic calculations are based on well-known singularity extraction techniques [49],[50],[51] and the numerical calculations are mostly based on Gaussian quadratures [52]. More details about these techniques are discussed in Section 2.6.

1.2.5 Low-Frequency Breakdown

As stated in the introduction and briefly reviewed in the literature, MLFMA suffers from an important breakdown when the box sizes at the lowest level (the leaf level) get smaller than a fraction of a wavelength. This is due to the fact that MLFMA is based on the plane-wave expansion and it becomes inaccurate for short distances. Therefore, it is not efficient for problems involving some small objects, which are discretized with large numbers of unknowns. As another type of breakdown, in such sort of problems, the matrix equations obtained from any combination of T-EFIE become increasingly ill-conditioned. However, N-MFIE is usually stable unless the frequency is extremely low [53]. In the first part of this dissertation, we will use MFIE along with Rao-Wilton-Glisson (RWG) [54] vector basis functions to solve scattering problems concerning PEC objects. The aim of this dissertation is not dealing with formulation breakdowns; we consider the breakdowns of MLFMA for dense discretizations encountered in multiscale problems.

1.2.6 Iterative Solvers

To solve the associated matrix equation of SIEs or VIEs efficiently, one needs to use iterative solvers based on Krylov-subspace method. There are many methods proposed in the literature, that perform differently depending on the problem. Obviously, it is not possible to find the best one that fits for all EM problems. Finding the best one for a given problem is a matter of trial and error. Some of the well-known methods are as follows [35]:

- BiCG: Biconjugate Gradient.
- BiCGStab: Biconjugate Gradient Stabilized.
- CG: Conjugate Gradient.
- CGS: Conjugate Gradient Squared.
- GMRES: Generalized Minimal Residual.
- LSQR: Least-Squares QR.
- QMR: Quasi-Minimal Residual.

Sequential and parallel implementations of these iterative algorithms can be found in [55]. In this dissertation, we will use GMRES due to its flexibility of solving wide range of EM problems.

1.3 Problem Statement and Motivation

Considering the aforementioned brief summery about SIEs, MLFMA, and the tree structure, we can deduce important points regarding possible bottlenecks when we start moving from normally meshed structures toward overmeshed ones, known as multiscale problems. We have to emphasize the following issues:

1. Generally speaking, most of the multiscale problems end up with an unbalanced condition regarding the number of unknowns within the leaf level boxes. This means that, overcrowded boxes will constrain the CPU to compute near-field interactions in the context of MoM, which in turn takes much longer time to be computed in comparison with normally loaded boxes. This issue gets worse for VIEs especially if the nearby boxes are also crowded.

2. Dividing all leaf-level boxes to gain one more level, somehow alleviates this puzzle, but it doesn't help the unbalanced distribution of the basis functions within each box. This will be problematic for parallel implementation of MLFMA among multiple CPUs.

3. Sometimes, multiscale nature of the problem prevents further dividing of some boxes. This happens when a group of individual boxes are occupied by large RWG functions, hence, bisecting those boxes may be quite difficult.

4. Increasing the number of levels, effectively decreases the number of near-field interactions, which in turn, lowers the burden of the CPU, at least from the computational point of view, but apparently by overdoing it, another critical issue arises when the smallest box size reaches to a fraction of a wavelength (e.g. under $\lambda/16$). In such cases, the box size at the leaf level is so small that the argument of the translation function tends to be zero and as we will see later, it causes a kind of numerical instability.

1.4 Contribution

We summarize major achievements of this dissertation as follows.

1.4.1 Regarding SIEs

1. We deployed an approximate method for the diagonalization of the Green's function proposed in [34] to handle the well-known low-frequency problems of MLFMA. We modified the method to increase its accuracy and implemented a corrected version of that to propose a broadband incomplete-leaf (IL) MLFMA to solve large multiscale electromagnetic problems.

2. We investigated one of the major sources of error in the solution of SIEs concerning PEC objects. We demonstrated that using the conventional fixed-size boxes in the context of FMM and MLFMA leads to a kind of computational error which is not evident at a first glance, and dramatically contaminate the computation of far-zone interactions, especially in multiscale EM problems where the surface discretization possess a wide range of mesh sizes.

3. We proposed a novel IL-tree structure, where the boxes are bisected into smaller boxes based on their content populations. We introduced new concepts about the near-field interactions and postulated new rules to identify the near and far boxes across the whole tree structure. Using the proposed IL-tree structures, both $\mathcal{O}(N^2)$ complexity arising from the near-field calculations of overcrowded boxes and the hidden computational errors originating from the protrusion of the basis/testing functions are eliminated.

4. We demonstrated that the proposed method is always more efficient than the conventional MLFMA for the same accuracy, while it is always more accurate if the efficiency is comparable. Also we conducted a series of studies to investigate in detail the fact that the accuracy of the IL-MLFMA remains almost fixed regardless of the number of levels, which eliminates

the need of testing different levels of the conventional MLFMA in order to find out the best choice. This provides us an optimization on efficiency without sacrificing the accuracy.

5. We presented a reliable method that always converges iteratively, regardless of the given population threshold.

6. We showed that the IL-MLFMA performs much better for larger EM problems with high multiscale factors, or for medium-sized problems with very high multiscale factors.

7. Due to the versatility of the proposed algorithms, other methods that are proposed to treat the well-known low-frequency breakdown [3],[27],[29],[31],[32],[33] can be used in conjunction with the proposed IL-MLFMA to increase the accuracy.

8. The proposed IL-MLFMA can be combined with other available algorithms, such as the equivalence principle algorithm (EPA) [56], hierarchical MLFMA [31], multi-resolution basis functions [57], and accelerated cartesian expansion (ACE) [23], which are also used together with the conventional MLFMA to attack multiscale electromagnetic problems with very large multiscale factors.

9. For uniform meshes, the IL tree structure reduces to the traditional ones and the proposed IL-MLFMA reduces into the conventional MLFMA if desired.

1.4.2 Regarding VIEs

All achievements regarding the SIEs are mostly achievable in VIEs with the following exceptions / differences.

1. VIEs show no visible low-frequency breakdown, so there is no need to deploy an approximate method as what we did for SIEs. This means that

the implemented IL-MLFMA regarding the volume formulations are more accurate than surface formulations.

2. The protrusion of the basis/testing functions for volume discretizations are much worse than surface discretizations. Therefore, the concept of having a fixed accuracy which is independent of the number of levels is slightly violated.

3. The intrinsic nature of a VIE requires higher amount of memory to be used and processed. Since IL-MLFMA has no practical limit regarding the number of used levels (in order to construct the IL-tree structure), and because IL-MLFMA almost always uses deeper levels of MLFMA, it consumes much less memory than the conventional MLFMA to solve EM problems.

1.5 Organization

This dissertation involves 7 main chapters. In Chapter 2, we commence by some basics about integral equations and briefly review the needed formulations. Then we proceed by introducing a mathematical background regarding FMM and MLFMA in Chapter 3. In Chapter 4, we discuss about a recently developed approximate method that can be used to overcome the low-frequency problem. A rigorous derivation followed by an amending factor to improve its accuracy will be given. Chapter 5 covers the proposed IL-MLFMA and its implementation regarding SIEs. In Chapter 6, we present the same concept of IL-MLFMA dedicated for the VIEs. Finally in Chapter 7, we propose some new ideas for possible future works.

In this dissertation, the e^{-iwt} time convention is used and it is suppressed for the sake of conciseness.

Chapter 2

Basics of EM Theory

2.1 Maxwell's Equations

The frequency domain Maxwell's equations inside a linear, isotropic, and homogeneous medium can be written as

$$\nabla \times \boldsymbol{E}(\boldsymbol{r}) = iw\mu \boldsymbol{H}(\boldsymbol{r}) - \boldsymbol{M}(\boldsymbol{r}) \tag{2.1}$$

$$\nabla \times \boldsymbol{H}(\boldsymbol{r}) = -iw\epsilon \boldsymbol{E}(\boldsymbol{r}) + \boldsymbol{J}(\boldsymbol{r}) \tag{2.2}$$

$$\nabla \cdot \boldsymbol{E}(\boldsymbol{r}) = \tfrac{1}{\epsilon}\rho_e(\boldsymbol{r}) \tag{2.3}$$

$$\nabla \cdot \boldsymbol{H}(\boldsymbol{r}) = \tfrac{1}{\mu}\rho_m(\boldsymbol{r}) \tag{2.4}$$

where $\boldsymbol{r}$ is the position vector and ϵ and μ are the constitutive parameters of the medium, respectively. It should be noted that the magnetic charge density ρ_m and the magnetic current density $\boldsymbol{M}$ are fictitious sources and they are extensively employed in EM related problems as mathematical tools to solve a wide range of scattering and radiation problems. The relation between different electric and magnetic sources are based on continuity equations, given by

$$\nabla \cdot \boldsymbol{J}(\boldsymbol{r}) = iw\rho_e(\boldsymbol{r}) \tag{2.5}$$

$$\nabla \cdot \boldsymbol{M}(\boldsymbol{r}) = iw\rho_m(\boldsymbol{r}). \tag{2.6}$$

By utilizing the potential theory [58], one can write the EM fields in terms of the vector and scalar potentials as

$$\boldsymbol{E}(\boldsymbol{r}) = iw\boldsymbol{A}_m(\boldsymbol{r}) - \nabla\phi_e(\boldsymbol{r}) - \tfrac{1}{\epsilon}\nabla \times \boldsymbol{A}_e(\boldsymbol{r}) \tag{2.7}$$

$$\boldsymbol{H}(\boldsymbol{r}) = iw\boldsymbol{A}_e(\boldsymbol{r}) - \nabla\phi_m(\boldsymbol{r}) + \tfrac{1}{\mu}\nabla \times \boldsymbol{A}_m(\boldsymbol{r}), \tag{2.8}$$

where $\boldsymbol{A}_e$ and $\boldsymbol{A}_m$ are the electric and magnetic vector potentials, respectively, whereas ϕ_e and ϕ_m are the electric and magnetic scalar potentials, respectively. These quantities are related to each other by the Lorentz gauges as

$$\nabla \cdot \boldsymbol{A}_m(\boldsymbol{r}) = iw\epsilon\mu\nabla\phi_e(\boldsymbol{r}) \tag{2.9}$$

$$\nabla \cdot \boldsymbol{A}_e(\boldsymbol{r}) = iw\epsilon\mu\nabla\phi_m(\boldsymbol{r}). \tag{2.10}$$

Using above mentioned potentials, the corresponding gauges, and the Maxwell's equations, a set of four well-know Helmholtz equations can be derived as

$$\nabla^2\boldsymbol{A}_e(\boldsymbol{r}) + k^2\boldsymbol{A}_e(\boldsymbol{r}) = -\epsilon\boldsymbol{M}(\boldsymbol{r}) \tag{2.11}$$

$$\nabla^2\boldsymbol{A}_m(\boldsymbol{r}) + k^2\boldsymbol{A}_m(\boldsymbol{r}) = -\mu\boldsymbol{J}(\boldsymbol{r}) \tag{2.12}$$

$$\nabla^2\phi_e(\boldsymbol{r}) + k^2\phi_e(\boldsymbol{r}) = -\tfrac{1}{\epsilon}\rho_e(\boldsymbol{r}) \tag{2.13}$$

$$\nabla^2\phi_m(\boldsymbol{r}) + k^2\phi_m(\boldsymbol{r}) = -\tfrac{1}{\mu}\rho_m(\boldsymbol{r}), \tag{2.14}$$

for electric and magnetic potentials, and

$$\nabla^2\boldsymbol{E}(\boldsymbol{r}) + k^2\boldsymbol{E}(\boldsymbol{r}) = -iw\mu\boldsymbol{J}(\boldsymbol{r}) + \tfrac{1}{\epsilon}\nabla\rho_e(\boldsymbol{r}) + \nabla \times \boldsymbol{M}(\boldsymbol{r}) \tag{2.15}$$

$$\nabla^2\boldsymbol{H}(\boldsymbol{r}) + k^2\boldsymbol{H}(\boldsymbol{r}) = -iw\epsilon\boldsymbol{M}(\boldsymbol{r}) + \tfrac{1}{\mu}\nabla\rho_m(\boldsymbol{r}) - \nabla \times \boldsymbol{J}(\boldsymbol{r}), \tag{2.16}$$

for the electric and magnetic fields, where $k = w\sqrt{\epsilon\mu}$ is the wavenumber.

Solving the equations in (2.11)-(2.14) for some arbitrary currents and charges results in

$$\boldsymbol{A}_e(\boldsymbol{r}) = \epsilon \int d\boldsymbol{r'} \boldsymbol{M}(\boldsymbol{r'}) g(\boldsymbol{r}, \boldsymbol{r'}) \tag{2.17}$$

$$\boldsymbol{A}_m(\boldsymbol{r}) = \mu \int d\boldsymbol{r'} \boldsymbol{J}(\boldsymbol{r'}) g(\boldsymbol{r}, \boldsymbol{r'}) \tag{2.18}$$

$$\phi_e(\boldsymbol{r}) = \tfrac{1}{\epsilon} \int d\boldsymbol{r'} \rho_e(\boldsymbol{r'}) g(\boldsymbol{r}, \boldsymbol{r'}) \tag{2.19}$$

$$\phi_m(\boldsymbol{r}) = \tfrac{1}{\mu} \int d\boldsymbol{r'} \rho_m(\boldsymbol{r'}) g(\boldsymbol{r}, \boldsymbol{r'}), \tag{2.20}$$

where

$$g(\boldsymbol{r}, \boldsymbol{r'}) = \frac{\exp(ikR)}{4\pi R} = \frac{\exp(ik|\boldsymbol{r} - \boldsymbol{r'}|)}{4\pi|\boldsymbol{r} - \boldsymbol{r'}|} \qquad (\boldsymbol{R} = \boldsymbol{r} - \boldsymbol{r'}) \tag{2.21}$$

is the homogeneous-space Green's function. By substituting the above solutions, i.e., (2.17)-(2.20) together with (2.21) into (2.7) and (2.8), one may arrive at

$$\begin{aligned} \boldsymbol{E}(\boldsymbol{r}) = ik\eta \int d\boldsymbol{r'} \Big[\boldsymbol{J}(\boldsymbol{r'}) + \frac{1}{k^2}\nabla' \cdot \boldsymbol{J}(\boldsymbol{r'})\nabla\Big] g(\boldsymbol{r}, \boldsymbol{r'}) \\ - \int d\boldsymbol{r'} \nabla g(\boldsymbol{r}, \boldsymbol{r'}) \times \boldsymbol{M}(\boldsymbol{r'}) \end{aligned} \tag{2.22}$$

$$\begin{aligned} \boldsymbol{H}(\boldsymbol{r}) = ik\eta^{-1} \int d\boldsymbol{r'} \Big[\boldsymbol{M}(\boldsymbol{r'}) + \frac{1}{k^2}\nabla' \cdot \boldsymbol{M}(\boldsymbol{r'})\nabla\Big] g(\boldsymbol{r}, \boldsymbol{r'}) \\ + \int d\boldsymbol{r'} \nabla g(\boldsymbol{r}, \boldsymbol{r'}) \times \boldsymbol{J}(\boldsymbol{r'}), \end{aligned} \tag{2.23}$$

as a set of solutions for Helmholtz equations regarding arbitrary source distributions $\boldsymbol{J}$ and $\boldsymbol{M}$ within a homogeneous background medium possessing an intrinsic impedance of η. Assuming $\boldsymbol{X}(\boldsymbol{r})$ to be a general position-dependent vector, it is more convenient to define and use the integro-differential $\mathcal{T}$, $\mathcal{K}$ and $\mathcal{I}$ operators [35] as

$$\mathcal{T}\{\boldsymbol{X}\}(\boldsymbol{r}) = ik \int d\boldsymbol{r'} \Big[\boldsymbol{X}(\boldsymbol{r'}) + \frac{1}{k^2}\nabla' \cdot \boldsymbol{X}(\boldsymbol{r'})\nabla\Big] g(\boldsymbol{r}, \boldsymbol{r'}) \tag{2.24}$$

$$\mathcal{K}\{\boldsymbol{X}\}(\boldsymbol{r}) = \int d\boldsymbol{r'} \boldsymbol{X}(\boldsymbol{r'}) \times \nabla' g(\boldsymbol{r}, \boldsymbol{r'}) \tag{2.25}$$

$$\mathcal{I}^{\times n}\{\boldsymbol{X}\}(\boldsymbol{r}) = \hat{\boldsymbol{n}} \times \mathcal{I}\{\boldsymbol{X}\}(\boldsymbol{r}) = \hat{\boldsymbol{n}} \times \boldsymbol{X}(\boldsymbol{r}) \tag{2.26}$$

to write (2.22) and (2.23) in a more compact form for electromagnetic fields as

$$\boldsymbol{E}(\boldsymbol{r}) = \eta\mathcal{T}\{\boldsymbol{J}\}(\boldsymbol{r}) - \mathcal{K}\{\boldsymbol{M}\}(\boldsymbol{r}) \tag{2.27}$$

$$\boldsymbol{H}(\boldsymbol{r}) = \eta^{-1}\mathcal{T}\{\boldsymbol{M}\}(\boldsymbol{r}) + \mathcal{K}\{\boldsymbol{J}\}(\boldsymbol{r}). \tag{2.28}$$

It is worth to mention that, even though these linear operators are generally defined based on volume integrals, they may reduce to surface integrals depending on the problem. We will use these notations through the rest of this dissertation.

2.2 Near and Far Fields

2.2.1 Near Fields

For near-field computations one may directly use (2.22) and (2.23). Apparently, when the observation point $\boldsymbol{r}$ approaches to a source point $\boldsymbol{r'}$, some advanced computational techniques [49],[59] are necessary to evaluate the integrals due to the singular behavior of the Green's function (as will be explained in Sections 2.5.4 and 2.6.2). Note that for many real-life problems there is no need to deal with magnetic sources and (2.27) and (2.28) reduce to

$$\boldsymbol{E}(\boldsymbol{r}) = \eta\mathcal{T}\{\boldsymbol{J}\}(\boldsymbol{r}) \tag{2.29}$$

$$\boldsymbol{H}(\boldsymbol{r}) = \eta^{-1}\mathcal{K}\{\boldsymbol{J}\}(\boldsymbol{r}). \tag{2.30}$$

2.2.2 Far Fields

For far away observation points in which $kr \gg 1$ and $r \gg r'$, it is more convenient to approximate the Green's function and its gradient. Using the well-known far-field approximations, i.e., $|\boldsymbol{R}| = R \approx r - \hat{\boldsymbol{r}} \cdot \boldsymbol{r'}$ for phase variations and $|\boldsymbol{R}| \approx r$

for amplitude variations, one can derive

$$g(\boldsymbol{r},\boldsymbol{r'}) = \frac{\exp(ikR)}{4\pi R} \approx \frac{\exp(ikr)}{4\pi r}\exp(-ik\hat{\boldsymbol{r}}\cdot\boldsymbol{r'}) \tag{2.31}$$

$$\nabla g(\boldsymbol{r},\boldsymbol{r'}) = (\boldsymbol{r}-\boldsymbol{r'})(ikR-1)\frac{\exp(ikR)}{4\pi R^3} \approx ik\hat{\boldsymbol{r}}\frac{\exp(ikr)}{4\pi r}\exp(-ik\hat{\boldsymbol{r}}\cdot\boldsymbol{r'}). \tag{2.32}$$

By inserting (2.31) and (2.32) into $\mathcal{T}$ and $\mathcal{K}$ operators defined in (2.24) and (2.25), and using the divergence theorem, $\mathcal{T}$ and $\mathcal{K}$ operators can be written as [35]

$$\mathcal{T}\{\boldsymbol{X}\}(\boldsymbol{r}) \approx ik\frac{\exp(ikr)}{4\pi r}(\bar{\bar{\boldsymbol{I}}}_{3\times 3}-\hat{\boldsymbol{r}}\hat{\boldsymbol{r}})\cdot\int d\boldsymbol{r'}\boldsymbol{X}(\boldsymbol{r'})\exp(-ik\hat{\boldsymbol{r}}\cdot\boldsymbol{r'}) \tag{2.33}$$

$$\mathcal{K}\{\boldsymbol{X}\}(\boldsymbol{r}) \approx ik\frac{\exp(ikr)}{4\pi r}\hat{\boldsymbol{r}}\times\int d\boldsymbol{r'}\boldsymbol{X}(\boldsymbol{r'})\exp(-ik\hat{\boldsymbol{r}}\cdot\boldsymbol{r'}), \tag{2.34}$$

where $\bar{\bar{\boldsymbol{I}}}_{3\times 3}$ is a unit dyad. Note that due to the existence of $(\bar{\bar{\boldsymbol{I}}}_{3\times 3}-\hat{\boldsymbol{r}}\hat{\boldsymbol{r}})$ in (2.33) and $\hat{\boldsymbol{r}}\times$ in (2.34), we expect no radial component for both approximations at $r=|\boldsymbol{r}|\to\infty$. This means that by ignoring the phase shift, i.e., $\exp(ikr)$, we can define a kind of normalized far-zone EM fields as

$$\boldsymbol{E}^{\infty}(\theta,\phi) = \lim_{r\to\infty}\{r\exp(-ikr)\boldsymbol{E}(\boldsymbol{r})\} \tag{2.35}$$

$$\boldsymbol{H}^{\infty}(\theta,\phi) = \lim_{r\to\infty}\{r\exp(-ikr)\boldsymbol{H}(\boldsymbol{r})\}, \tag{2.36}$$

which depend on the angular direction of the observation point. Note that to solve many real-life EM problems concerning metallic structures, only $\boldsymbol{J}$ is required to be known and hence, by substituting (2.33) into (2.29), one can easily obtain

$$\boldsymbol{E}^{\infty}(\theta,\phi) = \frac{1}{4\pi}k\eta(\bar{\bar{\boldsymbol{I}}}_{3\times 3}-\hat{\boldsymbol{r}}\hat{\boldsymbol{r}})\cdot\int d\boldsymbol{r'}\boldsymbol{J}(\boldsymbol{r'})\exp(-ik\hat{\boldsymbol{r}}\cdot\boldsymbol{r'}) \tag{2.37}$$

$$\boldsymbol{H}^{\infty}(\theta,\phi) = \frac{1}{\eta}\hat{\boldsymbol{r}}\times\boldsymbol{E}^{\infty}(\theta,\phi). \tag{2.38}$$

In the upcoming chapters, after finding a proper solution for induced currents on the surface of the metallic objects, we have to use (2.37) to calculate the far-zone electric fields and compare them with some analytic solutions to study the accuracy of the proposed algorithms.

2.3 Equivalent Problems

It is more convenient to formulate scattering and radiation problems in terms of other equivalent problems that may be easier to solve. These equivalent problems are usually based on either equivalent surface currents for merely homogeneous objects or equivalent volume currents for inhomogeneous objects. In all of these equivalent problems, one tries to mathematically eliminate the presence of the object inside the original problem by introducing some auxiliary surface (volume) currents.

In this dissertation we use two well-know equivalence theorems to establish the corresponding integral equations, i.e., the physical equivalence theorem is used for PEC objects in the framework of SIEs, and volume equivalence theorem is used for inhomogeneous dielectric objects in the framework of VIEs. Solutions of dielectric objects based on SIEs are not addressed here and hence, surface equivalence theorem is not discussed.

2.3.1 Physical Equivalence Theorem

Assume that $\boldsymbol{J}$ and $\boldsymbol{M}$ radiate in a homogeneous region with parameters ϵ and μ creating the incident fields $\boldsymbol{E}^{inc}$ and $\boldsymbol{H}^{inc}$. If we put a PEC object inside the region, the scattered fields $\boldsymbol{E}^{scat}$ and $\boldsymbol{H}^{scat}$ are created outside the object. Obviously the inside fields are zero. This is illustrated in Fig. 2.1(a) as the original problem. We need to find those scattered fields outside of the object. Hence, we introduce an equivalent problem that permits the elimination of the object inside the region. This is useful due to the fact that now we have a homogeneous region again. Fig. 2.1(b) shows the equivalent problem. To do this, one needs to eliminate the original sources and then put appropriate equivalent surface currents on the surface of the PEC object and satisfy the BCs as

$$\boldsymbol{M}_S = -\hat{\boldsymbol{n}} \times (\boldsymbol{E}^{inc} + \boldsymbol{E}^{scat}) = -\hat{\boldsymbol{n}} \times (\boldsymbol{E}^{scat} - (-\boldsymbol{E}^{inc})) = 0 \quad (2.39)$$

$$\boldsymbol{J}_S = \hat{\boldsymbol{n}} \times (\boldsymbol{H}^{inc} + \boldsymbol{H}^{scat}) = \hat{\boldsymbol{n}} \times (\boldsymbol{H}^{scat} - (-\boldsymbol{H}^{inc})). \quad (2.40)$$

Note that the second term at the right hand side of (2.40) implies that $\boldsymbol{J}_S$ by itself can create the scattered fields outside and the negative incident fields inside the eliminated object. Apparently, by taking into account the existence of the original sources, the total fields inside and outside of the PEC object would be zero and $\boldsymbol{E}, \boldsymbol{H}$, respectively, as shown in the original problem.

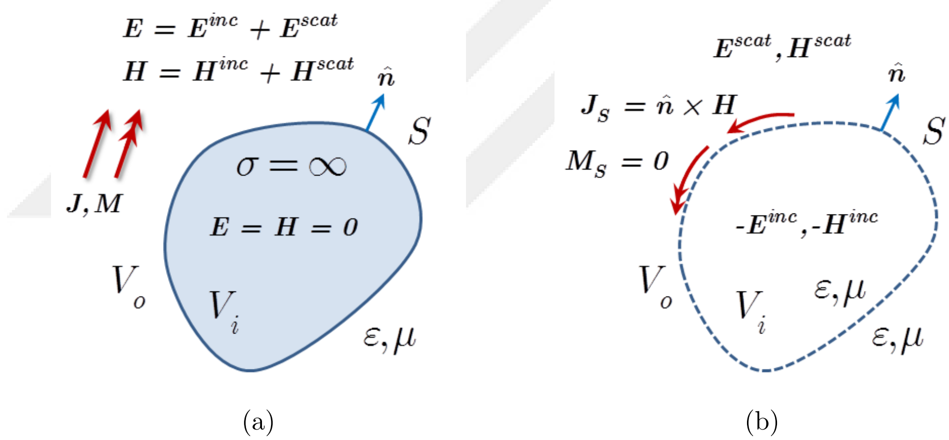


Figure 2.1: Physical Equivalence. (a) Original problem consisting of a PEC object illuminated by incident fields $\boldsymbol{E}^{inc}$ and $\boldsymbol{H}^{inc}$. The total fields outside the object are a superposition of the incident fields plus the scattered fields $\boldsymbol{E}^{scat}$ and $\boldsymbol{H}^{scat}$. (b) Equivalent problem without the object. Introduced surface currents can radiate in a homogeneous space and create the scattered fields outside the object.

It is worth to mention that even though eliminating the object and introducing $\boldsymbol{J}_S$ reduce the original problem into a simple radiation problem, the induced current $\boldsymbol{J}_S$ depends on both of the fields, i.e., the known incident field and the unknown scattered field. To overcome this issue, we use (2.39) and (2.40) to establish a set of integral equations (IEs) that only need the incident fields to be known on the surface of the object.

2.3.2 Volume Equivalence Principle

For the sake of convenience, the volume equivalence principle will be addressed at the second part of this dissertation when the VIEs are discussed.

2.4 Surface Integral Equations

Radiation and scattering problems are among the most important EM problems to be solved analytically or numerically in the literature or engineering applications. For the radiation problems, one needs to integrate a known electric current $\boldsymbol{J}$ using (2.29) and (2.30), or their more general versions given by (2.27) and (2.28) if the magnetic $\boldsymbol{M}$ exists, to obtain the needed near or far fields. But regarding the scattering problems, the currents used in these equations are unknown. This means that to solve a scattering problem, one has to carry out two tasks: (a) Solve a proper integral equation to find out the unknown currents induced by some incident fields $\boldsymbol{E}^{inc}$ and $\boldsymbol{H}^{inc}$. (b) Integrate the computed induced currents to obtain the scattered fields quite similar to radiation problems.

In this section, based on the physical equivalence theorem, which was recently discussed, we derive two well-known integral equations for PEC objects. These are electric-field integral equation (EFIE) and magnetic-field integral equation (MFIE). Note that a convex combination of these two, known as the combined-field integral equation (CFIE), is also available. All of these SIEs are extensively discussed in [35].

2.4.1 EFIE

When there is an induced current on the surface of a PEC object which is illuminated by an incident field, it will radiate and create a secondary field. This field, which is commonly known as the scattered field, can be obtained from the induced current $\boldsymbol{J}$ using (2.22) as

$$\boldsymbol{E}^{scat}(\boldsymbol{r}) = ik\eta \int d\boldsymbol{r'} \left[\boldsymbol{J}(\boldsymbol{r'}) + \frac{1}{k^2}\nabla' \cdot \boldsymbol{J}(\boldsymbol{r'})\nabla\right] g(\boldsymbol{r}, \boldsymbol{r'}). \tag{2.41}$$

To eliminate $\boldsymbol{E}^{scat}$ in this equation, it is enough to enforce the BC regarding the tangential component of the electric field, i.e,

$$\hat{\boldsymbol{n}} \times \boldsymbol{E}^{scat}(\boldsymbol{r}) = -\hat{\boldsymbol{n}} \times \boldsymbol{E}^{inc}(\boldsymbol{r}), \tag{2.42}$$

where $\boldsymbol{r} \in S$ and $\hat{\boldsymbol{n}}$ is the outward normal unit vector on the surface of the object as shown in Fig. 2.1. By rearranging the equation, we obtain N-EFIE formulation as

$$\eta\, \hat{\boldsymbol{n}} \times \mathcal{T}\{\boldsymbol{J}\}(\boldsymbol{r}) = -\hat{\boldsymbol{n}} \times \boldsymbol{E}^{inc}(\boldsymbol{r}), \tag{2.43}$$

and by performing a cross product with $-\hat{\boldsymbol{n}}$ (from the left), we obtain T-EFIE as

$$-\eta\, \hat{\boldsymbol{n}} \times \hat{\boldsymbol{n}} \times \mathcal{T}\{\boldsymbol{J}\}(\boldsymbol{r}) = \hat{\boldsymbol{n}} \times \hat{\boldsymbol{n}} \times \boldsymbol{E}^{inc}(\boldsymbol{r}), \tag{2.44}$$

which is commonly known as EFIE for perfectly conducting objects. Such sort of IEs, where the unknown only appears inside the integral, are known as Fredholm integral equations of the first kind. EFIE may be applied for all types of open and closed objects due to the fact that its derivation does not need any constrain to be imposed on the shape of the object.

2.4.2 MFIE

By following the same method, i.e., enforcing the BC for the tangential component of the magnetic field on the surface of a PEC object, a similar IE can be derived. Using physical equivalent of (2.40), one can write

$$\boldsymbol{J} = \hat{\boldsymbol{n}} \times (\boldsymbol{H}^{scat} + \boldsymbol{H}^{inc}), \tag{2.45}$$

where $\boldsymbol{J}_S$ is the induced current on the surface of the object and the observation point $\boldsymbol{r}$ approaches the surface from outside of the object. Due to the fact that the scattered magnetic field $\boldsymbol{H}^{scat}$ can be computed via (2.30) by invoking the definition of $\mathcal{K}$ operator, (2.45) can be written as

$$\boldsymbol{J}(\boldsymbol{r}) - \lim_{\boldsymbol{r}\to S^+} \left\{ \hat{\boldsymbol{n}} \times \int_S d\boldsymbol{r'} \boldsymbol{J}(\boldsymbol{r'}) \times \nabla' g(\boldsymbol{r}, \boldsymbol{r'}) \right\} = \hat{\boldsymbol{n}} \times \boldsymbol{H}^{inc}(\boldsymbol{r}) \tag{2.46}$$

where S^+ implies approaching the surface from the outside. To evaluate the integral inside the bracket, we can separate it into two distinct parts, i.e.,

$$\begin{aligned} \lim_{\boldsymbol{r}\to S^+} \int_S d\boldsymbol{r'} \boldsymbol{J}(\boldsymbol{r'}) \times \nabla' g(\boldsymbol{r}, \boldsymbol{r'}) = \lim_{\delta S\to 0} & \int_{S-\delta S} d\boldsymbol{r'} \boldsymbol{J}(\boldsymbol{r'}) \times \nabla' g(\boldsymbol{r}, \boldsymbol{r'}) \\ & + \lim_{\delta S\to 0} \int_{\delta S} d\boldsymbol{r'} \boldsymbol{J}(\boldsymbol{r'}) \times \nabla' g(\boldsymbol{r}, \boldsymbol{r'}) \end{aligned} \tag{2.47}$$

where δS is a tiny circular region on S in the vicinity of the observation point $\boldsymbol{r}$, as shown in Fig. 2.2. The first and second terms on the right hand side are known as the principle value and the limit value of the above integral, respectively. It can be shown [60] that the limit value depends on Ω_i, which is defined as the internal solid angle of the surface at the observation point. Based on this definition, the limit value can be written as

$$\lim_{\delta S\to 0} \int_{\delta S} d\boldsymbol{r'} \boldsymbol{J}(\boldsymbol{r'}) \times \nabla' g(\boldsymbol{r}, \boldsymbol{r'}) = \frac{\Omega_i}{4\pi} \boldsymbol{J}(\boldsymbol{r}) \times \hat{\boldsymbol{n}}. \tag{2.48}$$

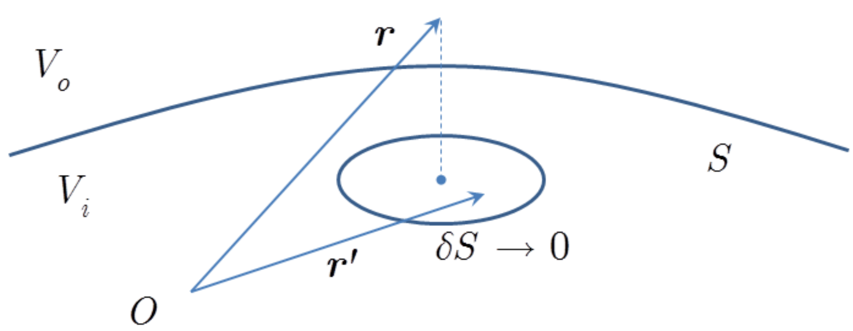


Figure 2.2: The observation point $\boldsymbol{r}$ approaches to a tiny circular portion of the smooth surface S from the outside.

Using a “PV” sign to imply the principal value, (2.46) can be written in a more compact form as

$$\begin{aligned}\boldsymbol{J}(\boldsymbol{r}) - \hat{\boldsymbol{n}} \times \mathcal{K}_{\mathrm{PV}}\{\boldsymbol{J}\}(\boldsymbol{r}) - \hat{\boldsymbol{n}} \times \frac{\Omega_i}{4\pi}\boldsymbol{J}(\boldsymbol{r}) \times \hat{\boldsymbol{n}} &= \hat{\boldsymbol{n}} \times \boldsymbol{H}^{inc}(\boldsymbol{r}) \\ \frac{\Omega_o}{4\pi}\mathcal{I}\{\boldsymbol{J}\}(\boldsymbol{r}) - \hat{\boldsymbol{n}} \times \mathcal{K}_{\mathrm{PV}}\{\boldsymbol{J}\}(\boldsymbol{r}) &= \hat{\boldsymbol{n}} \times \boldsymbol{H}^{inc}(\boldsymbol{r}),\end{aligned} \tag{2.49}$$

where $\Omega_o = 4\pi - \Omega_i$ represents the external solid angle of the surface at the observation point. The IE given in (2.49) is known as the N-MFIE formulation. Considering the fact that the exterior solid angle is simply 2π for smooth surfaces, which is the case when we use planar triangles to discretize the surface of the object [35], the above general form of the N-MFIE may be reduced to

$$\frac{1}{2}\mathcal{I}\{\boldsymbol{J}\}(\boldsymbol{r}) - \hat{\boldsymbol{n}} \times \mathcal{K}_{\mathrm{PV}}\{\boldsymbol{J}\}(\boldsymbol{r}) = \hat{\boldsymbol{n}} \times \boldsymbol{H}^{inc}(\boldsymbol{r}). \tag{2.50}$$

By simply performing a cross product with $\hat{\boldsymbol{n}}$ (from the left), it is also possible to derive the T-MFIE formulation as

$$\frac{1}{2}\mathcal{I}^{\times n}\{\boldsymbol{J}\}(\boldsymbol{r}) - \hat{\boldsymbol{n}} \times \hat{\boldsymbol{n}} \times \mathcal{K}_{\mathrm{PV}}\{\boldsymbol{J}\}(\boldsymbol{r}) = \hat{\boldsymbol{n}} \times \hat{\boldsymbol{n}} \times \boldsymbol{H}^{inc}(\boldsymbol{r}). \tag{2.51}$$

Once again, after solving MFIE for unknown surface currents, (2.29) and (2.30) can be used to compute the EM fields. It is worth to mention that MFIE is a Fredholm IE of the second kind, which means that the unknown $\boldsymbol{J}$ appears outside and inside of the integral. From the computational point of view, this type of IEs seem to show better convergence when solved by iterative solvers rather than the first type, but their accuracy are generally worse than the first type [1].

It should be mentioned that MFIE can only be used for closed surfaces due to the fact that its derivation is based on a limiting approach from outside (or even inside) of the object. In this dissertation, we implement a broadband solver. This simply means that, we need a formulation that is stable at low frequencies. Fortunately, N-MFIE is capable of handling this issue [34] and hence, we use it through the first part of this dissertation.

2.5 Method of Moments

Due to the fact that both of the differentiation and the integration are linear operators, any IE can be treated as linear operator $\mathcal{L}$ acting on an unknown function $\boldsymbol{f}(\boldsymbol{r})$, where the result is another function $\boldsymbol{g}(\boldsymbol{r})$, and they can be written in a compact form as

$$\mathcal{L}\{\boldsymbol{f}(\boldsymbol{r})\} = \boldsymbol{g}(\boldsymbol{r}). \tag{2.52}$$

In EM related problems, $\boldsymbol{f}$ may be either a surface or volume current density ($\boldsymbol{J}$ or $\boldsymbol{M}$), or it may be a scalar function when (2.52) reduces from a 3-D to a 1-D problem. On the other side of this equation, we have a known function $\boldsymbol{g}$, which is usually an excitation due to some external (or internal) EM sources.

In this method, we try to expand the unknown function $\boldsymbol{f}$ in a series of locally defined linearly independent basis functions $\boldsymbol{b}_n(\boldsymbol{r})$ as

$$\boldsymbol{f}(\boldsymbol{r}) \approx \sum_{n=1}^{N} a_n \boldsymbol{b}_n(\boldsymbol{r}), \tag{2.53}$$

where $a_n, n = 1 \cdots N$ are the unknown coefficients that should be found at the end of the solution process. These basis functions have to be chosen in such a way that they can approximate the $\boldsymbol{f}$ function appropriately. Obviously there would be a residual error in this approximation, but the goal is to make this error controllable. By inserting (2.53) into (2.52) and using the fact that $\mathcal{L}$ is a linear operator, we can write

$$\sum_{n=1}^{N} a_n \mathcal{L}\{\boldsymbol{b}_n(\boldsymbol{r})\} = \boldsymbol{g}(\boldsymbol{r}). \tag{2.54}$$

Equation (2.54) consists of N unknown coefficients. To solve this equation, one can use another set of N properly chosen testing functions $\boldsymbol{t}_m(\boldsymbol{r})$ to test both sides of it. To perform this task, we need to define and use an inner product defined as

$$< \boldsymbol{f}_1(\boldsymbol{r}), \boldsymbol{f}_2(\boldsymbol{r}) >= \int d\boldsymbol{r} \boldsymbol{f}_1(\boldsymbol{r}) \cdot \boldsymbol{f}_2(\boldsymbol{r}), \tag{2.55}$$

where the integration may be over a line, on a surface, or inside a volume, depending on the problem of interest. By testing both sides of (2.54), and interchanging the order of summation and integration we may arrive at

$$\sum_{n=1}^{N} a_n \int d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \mathcal{L}\{\boldsymbol{b}_n(\boldsymbol{r})\} = \int d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{g}(\boldsymbol{r}). \tag{2.56}$$

Equation (2.56) still consists of N unknowns. To solve it properly, one needs to use M independent testing functions, which leads us to a matrix equation in the form of

$$\sum_{n=1}^{N} a_n Z_{mn} = v_m, \quad m = 1, 2, \ldots, M, \tag{2.57}$$

where the matrix elements are given by

$$Z_{mn} = \int d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \mathcal{L}\{\boldsymbol{b}_n(\boldsymbol{r})\} \tag{2.58}$$

and the elements of the right-hand side vector are given by

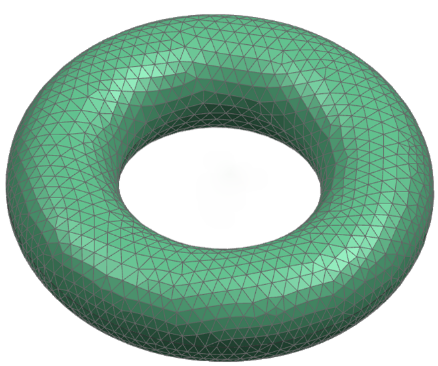
$$v_m = \int d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{g}(\boldsymbol{r}). \tag{2.59}$$

Usually $\bar{\boldsymbol{Z}}$ and $\boldsymbol{v}$ are called the impedance matrix and the excitation vector, respectively. Utilizing such an approach to transform an IE into a linear matrix equation is referred to as the MoM. For EM problems it is very common to utilize the Galerkin's scheme, which means that the testing functions can be chosen to be the same as the basis functions. Using this scheme, the final impedance matrix is a square one, which can be solved by direct or iterative methods.

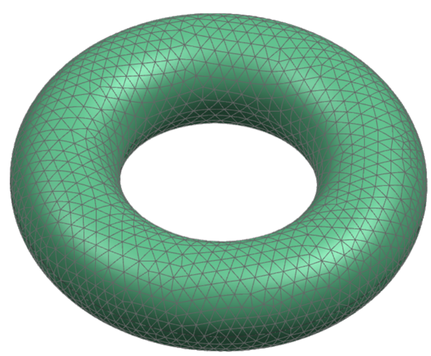
Finally it should be noted that for a problem with N unknowns, the square impedance matrix generated by the MoM has N^2 elements, each of them to be calculated separately. Both the CPU time (required to generate the matrix) and the required memory (to store the matrix) have $\mathcal{O}(N^2)$ complexity. Apparently, the resultant matrix equation has to be solved either directly or iteratively. By using a direct method, like Gaussian elimination, the computational complexity reaches to $\mathcal{O}(N^3)$, while using iterative solvers (via MVM) this complexity may be reduced to $\mathcal{O}(qN^2)$ operations, where q denotes the total number of iterations. Consequently, if $q \approx \mathcal{O}(N)$, the total computational complexity may be close to $\mathcal{O}(N^3)$, which becomes increasingly inefficient for large values of N.

2.5.1 Geometry Discretization

In order to use MOM, one needs to discretize the geometry as well as the SIE formulations. This means that, depending on the type of the basis function to be used, the surface of the object has to be meshed. For example in Fig. 2.3(a), planar (linear) triangles are used to mesh the surface of a typical toroid. Low-order basis functions are proper choices for this type of discretization. It is also possible to use parabolic triangles to mesh the same surface as shown in Fig. 2.3(b). Higher-order basis functions can be used for this type of discretizations.



(a)



(b)

Figure 2.3: Discretization of a typical toroid using two common types of meshes. (a) Using planar (linear) triangles. (b) Using parabolic triangles.

To have a reasonable efficiency and also a small error, the size of the mesh should be kept smaller than $\lambda/10$. Although a smaller mesh seems to be better from the accuracy point of view, it increases the number of unknowns N, which in turn, increases the solution time.

It should be noted that there are two sources of error regarding geometry discretization. The first one is due to the approximation of the surface of an object with planar (linear) triangles. This can be effectively reduced by using parabolic triangles along with higher-order basis functions. The second error, that always exists, is due to the approximation of a continuous surface current utilizing discrete functions. As an expensive remedy, one can use smaller triangles to reduce this error.

2.5.2 Triangular Basis Functions

In the first part of this dissertation (for SIEs), we use linearly varying RWG vector functions, which are defined over planar triangular domains. They are commonly used in MOM applications as basis and testing functions due to their useful properties. An RWG function defined on a couple of triangles with a common edge of length l_n is shown in Fig. 2.4, which is associated with the nth edge of the mesh. The mathematical definition for such a vector basis function is given by

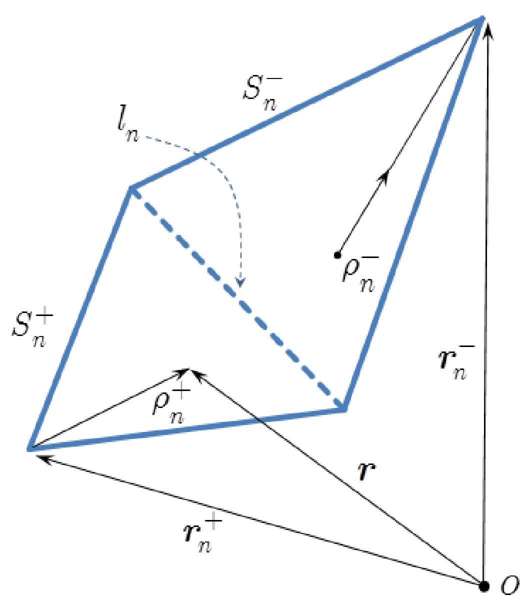


Figure 2.4: An RWG function defined on a couple of planar (linear) triangles with a common edge of length l_n.

$$
\boldsymbol{b}_n^{RWG}(\boldsymbol{r}) = \begin{cases} \dfrac{l_n}{2A_n^+}(\boldsymbol{r} - \boldsymbol{r}_n^+), & \boldsymbol{r} \in S_n^+ \\ \dfrac{l_n}{2A_n^-}(\boldsymbol{r}_n^- - \boldsymbol{r}), & \boldsymbol{r} \in S_n^- \\ 0, & \boldsymbol{r} \notin S_n, \end{cases} \tag{2.60}
$$

where A_n^+ and A_n^- are the areas of S_n^+ and S_n^- triangles, respectively. Some useful properties of an RWG function can be summarized as follows.

1. Inside each triangle, a vectorial summation of three linearly independent RWG functions, one associated with each edge, can represent any constant vector in any desired direction.

2. An RWG function is tangential to both side edges and has only a normal component on the common edge.

3. The component of $\boldsymbol{b}_n(\boldsymbol{r})$ normal to the nth edge is constant and continuous across the edge because the normal component of $\boldsymbol{\rho}_n^{\pm}$ along the common edge is just the height of $S_n^{\pm}$ with the common edge n as the base, and its value is equal to $2A_n^{\pm}/l_n$. This is crucial in order to normalize $\boldsymbol{b}_n(\boldsymbol{r})$ in (2.60) and force the surface current passing through out the common edge to be continuous.

4. The charge density represented by the nth RWG function can be derived as

$$
\nabla \cdot \boldsymbol{b}_n^{RWG}(\boldsymbol{r}) = \begin{cases} \dfrac{l_n}{A_n^+}, & \boldsymbol{r} \in S_n^+ \\ -\dfrac{l_n}{A_n^-}, & \boldsymbol{r} \in S_n^- \\ 0, & \boldsymbol{r} \notin S_n. \end{cases} \tag{2.61}
$$

   Note that the charge density is constant within each triangle and also the total charge density associated with a pair of adjacent triangles remains zero. Due to the fact that no charge accumulation occurs on any edge, RWG basis functions are known to belong to the category of divergence-conforming basis functions.

   It should be emphasized that other types of basis functions, i.e., curl-conforming basis functions as well as higher-order basis functions [61] are also common in computational EM problems.

2.5.3 Operator Discretization

In order to solve any of the aforementioned formulations numerically, one has to discretize the associated operators, and calculate the elements of the impedance matrix, as well as the elements of the RHS of the matrix equation obtained from MoM. Due to the fact that in the proposed IL-MLFMA we only deal with MFIE (≡N-MFIE) formulation for SIEs, it is enough to calculate only the normally-tested $\mathcal{K}_{PV}$ and tangentially-tested $\mathcal{I}$ operators. These operators are discretized as

$$\bar{\boldsymbol{K}}_{PV}^{N}[m,n] = \Big\langle \boldsymbol{t}_m(\boldsymbol{r}),\, \hat{\boldsymbol{n}} \times \mathcal{K}_{PV}\{\boldsymbol{b}_n\}(\boldsymbol{r}) \Big\rangle \tag{2.62}$$

$$\begin{aligned} \bar{\boldsymbol{I}}^{T}[m,n] &= \Big\langle \boldsymbol{t}_m(\boldsymbol{r}),\, \mathcal{I}\{\boldsymbol{b}_n\}(\boldsymbol{r}) \Big\rangle \\ &= \Big\langle \boldsymbol{t}_m(\boldsymbol{r}),\, \boldsymbol{b}_n(\boldsymbol{r}) \Big\rangle, \end{aligned} \tag{2.63}$$

where $< \cdot, \cdot >$ is defined in (2.55), and the integration is on the support of the testing function S_m. Putting all together, the corresponding equations for Z_{mn} and v_m in (2.58) and (2.59) can be written as [35]

$$\bar{\boldsymbol{Z}}^{\mathrm{MFIE}}[m,n] = -\frac{1}{2}\bar{\boldsymbol{I}}^{T}[m,n] + \bar{\boldsymbol{K}}_{PV}^{N}[m,n] \tag{2.64}$$

$$\boldsymbol{v}_{\boldsymbol{H}}^{N}[m] = \Big\langle \boldsymbol{t}_m(\boldsymbol{r}),\, -\hat{\boldsymbol{n}} \times \boldsymbol{H}^{inc}(\boldsymbol{r}) \Big\rangle. \tag{2.65}$$

2.5.4 Elements of the Impedance Matrix (N-MFIE)

By invoking the Galerkin's scheme and deploying the divergence-conforming RWG functions, and substituting (2.62) and (2.63) into (2.64), the explicit expressions for the impedance matrix elements of (2.64) can be obtained as [35]

$$\bar{\boldsymbol{Z}}^{\text{MFIE}}[m,n] = -\frac{1}{2}\int_{S_m} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{b}_n(\boldsymbol{r}) + \int_{S_m} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \hat{\boldsymbol{n}} \times \int_{PV,S_n} d\boldsymbol{r'} \boldsymbol{b}_n(\boldsymbol{r'}) \times \nabla' g_0(\boldsymbol{r},\boldsymbol{r'}). \tag{2.66}$$

Using the definition of an RWG function given in (2.60), we can write (2.66) as

$$\bar{\boldsymbol{Z}}^{\text{MFIE}}[m,n,a,b] = -\frac{\mathcal{C}_{ma,nb}}{2}\delta_{ma,nb}\int_{S_{ma}} d\boldsymbol{r}\,(\boldsymbol{r}-\boldsymbol{r}_{ma})\cdot(\boldsymbol{r}-\boldsymbol{r}_{nb}) + \mathcal{C}_{ma,nb}\int_{S_{ma}} d\boldsymbol{r}\,(\boldsymbol{r}-\boldsymbol{r}_{ma})\cdot\hat{\boldsymbol{n}}\times\int_{PV,S_{nb}} d\boldsymbol{r'}(\boldsymbol{r'}-\boldsymbol{r}_{nb})\times\nabla' g_0(\boldsymbol{r},\boldsymbol{r'}), \tag{2.67}$$

where

$$\mathcal{C}_{ma,nb} = \frac{l_m l_n}{4A_{ma}A_{nb}}\gamma_{ma}\gamma_{nb} \tag{2.68}$$

$$\delta_{m,n} = \begin{cases} 1, & m = n \\ 0, & m \neq n \end{cases} \tag{2.69}$$

and $\gamma_{ma}, \gamma_{nb} = \pm 1$, depending on the orientation of the n^{th} basis and m^{th} testing functions inside the associated triangles. Although it seems to be clear, m and n are used to indicate that the interaction computed in (2.67) is between the m^{th} testing and the n^{th} basis triangles, while a and b denote the orientation of those functions. Also note that the Kronecker delta function used in the first term of the above expression denotes a self-interaction term, and should only be considered in near-field interactions.

To simplify (2.67) one more step, one can take the gradient of the Green's function, and calculate the rightmost integral of (2.67) as

$$I = \int_{PV,S_{nb}} d\boldsymbol{r'}(\boldsymbol{r'} - \boldsymbol{r}_{nb}) \times \nabla' g_0(\boldsymbol{r}, \boldsymbol{r'})$$
$$= \int_{PV,S_{nb}} d\boldsymbol{r'} \left[(\boldsymbol{r'} - \boldsymbol{r}_{nb}) \times (\boldsymbol{r} - \boldsymbol{r'})\right] (1 - ikR)\frac{\exp(ikR)}{4\pi R^3}. \tag{2.70}$$

Using the identity $(\boldsymbol{r'} - \boldsymbol{r}_{nb}) \times (\boldsymbol{r} - \boldsymbol{r'}) = (\boldsymbol{r} - \boldsymbol{r}_{nb}) \times (\boldsymbol{r} - \boldsymbol{r'})$, (2.70) can be written as

$$I = (\boldsymbol{r} - \boldsymbol{r}_{nb}) \times \int_{PV,S_{nb}} d\boldsymbol{r'} \left[(\boldsymbol{r} - \boldsymbol{r'})(1 - ikR)\frac{\exp(ikR)}{4\pi R^3}\right]$$
$$I = (\boldsymbol{r} - \boldsymbol{r}_{nb}) \times \int_{PV,S_{nb}} d\boldsymbol{r'} \nabla' g_0(\boldsymbol{r}, \boldsymbol{r'}), \tag{2.71}$$

and hence, (2.67) becomes

$$\bar{\boldsymbol{Z}}^{\text{MFIE}}[m,n,a,b] = -\frac{\mathcal{C}_{ma,nb}}{2}\delta_{ma,nb} \int_{S_{ma}} d\boldsymbol{r}\,(\boldsymbol{r} - \boldsymbol{r}_{ma}) \cdot (\boldsymbol{r} - \boldsymbol{r}_{nb})$$
$$+\,\mathcal{C}_{ma,nb} \int_{S_{ma}} d\boldsymbol{r}\,(\boldsymbol{r} - \boldsymbol{r}_{ma}) \cdot \hat{\boldsymbol{n}} \times (\boldsymbol{r} - \boldsymbol{r}_{nb}) \times \int_{PV,S_{nb}} d\boldsymbol{r'} \nabla' g_0(\boldsymbol{r}, \boldsymbol{r'}). \tag{2.72}$$

Finally, by using the vector identity $\boldsymbol{A} \cdot \boldsymbol{B} \times \boldsymbol{C} = (\boldsymbol{A} \times \boldsymbol{B}) \cdot \boldsymbol{C}$ we can rewrite (2.72) as [35]

$$\bar{\boldsymbol{Z}}^{\text{MFIE}}[m,n,a,b] = -\,\mathcal{C}_{ma,nb} \left\{ \frac{\delta_{ma,nb}}{2} \int_{S_{ma}} d\boldsymbol{r}\,(\boldsymbol{r} - \boldsymbol{r}_{ma}) \cdot (\boldsymbol{r} - \boldsymbol{r}_{nb}) \right.$$
$$\left. - \int_{S_{ma}} d\boldsymbol{r}\,\left[\,(\boldsymbol{r} - \boldsymbol{r}_{ma}) \times \hat{\boldsymbol{n}}\,\right] \cdot \left[(\boldsymbol{r} - \boldsymbol{r}_{nb}) \times \int_{PV,S_{nb}} d\boldsymbol{r'}\,\nabla' g_0(\boldsymbol{r}, \boldsymbol{r'})\right] \right\}. \tag{2.73}$$

2.5.5 Excitation

To illuminate the scatterer, we can use a plane-wave excitation, i.e.,

$$\boldsymbol{H}^{inc}(\boldsymbol{r}) = \frac{1}{\eta}\,\hat{\boldsymbol{k}} \times \boldsymbol{E}_0\, e^{ik\hat{\boldsymbol{k}}\cdot\boldsymbol{r}}, \tag{2.74}$$

where $\boldsymbol{E}_0$ is a complex vector and it should satisfy $\boldsymbol{E}_0 \cdot \hat{\boldsymbol{k}} = 0$. Inserting (2.74) into (2.65) gives us the elements of the RHS vector of the MoM system given by

$$\boldsymbol{v}_{\boldsymbol{H}}^{N}[m] = \frac{1}{\eta} \int_{S_m} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \left(\hat{\boldsymbol{n}} \times \hat{\boldsymbol{k}} \times \boldsymbol{E}_0\, e^{ik\hat{\boldsymbol{k}}\cdot\boldsymbol{r}} \right). \tag{2.75}$$

2.6 Evaluation of the Integrals

In the context of the MoM, there are two types of integrals that should be evaluated accurately: (a) Those integrals that can be handled by using numerical integration (quadrature) methods. (b) Those integrals with singular kernels that can be evaluated using some special singularity-extraction techniques.

For example in the N-MFIE formulation given by (2.73), the first term (the self-interaction integral) can be easily evaluated by a second-order quadrature. But the inner integral of the second term needs more careful evaluation due to the singular behavior of its integrand when $\boldsymbol{r}$ approaches to $\boldsymbol{r'}$. This happens when the source and testing triangles are overlapping or they are at close proximity of each other.

2.6.1 Numerical Integration

There are many methods in order to numerically evaluate an integral over triangular domains. The most commonly used method is the Gaussian-Legendre quadrature with symmetric rules presented in [52]. In this method, it is possible to choose the optimum abscissas to integrate a function $f(x)$ as

$$\int_{-1}^{1} f(x)dx = \sum_{i=1}^{N} w_i f(x_i) \tag{2.76}$$

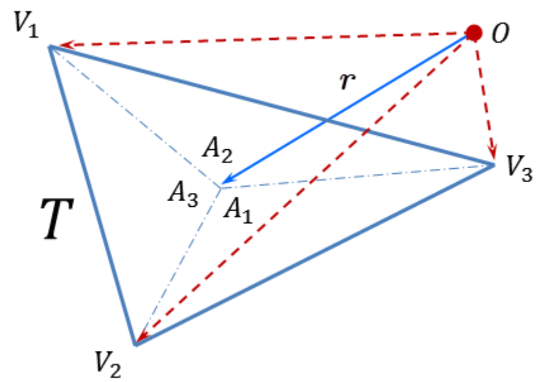


Figure 2.5: Definition of simplex coordinates inside a triangle T.

where $x_i, i = 1 \ldots N$ are the roots of $\mathrm{P}_n(x)$, i.e., the Legendre polynomial of order n, and the weights w_i are [62]

$$w_i = \frac{2}{(1 - x_i^2)\left[\mathrm{P}'_N(x_i)\right]^2}. \tag{2.77}$$

Apparently, to use the rule, a proper change of variable has to be used to map the interval from $[-1, 1]$ to $[a, b]$.

Note that these rules cannot be directly used to integrate a given vectorial function over triangular domains. To carry out such sort of integrals, one needs to use *simplex coordinates*, which enables us to easily find the coordinates of an arbitrary point inside a triangle based on the coordinates of its vertices. To clarify the concept, we briefly consider a triangle T as shown in Fig. 2.5 by defining three position vectors $\boldsymbol{V}_1$, $\boldsymbol{V}_2$, and $\boldsymbol{V}_3$ as its vertices. An arbitrary point $\boldsymbol{r}$ inside T can be found as a weighted sum of those three vectors associated with the vertices via

$$\boldsymbol{r} = \alpha\boldsymbol{V}_1 + \beta\boldsymbol{V}_2 + \gamma\boldsymbol{V}_3, \tag{2.78}$$

where α, β, and γ are the simplex coordinates defined by

$$\alpha = \frac{A_1}{A}, \qquad \beta = \frac{A_2}{A}, \qquad \gamma = \frac{A_3}{A}, \tag{2.79}$$

and A is the total area of the triangle consisting of A_1, A_2, and A_3 as shown in Fig. 2.5. Apparently, $A_1 + A_2 + A_3 = A$ and hence, $\alpha + \beta + \gamma = 1$. Now by extending the one-dimensional quadrature rule of (2.76) over a triangle (and later on inside a tetrahedron), one can write

$$\int_T f(\boldsymbol{r})dx = A\sum_{i=1}^{N} w_i f(\boldsymbol{r}_i), \tag{2.80}$$

where $\boldsymbol{r}_i$ is calculated via tabulated values for α, β, and γ using (2.78).

2.6.2 Analytic Integration

To evaluate the singular integrand of the gradient of the Green's function in (2.73), i.e.,

$$\boldsymbol{I} = \int_{PV,S_{nb}} d\boldsymbol{r'}\,\nabla' g_0(\boldsymbol{r},\boldsymbol{r'}), \tag{2.81}$$

some singularity extraction methods are needed. Due to the fact that $\boldsymbol{I}$ is infinite when the testing point is on the edge of the source triangle S_n, the interaction of the touching triangles can be computed accurately by using those observation points that are strictly inside the testing triangle. Now, in order to remove the singularity from numerical integration [51], one can write $\boldsymbol{I}$ as

$$\begin{aligned}\boldsymbol{I} = &\frac{1}{4\pi}\int_{S_{nb}} d\boldsymbol{r'}\,\nabla'\left(\frac{\exp(ikR) - 1 + 0.5k^2R^2}{R}\right)\\ &+ \frac{1}{4\pi}\int_{S_{nb}} d\boldsymbol{r'}\,\nabla'\left(\frac{1}{R}\right) - \frac{k^2}{8\pi}\int_{S_{nb}} d\boldsymbol{r'}\,\nabla' R.\end{aligned} \tag{2.82}$$

The expression in the parentheses in the first integral has two continuous derivatives with respect to R and hence, it can be evaluated accurately using Gaussian quadrature. To evaluate two other integrals we prefer to use the analytical methods elaborated in [49] and [50], which are briefly repeated here for convenience. It should be noted that in these methods, the source triangle S_{nb} is assumed to be located on the x-y plane as shown in Fig. 2.6 and hence, a proper geometrical manipulation may be needed for an arbitrary triangle in 3-D space [35].

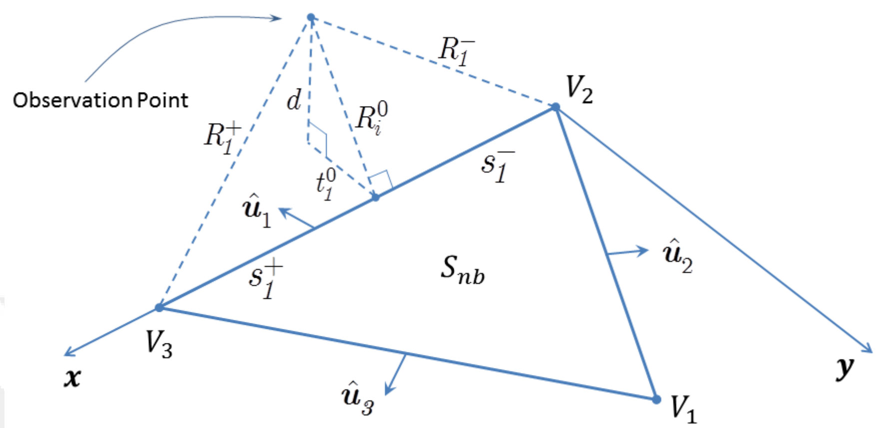


Figure 2.6: Various geometrical parameters regarding singularity extraction [35].

$$\int_{S_{nb}} d\boldsymbol{r'}\,\nabla'\left(\frac{1}{R}\right) = \sum_{i=1}^{3} \hat{\boldsymbol{u}}_i f_{2,i} + \hat{\boldsymbol{z}}\frac{z}{|z|}\sum_{i=1}^{3}\beta_i \tag{2.83}$$

$$\int_{S_{nb}} d\boldsymbol{r'}\,\nabla' R = \frac{1}{2}\sum_{i=1}^{3} \hat{\boldsymbol{u}}_i f_{3,i} + \hat{\boldsymbol{z}}z|z|\sum_{i=1}^{3}\beta_i - \hat{\boldsymbol{z}}z\sum_{i=1}^{3} t_i^0 f_{2,i}, \tag{2.84}$$

where

$$\beta_i = \tan^{-1}\frac{t_i^0 s_i^+}{(R_i^0)^2 + |z|R_i^+} - \tan^{-1}\frac{t_i^0 s_i^-}{(R_i^0)^2 + |z|R_i^-} \tag{2.85}$$

$$f_{2,i} = \ln\left(\frac{R_i^+ + s_i^+}{R_i^- + s_i^-}\right) \tag{2.86}$$

$$f_{3,i} = s_i^+ R_i^+ - s_i^- R_i^- + (R_i^0)^2 f_{2,i}, \tag{2.87}$$

The remaining geometrical parameters depend on each edge ($i = 1, 2, 3$) and based on Fig. 2.6 they can be listed as

- $\hat{\boldsymbol{u}}_i (i = 1, 2, 3)$ denotes the outward unit vector on the plane of the triangle that is prependicular to the i^{th} edge.
- R_i^+ and R_i^- denote the distances between the observation point and the end points of the i^{th} edge. The plus and minus signs are determined based on the right-hand rule regarding the outward normal unit vector of the triangles (here $V_1 \to V_2 \to V_3 \to V_1$).

- R_i^0 is the prependicular distance between the observation point and the i^{th} edge (or its extension).

- t_i^0 is the prependicular distance from the projection of the observation point on the plane of the triangle toward the i^{th} edge (in Fig. 2.6 t_1^0 is shown). Its sign is determined based on its alignment with $\hat{\boldsymbol{u}}$. Obviously $R_i^0 = \sqrt{z^2 + |t_i^0|^2}$.

- A point on the i^{th} edge, where its distance from the observation point is R_i^0, breaks the edge into two segments. s_i^+ and s_i^- are the algebric distances from this point toward the plus and minus end points of the edge via the right-hand rule.

Chapter 3

Basics of FMM and MLFMA

3.1 Introduction to FMM

We briefly discussed the MoM procedure in the previous chapter. For a given EM problem consisting of N unknowns, its memory cost is $\mathcal{O}(N^2)$ and the computational cost reaches to $\mathcal{O}(N^3)$ when direct solvers are used. By using iterative solvers, the computational cost may be reduced to $\mathcal{O}(qN^2)$, where q is the number of iterations. In this chapter we briefly review the FMM, which is originally based on the MoM, but with a different strategy to calculate the interactions. Based on a special technique used to diagonalize the Green's function, FMM is capable of reducing the memory requirement and the CPU time down to $\mathcal{O}(N^{1.5})$. The multilevel version of FMM or MLFMA is also addressed in this chapter, which can reduce the complexity to $\mathcal{O}(N \log N)$, making it feasible to solve larger real-life problems consisting of millions of unknowns [18].

It should be emphasized that FMM (also MLFMA) is based on fast MVM, and its efficiency strongly depends on the choice of the integral equation to be used, as well as the iterative solver to be deployed.

3.1.1 Matrix-Vector Multiplication

Based on a brief review already discussed in Section 2.5, the MoM procedure ends up with a linear matrix equation as

$$\bar{\boldsymbol{Z}}\,\boldsymbol{a} = \boldsymbol{v}, \tag{3.1}$$

where the system matrix $\bar{\boldsymbol{Z}}$ represents the electromagnetic interactions between the basis and the testing functions, and the left-hand side unknown vector $\boldsymbol{a}$ contains the unknown amplitudes of the basis functions. The result of $\bar{\boldsymbol{Z}}\,\boldsymbol{a}$ as an MVM is a vector, where its rows are the coherent sum of those fields "received" by each testing function due to the fields "radiated" by all basis functions. Considering the fact that the amplitude of the interaction between the basis and testing functions tapers off quickly with distance, we can divide the space around a basis function into two regions: a near-field region where some strong interactions occur, and a far-field region where weaker interactions exist. Such a separation between the near and far regions is illustrated in Fig. 1.1. Taking the red box as the one containing the radiating basis functions, other basis functions within the transparent boxes are considered to be in the near region, and those outside, i.e., light or dark blue boxes, are considered as the far region boxes. Writing the impedance matrix as $\bar{\boldsymbol{Z}} = \bar{\boldsymbol{Z}}^{near} + \bar{\boldsymbol{Z}}^{far}$, where $\bar{\boldsymbol{Z}}^{near}$ is a sparse matrix, MVM can be written as the sum of a near product and a far product yielding

$$\bar{\boldsymbol{Z}}^{near}\boldsymbol{a} + \bar{\boldsymbol{Z}}^{far}\boldsymbol{a} = \boldsymbol{v}. \tag{3.2}$$

In FMM, the MVM is performed in two separate steps: an MVM involving $\bar{\boldsymbol{Z}}^{near}$, where the elements are computed directly as usual and stored explicitly, and an MVM involving $\bar{\boldsymbol{Z}}^{far}$, where the individual elements are not computed or available explicitly. An important reason why FMM is one step ahead of the MoM is originated from the fact that a large number of the matrix elements are located inside $\bar{\boldsymbol{Z}}^{far}$, leaving behind a relatively very small number of matrix elements to be computed and stored for $\bar{\boldsymbol{Z}}^{near}$. Based on this perspective, $\bar{\boldsymbol{Z}}^{near}$ is highly sparse and its computation takes much less time than the original dense impedance matrix. This leads to a considerable amount of saving in the memory consumption and CPU time. On the other hand, the computations of $\bar{\boldsymbol{Z}}^{far}$ elements in each iteration are based on group-wise interactions, which take much shorter time than their equivalent MoM based calculations. It is noteworthy to remind that MLFMA can extend this capability by deploying a higher number of levels.

3.1.2 The Gegenbauer's Addition Theorem

Consider the three dimensional free space Green's function, which can be written as

$$g(\boldsymbol{r}, \boldsymbol{r'}) = \frac{e^{ik|\boldsymbol{r}-\boldsymbol{r'}|}}{4\pi|\boldsymbol{r}-\boldsymbol{r'}|} = \frac{e^{ik|\boldsymbol{R}|}}{4\pi|\boldsymbol{R}|}, \tag{3.3}$$

where $\boldsymbol{r'}$ and $\boldsymbol{r}$ are the source and field points, respectively. By defining a small vector offset $\boldsymbol{v}$ and a larger vector $\boldsymbol{w}$ such that $\boldsymbol{R} = \boldsymbol{r} - \boldsymbol{r'} = \boldsymbol{w} + \boldsymbol{v}$, we can rewrite (3.3) as

$$\frac{e^{ik|\boldsymbol{r}-\boldsymbol{r'}|}}{4\pi|\boldsymbol{r}-\boldsymbol{r'}|} = \frac{e^{ik|\boldsymbol{w}+\boldsymbol{v}|}}{4\pi|\boldsymbol{w}+\boldsymbol{v}|}. \tag{3.4}$$

Assuming that $|\boldsymbol{v}| < |\boldsymbol{w}|$, we may define the *addition theorem* for spherical waves [63], as

$$\frac{e^{ik|\boldsymbol{w}+\boldsymbol{v}|}}{4\pi|\boldsymbol{w}+\boldsymbol{v}|} = \frac{ik}{4\pi}\sum_{l=0}^{\infty}(-1)^l(2l+1)j_l(k|\boldsymbol{v}|)h_l^{(1)}(k|\boldsymbol{w}|)P_l(\hat{\boldsymbol{v}}\cdot\hat{\boldsymbol{w}}), \tag{3.5}$$

where j_l and $h_l^{(1)}$ are the spherical Bessel function and the spherical Hankel function of the first kind, respectively, and $P_l(x)$ is known as the Legendre polynomial of order l [62],[64].

3.1.3 Diagonalization

Looking closely at (3.5), one can see that it is a different way of writing a wave, which is radiating from a given point, as if it were radiating from a different point close to it. As explained in [63], (3.5) can be transformed into a surface integral over the unit sphere via the identity

$$4\pi(i)^l j_l(k|\boldsymbol{v}|)P_l(\hat{\boldsymbol{v}}\cdot\hat{\boldsymbol{w}}) = \oint_s e^{ik\hat{\boldsymbol{k}}\cdot\boldsymbol{v}} P_l(\hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{w}})dS, \tag{3.6}$$

where $\hat{\boldsymbol{k}}$ is the radially oriented unit vector over the sphere. Another important aspect of this identity is due to the fact that, it presents an expansion to express a spherical wave in terms of the plane-waves. Note that the orthogonality relationship among the plane-waves in different directions provides us a complete base in $\hat{\boldsymbol{k}}$-space. Using this identity and another common notation for integration over the unit sphere as $\oint_s dS = \int\int sin(\theta)d\theta d\phi = \int d^2\hat{\boldsymbol{k}}$, one can write the *digonalized* form of the Green's function as

$$\frac{e^{ik|\boldsymbol{w}+\boldsymbol{v}|}}{4\pi|\boldsymbol{w}+\boldsymbol{v}|} = \frac{ik}{(4\pi)^2}\int d^2\hat{\boldsymbol{k}}\; e^{ik\hat{\boldsymbol{k}}\cdot\boldsymbol{v}} \sum_{l=0}^{\infty}(i)^l(2l+1)h_l^{(1)}(k|\boldsymbol{w}|)P_l(\hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{w}}), \tag{3.7}$$

where the orders of the integration and summation have been interchanged. This interchange will be practical if we truncate the summation at some finite order τ. Making this truncation, we can now write (3.7) as

$$\frac{e^{ik|\boldsymbol{w}+\boldsymbol{v}|}}{4\pi|\boldsymbol{w}+\boldsymbol{v}|} \approx \frac{ik}{(4\pi)^2}\int d^2\hat{\boldsymbol{k}}\; \beta(k,\hat{\boldsymbol{k}},\boldsymbol{v})T_\tau(k,\hat{\boldsymbol{k}},\boldsymbol{w}) \tag{3.8}$$

using

$$\beta(k,\hat{\boldsymbol{k}},\boldsymbol{v}) = e^{ik\hat{\boldsymbol{k}}\cdot\boldsymbol{v}} \tag{3.9}$$

$$T_\tau(k,\hat{\boldsymbol{k}},\boldsymbol{w}) = \sum_{l=0}^{\tau}(i)^l(2l+1)h_l^{(1)}(k|\boldsymbol{w}|)P_l(\hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{w}}), \tag{3.10}$$

where $\beta(k,\hat{\boldsymbol{k}},\boldsymbol{v})$ and $T_\tau(k,\hat{\boldsymbol{k}},\boldsymbol{w})$ are defined as the diagonal *shift* and *translation* functions for plane-waves, respectively.

From a practical point of view, we have to consider two important issues, namely, integration over the unit sphere $\int d^2\hat{\boldsymbol{k}}$ and the truncation number τ. As a remedy for the first issue, the integration over the unit sphere in (3.8) is carried out numerically with quadratures in the θ and ϕ dimensions. It is shown in [65] that for a function $f(\theta, \phi)$ of bandwidth L, such kind of integrals can be evaluated optimally using an (L+1) point Gauss-Legendre quadrature rule in θ, and an equally spaced 2(L+1) point quadrature in ϕ. The solution for the second issue comes from a variety of suggestions given by different authors [35] to choose the optimal value for τ using

$$\tau \approx 1.73ka + 2.16(\text{dig})^{\frac{2}{3}}(ka)^{\frac{1}{3}} \tag{3.11}$$

where "dig" is the number of desired accurate digits, and a is the box size containing the basis functions. It is worthy to note that the box size a has to be changed across different levels inside the hierarchical tree structure when the multilevel version of FMM has to be deployed.

3.1.4 Wave Translation

Now we illustrate the addition theorem by calculating the Green's function between a source point $\boldsymbol{r'}$ and an observation point $\boldsymbol{r}$. We first introduce two arbitrary points in space, defined by two position vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, that are very close to the field point $\boldsymbol{r}$ and the source point $\boldsymbol{r'}$, respectively, as shown in Fig. 3.1. Breaking up the vector between the source and observation points as

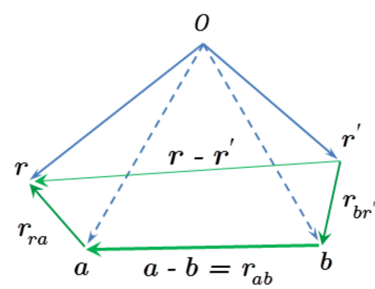


Figure 3.1: Wave translation.

$$\boldsymbol{r} - \boldsymbol{r'} = (\boldsymbol{r} - \boldsymbol{a}) + (\boldsymbol{a} - \boldsymbol{b}) + (\boldsymbol{b} - \boldsymbol{r'}) \tag{3.12}$$

and then

$$\boldsymbol{r} - \boldsymbol{r'} = \boldsymbol{r}_{ra} + \boldsymbol{r}_{ab} + \boldsymbol{r}_{br'}, \tag{3.13}$$

we can rewrite (3.8) using (3.12) and (3.13) as

$$\frac{e^{ik|\boldsymbol{r}-\boldsymbol{r'}|}}{4\pi|\boldsymbol{r}-\boldsymbol{r'}|} \approx \frac{ik}{(4\pi)^2}\int d^2\hat{\boldsymbol{k}}\; e^{ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}_{ra}+\boldsymbol{r}_{br'})} T_\tau(k,\hat{\boldsymbol{k}},\boldsymbol{r}_{ab}) \tag{3.14}$$

$$\approx \frac{ik}{(4\pi)^2}\int d^2\hat{\boldsymbol{k}}\; \beta(k,\hat{\boldsymbol{k}},\boldsymbol{r}_{ra})\beta(k,\hat{\boldsymbol{k}},\boldsymbol{r}_{br'}) T_\tau(k,\hat{\boldsymbol{k}},\boldsymbol{r}_{ab}), \tag{3.15}$$

provided that the assumption considered in the addition theorem is satisfied (that is $\boldsymbol{r}_{ra}+\boldsymbol{r}_{br'} < \boldsymbol{r}_{ab}$). This is an important result due to the fact that the translation function T_τ only depends on $\boldsymbol{r}_{ab}$. Recalling our assumption about the closeness of $\boldsymbol{a}$ to $\boldsymbol{r}$ and $\boldsymbol{b}$ to $\boldsymbol{r'}$, if we move both of $\boldsymbol{r}$ and $\boldsymbol{r'}$ a small distance from their previous locations, the translation function remains intact, and only the phase shifts due to β function change. Therefore, we can compute any interaction between any pair of points near $\boldsymbol{a}$ and $\boldsymbol{b}$ by using the same translation function. Now we write (3.15) as

$$\frac{e^{ik|\boldsymbol{r}-\boldsymbol{r'}|}}{4\pi|\boldsymbol{r}-\boldsymbol{r'}|} \approx \frac{ik}{(4\pi)^2}\int d^2\hat{\boldsymbol{k}}\; R(\hat{\boldsymbol{k}},\boldsymbol{r}) T_\tau(k,\hat{\boldsymbol{k}},\boldsymbol{r}_{ab}) S(\hat{\boldsymbol{k}},\boldsymbol{r'}), \tag{3.16}$$

where, at the source point $\boldsymbol{r'}$, it is beneficial to define the *radiation function* as

$$S(\hat{\boldsymbol{k}},\boldsymbol{r'}) = e^{ik\hat{\boldsymbol{k}}\cdot\boldsymbol{r}_{br'}}, \tag{3.17}$$

and at the field point $\boldsymbol{r}$, the *receiving function* as

$$R(\hat{\boldsymbol{k}},\boldsymbol{r}) = e^{ik\hat{\boldsymbol{k}}\cdot\boldsymbol{r}_{ra}}. \tag{3.18}$$

If we now move $\boldsymbol{r'}$ or $\boldsymbol{r}$ to another close location, only the radiation or receiving function changes. As a result, we can now compute a sum of the Green's functions evaluated at the field point $\boldsymbol{r}$ due to many source points $\boldsymbol{r}'_n$ located close to $\boldsymbol{b}$. Using (3.16), this can be written as

$$\sum_{n=1}^{N}\frac{e^{ik|\boldsymbol{r}-\boldsymbol{r}'_n|}}{4\pi|\boldsymbol{r}-\boldsymbol{r}'_n|} \approx \frac{ik}{(4\pi)^2}\int d^2\hat{\boldsymbol{k}}\; R(\hat{\boldsymbol{k}},\boldsymbol{r}) T_\tau(k,\hat{\boldsymbol{k}},\boldsymbol{r}_{ab}) \sum_{n=1}^{N} S_n(\hat{\boldsymbol{k}},\boldsymbol{r}'_n) \tag{3.19}$$

where the radiation functions are

$$S_n(\hat{\boldsymbol{k}}, \boldsymbol{r}'_n) = e^{ik\hat{\boldsymbol{k}}\cdot\boldsymbol{r}_{br'_n}}. \tag{3.20}$$

The arrangement shown in (3.19) is what enables us to quickly compute a MVM and deploy it within the context of the FMM. The radiation functions for all source points $\boldsymbol{r}'_n$ should be consistently added (*aggregated*) to create a local field at $\boldsymbol{b}$. This field is then *transmitted* via the translation function yielding a local field at $\boldsymbol{a}$. This "incoming" local field is then multiplied with the receiving function at the field point $\boldsymbol{r}$ and integrated over the unit sphere (*disaggregated*) producing the desired sum.

In a practical FMM implementation, the basis and testing functions are distributed into a pre-ordered set of clusters based on their spatial positions. Consequently, the radiation and receiving functions, as well as the inter-cluster translation functions are computed and stored in the memory in order to effectively carry out the aggregation, translation and disaggregation processes.

We have to emphasize that, even though the provided derivations regarding the FMM concepts and related radiation and receiving functions have been derived in the frame of point sources, the introduced concepts remain valid for arbitrary distributed sources as well, apparently with some extra mathematical complications.

3.1.5 Matrix Elements for FMM

In order to derive the matrix elements for FMM, it is more convenient to use the vector identity of $\boldsymbol{A}\cdot\boldsymbol{B}\times\boldsymbol{C} = (\boldsymbol{A}\times\boldsymbol{B})\cdot\boldsymbol{C}$ and rewrite the second term of (2.67) in a more suitable form as

$$\bar{\boldsymbol{Z}}^{\mathrm{MFIE}}[m,n,a,b] = \mathcal{C}_{ma,nb}\int_{S_{ma}} d\boldsymbol{r}\,[(\boldsymbol{r}-\boldsymbol{r}_{ma})\times\hat{\boldsymbol{n}}]\cdot\int_{S_{nb}} d\boldsymbol{r}'(\boldsymbol{r}'-\boldsymbol{r}_{nb})\times\nabla' g_0(\boldsymbol{r},\boldsymbol{r}'). \tag{3.21}$$

Note that since the FMM can only be deployed for far interactions (due to the use of the addition theorem), the self-interaction term is ignored and $\int_{PV,S_{nb}}$ is changed to $\int_{S_{nb}}$.

Now, we consider a pair of RWG functions located at two far boxes as shown in Fig. 3.2. It is clear that

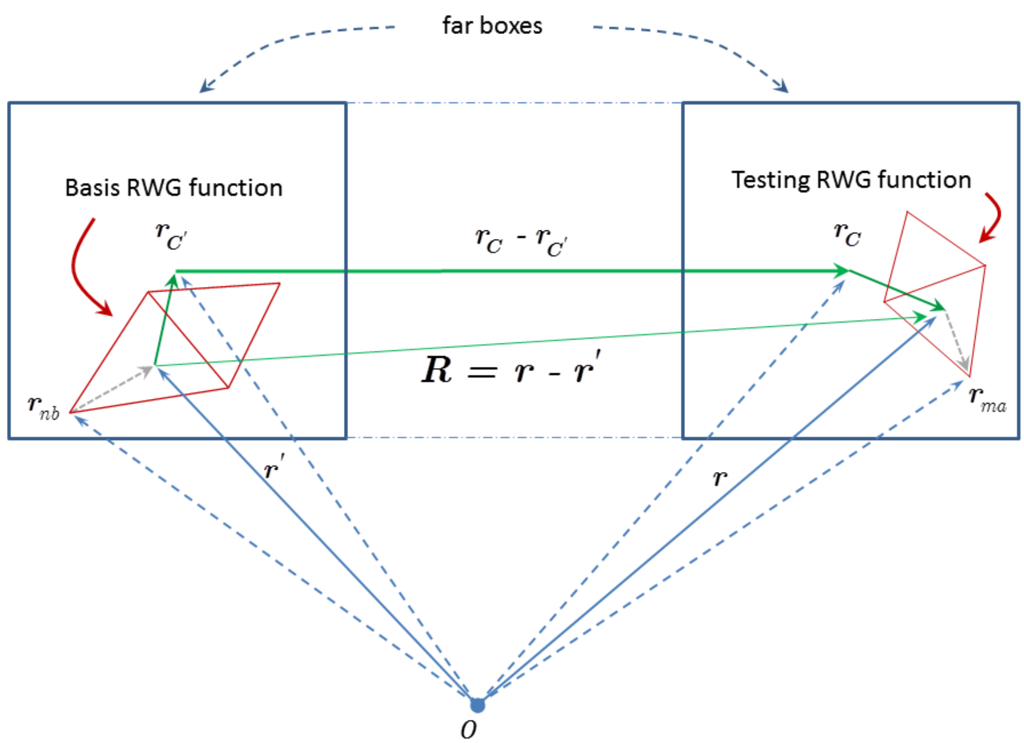


Figure 3.2: Interaction between a pair of RWG functions located at two far boxes in the context of the FMM.

$$\boldsymbol{R} = \boldsymbol{r} - \boldsymbol{r'} = (\boldsymbol{r} - \boldsymbol{r}_C) + (\boldsymbol{r}_C - \boldsymbol{r}_{C'}) + (\boldsymbol{r}_{C'} - \boldsymbol{r'}), \tag{3.22}$$

where $\boldsymbol{r}_C$ and $\boldsymbol{r}_{C'}$ are fixed vectors pointing to the center of two far boxes, where the interacting basis and testing function are located. Recalling the addition theorem given in (3.8)-(3.10) and by using $\boldsymbol{w} = \boldsymbol{r}_C - \boldsymbol{r}_{C'}$ and $\boldsymbol{v} = (\boldsymbol{r} - \boldsymbol{r}_C) +$

$(\boldsymbol{r}_{C'} - \boldsymbol{r'})$ one can write [35]

$$g(\boldsymbol{r},\boldsymbol{r'}) = \frac{ik}{(4\pi)^2}\int d^2\hat{\boldsymbol{k}}\, e^{-ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r'}-\boldsymbol{r}_{C'})}\, T_\tau(k,\hat{\boldsymbol{k}},\boldsymbol{r}_C - \boldsymbol{r}_{C'})\, e^{ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}-\boldsymbol{r}_C)}. \tag{3.23}$$

To take the gradient of the Green's function in the primed coordinates it is enough to consider only the first exponential term and write

$$\nabla' g(\boldsymbol{r},\boldsymbol{r'}) = \frac{ik}{(4\pi)^2}\int d^2\hat{\boldsymbol{k}}\,(-ik\hat{\boldsymbol{k}}) e^{-ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r'}-\boldsymbol{r}_{C'})}\, T_\tau(k,\hat{\boldsymbol{k}},\boldsymbol{r}_C - \boldsymbol{r}_{C'})\, e^{ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}-\boldsymbol{r}_C)}. \tag{3.24}$$

By inserting (3.24) in (3.21) we arrive at

$$\begin{aligned}\bar{\boldsymbol{Z}}^{\mathrm{MFIE}}[m,n,a,b] = \mathcal{C}_{ma,nb}\int_{S_{ma}} d\boldsymbol{r}\,[(\boldsymbol{r}-\boldsymbol{r}_{ma})\times\hat{\boldsymbol{n}}]\cdot\int_{S_{nb}} d\boldsymbol{r'}(\boldsymbol{r'}-\boldsymbol{r}_{nb})\times \\ \frac{ik}{(4\pi)^2}\int d^2\hat{\boldsymbol{k}}\,(-ik\hat{\boldsymbol{k}}) e^{-ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r'}-\boldsymbol{r}_{C'})}\, T_\tau(k,\hat{\boldsymbol{k}},\boldsymbol{r}_C - \boldsymbol{r}_{C'})\, e^{ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}-\boldsymbol{r}_C)}.\end{aligned} \tag{3.25}$$

By exchanging the order of the integrals and using the vector identity $\boldsymbol{A}\cdot\boldsymbol{B}\times\boldsymbol{C} = (\boldsymbol{A}\times\boldsymbol{B})\cdot\boldsymbol{C}$ one more time, one may arrive at

$$\begin{aligned}\bar{\boldsymbol{Z}}^{\mathrm{MFIE}}[m,n,a,b] &= -\mathcal{C}_{ma,nb}\left(\frac{ik}{4\pi}\right)^2\int d^2\hat{\boldsymbol{k}}\int_{S_{ma}} d\boldsymbol{r}\int_{S_{nb}} d\boldsymbol{r'}\,[(\boldsymbol{r}-\boldsymbol{r}_{ma})\times\hat{\boldsymbol{n}}]\cdot \\ &\quad (\boldsymbol{r'}-\boldsymbol{r}_{nb})\times\hat{\boldsymbol{k}}\, e^{-ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r'}-\boldsymbol{r}_{C'})}\, T_\tau(k,\hat{\boldsymbol{k}},\boldsymbol{r}_C - \boldsymbol{r}_{C'})\, e^{ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}-\boldsymbol{r}_C)} \\ &= \mathcal{C}_{ma,nb}\left(\frac{ik}{4\pi}\right)^2\int d^2\hat{\boldsymbol{k}}\int_{S_{ma}} d\boldsymbol{r}\int_{S_{nb}} d\boldsymbol{r'}\,\hat{\boldsymbol{k}}\times[\hat{\boldsymbol{n}}\times(\boldsymbol{r}-\boldsymbol{r}_{ma})]\cdot \\ &\quad (\boldsymbol{r'}-\boldsymbol{r}_{nb})\, e^{-ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r'}-\boldsymbol{r}_{C'})}\, T_\tau(k,\hat{\boldsymbol{k}},\boldsymbol{r}_C - \boldsymbol{r}_{C'})\, e^{ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}-\boldsymbol{r}_C)} \\ &= \mathcal{C}_{ma,nb}\left(\frac{ik}{4\pi}\right)^2\int d^2\hat{\boldsymbol{k}}\left\{\hat{\boldsymbol{k}}\times\int_{S_{ma}} d\boldsymbol{r}\,[\hat{\boldsymbol{n}}\times(\boldsymbol{r}-\boldsymbol{r}_{ma})]\, e^{ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}-\boldsymbol{r}_C)}\right\} \\ &\quad \cdot T_\tau(k,\hat{\boldsymbol{k}},\boldsymbol{r}_C - \boldsymbol{r}_{C'})\left\{\int_{S_{nb}} d\boldsymbol{r'}\,(\boldsymbol{r'}-\boldsymbol{r}_{nb}) e^{-ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r'}-\boldsymbol{r}_{C'})}\right\}.\end{aligned} \tag{3.26}$$

Now by recalling the definition of $\mathcal{C}_{ma,nb}$ from (2.68) and using the relation

$$\hat{\boldsymbol{k}}\times[\hat{\boldsymbol{n}}\times(\boldsymbol{r}-\boldsymbol{r}_{ma})]\cdot(\boldsymbol{r'}-\boldsymbol{r}_{nb}) = \hat{\boldsymbol{k}}\times[\hat{\boldsymbol{n}}\times(\boldsymbol{r}-\boldsymbol{r}_{ma})]\cdot\left[(\bar{\bar{\boldsymbol{I}}}_{3\times3} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}})\cdot(\boldsymbol{r'}-\boldsymbol{r}_{nb})\right] \tag{3.27}$$

we can write the final form of $\bar{\boldsymbol{Z}}^{\mathrm{MFIE}}$ as [35]

$$\bar{\boldsymbol{Z}}^{\mathrm{MFIE}}[m,n,a,b] = \left(\frac{ik}{4\pi}\right)^2 \int d^2\hat{\boldsymbol{k}}\, \boldsymbol{R}_{ma}(k,\hat{\boldsymbol{k}},\boldsymbol{r}_C)\; \cdot T_\tau(k,\hat{\boldsymbol{k}},\boldsymbol{r}_C-\boldsymbol{r}_{C'})\boldsymbol{S}_{nb}(k,\hat{\boldsymbol{k}},\boldsymbol{r}_{C'}), \tag{3.28}$$

where

$$\boldsymbol{S}_{nb}(k,\hat{\boldsymbol{k}},\boldsymbol{r}_{C'}) = \frac{\gamma_{nb} l_n}{2A_{nb}} \,(\bar{\bar{\boldsymbol{I}}}_{3\times3} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}}) \cdot \int_{S_{nb}} d\boldsymbol{r'}\,(\boldsymbol{r'} - \boldsymbol{r}_{nb})\, e^{-ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r'}-\boldsymbol{r}_{C'})} \tag{3.29}$$

$$\boldsymbol{R}_{ma}(k,\hat{\boldsymbol{k}},\boldsymbol{r}_C) = \frac{\gamma_{ma} l_m}{2A_{ma}} \,\hat{\boldsymbol{k}} \times \int_{S_{ma}} d\boldsymbol{r}\,[\hat{\boldsymbol{n}} \times (\boldsymbol{r} - \boldsymbol{r}_{ma})]\, e^{ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}-\boldsymbol{r}_C)} \tag{3.30}$$

are the radiation and receiving functions for N-MFIE formulation, respectively. Note that all the integrations are performed over the triangular domains of the respective RWG functions.

3.2 The Multilevel FMM (MLFMA)

We briefly introduced MLFMA in Section 1.2.4. The main idea in FMM and specifically in MLFMA is to compute the interactions between basis and testing functions in a group-wise manner. Creating higher-level groups from one side and organizing the far-field interactions between these groups across multiple levels from the other side are the key features of this powerful algorithm, which makes it suitable for the large EM problems.

3.2.1 The Tree Structure

To calculate far-field interactions in a multilevel fashion, a proper tree structure is needed. To construct the tree, one needs to put the object(s) in a cubic box and start a recursive process to divide the computational domain into subdomains through bisecting the box edges. This process stops either in a given level L, or when the box size reaches to $\lambda/4$ (and sometimes $\lambda/8$), due to the well-known low-frequency breakdown. Note that in such a tree structure, the highest level is the object itself in one piece, and the second level is the MoM level with at most $2^3 = 8$ sub-boxes. Consequently, the first level of the multi-level FMM starts from the third level (i.e., the FMM level) and continues to some deeper levels. Apparently, for an arbitrary geometry, some of the sub-boxes would be empty that should be eliminated.

Somehow similar to FMM, where a box is either a near box or a far box regarding a given box of interest, the tree structure of MLFMA categorizes the boxes into three different groups (as already illustrated in Fig. 1.1):

1. Near boxes with at most $3^3 = 27$ boxes, where the interactions are computed directly based on MoM at the deepest level.
2. Far boxes with at most $7^3 - 3^3 = 316$ boxes per level, where the interactions are computed explicitly in a group-wise manner and based on FMM in each and every level.
3. Very far boxes (with no limit), where some indirect interactions exist among the super groups of the boxes at the higher levels.

3.2.2 MLFMA Stages

A typical MLFMA scenario is illustrated in Fig. 3.3. Here, two sub-boxes labeled as (a) and (b) are in the same level and hence, a conventional FMM scenario can be applied regarding the far-field interactions. But considering the sub-boxes labeled as (a) and (c), they are very far from each other and hence, their interaction must be computed at higher levels, i.e., among the parent interactions. This simply means that, a kind of FMM-based computation is needed, but this time among the parent boxes.

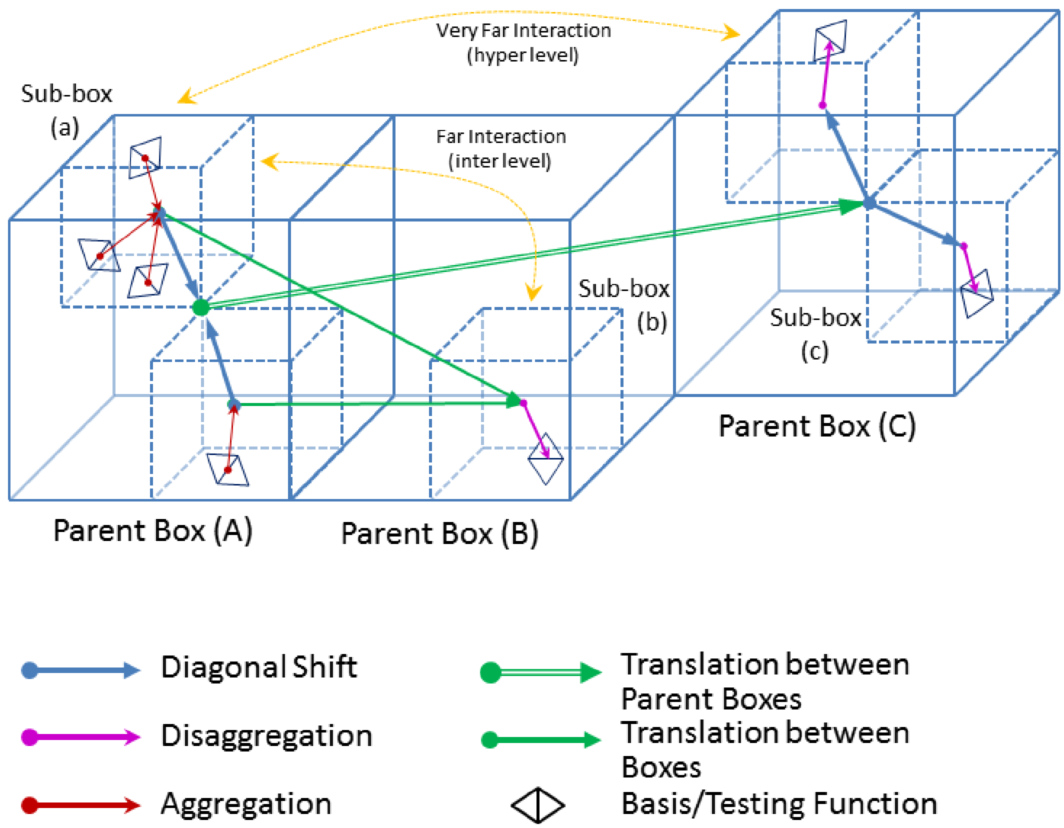


Figure 3.3: Aggregation, translation, and disaggregation processes inside a typical MLFMA scenario.

3.2.2.1 Aggregation

By starting from the lowest level L (i.e., the leaf level), the radiation functions of all basis functions $\boldsymbol{b}_n$ inside the sub-box (a) are computed, stored, and then aggregated at the center of the sub-box by taking into account their coefficients $a[n]$ as

$$\boldsymbol{S}_{c'}(k, \hat{\boldsymbol{k}}, \boldsymbol{r}_{c'}) = \sum_{\boldsymbol{b}_n \in \text{(a)}} a[n]\, \boldsymbol{S}_{nb}(k, \hat{\boldsymbol{k}}, \boldsymbol{r}_{c'}), \tag{3.31}$$

where $\boldsymbol{S}_{nb}$ denotes the radiation function of the respective basis function and $\boldsymbol{r}_{c'}$ refers to the center of the sub-box (a). Aggregation at the leaf level is depicted by red arrows in Fig. 3.3. By climbing to a higher level ($l = L-1, \ldots, 4, 3$), the aggregation process splits into two separate tasks. The first task is to up-sample (interpolate) the radiated fields of all sub-boxes, since higher level (larger) parent box possesses a higher bandwidth than its own sub-boxes. The second task is to perform a proper diagonal shift regarding each of the sub-boxes and then obtain the cumulative radiation function for the parent box (A) at its own center $\boldsymbol{r}_{C'}$. Note that a diagonal shift is nothing more than multiplying with a proper complex exponential function (from the center of the sub-box to the center of the parent box). Both tasks are denoted by upward blue arrows in Fig. 3.3 and can be written as

$$\boldsymbol{O}_{C'}(k, \hat{\boldsymbol{k}}, \boldsymbol{r}_{C'}) = \sum_{\text{boxes} \in (\text{A})} \Big[\boldsymbol{S}_{c'}(k, \hat{\boldsymbol{k}}, \boldsymbol{r}_{c'})\Big]^{\text{up-sampled}} e^{-ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}_{c'} - \boldsymbol{r}_{C'})}, \tag{3.32}$$

where $\boldsymbol{r}_{C'}$ and $\boldsymbol{O}_{C'}$ refer to the center of the parent box (A) and the total outgoing fields transferred to this center, respectively. Based on the bisecting strategy discussed before, there are at most 8 sub-boxes inside a parent box. It is worth to state that, by such a higher-level aggregation process, in fact, we re-aggregate the radiation of all the basis functions that reside inside the parent box to its own center. This is simply because

$$e^{-ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}_{c'} - \boldsymbol{r}_{C'})} \sum e^{-ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}' - \boldsymbol{r}_{c'})} = \sum e^{-ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}' - \boldsymbol{r}_{C'})}. \tag{3.33}$$

3.2.2.2 Translation

When all of the aggregation processes are finished from the leaf level up to the third level (the FMM level), in the translation stage, the outgoing radiated fields of the boxes across the levels are transformed into appropriate incoming fields for other far boxes in the very same level ($l = L, \ldots, 4, 3$). This can be written as

$$\boldsymbol{I}_C(k, \hat{\boldsymbol{k}}, \boldsymbol{r}_C) = \sum_{C' \in Far\{C\}} \boldsymbol{O}_{C'}(k, \hat{\boldsymbol{k}}, \boldsymbol{r}_{C'})\, T_\tau(\boldsymbol{k}, \boldsymbol{r}_C - \boldsymbol{r}_{C'}), \tag{3.34}$$

where $Far\{C\}$ denotes a list of far boxes with respect to a given box center C (i.e., a testing box). Clearly, T_τ represents the translation function, which is responsible for such a transformation from outgoing radiated fields $\boldsymbol{O}_{C'}$ into incoming received fields $\boldsymbol{I}_C$. These translations are shown by two types of green arrows in Fig. 3.3, one for sub-boxes and the other one for parent boxes (at a higher level).

3.2.2.3 Disaggregation

Disaggregation stage starts from the topmost level (level 3) and continues down to the leaf level (level L). For all higher levels other than the leaf level, all of the translated incoming fields are collected at the center of the receiving parent box and then, they are down-sampled and diagonally shifted toward their respective sub-boxes. This can be written as

$$\boldsymbol{I}_c(k, \hat{\boldsymbol{k}}, \boldsymbol{r}_c) = \Big[\boldsymbol{I}_C(k, \hat{\boldsymbol{k}}, \boldsymbol{r}_C)\Big]_{\text{down-sampled}} e^{ik\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}_c - \boldsymbol{r}_C)}, \tag{3.35}$$

where $\boldsymbol{r}_c$ refers to the center of an intended sub-box. Finally at the leaf level, the total incoming fields, either from the higher levels or from the same level, are received by a testing function $\boldsymbol{t}_m(\boldsymbol{r})$ as

$$\sum_{n=1}^{N} \bar{\boldsymbol{Z}}^{far}[m,n]a[n] = \left(\frac{ik}{4\pi}\right)^2 \int d^2\hat{\boldsymbol{k}}\, \boldsymbol{R}_{ma}(k, \hat{\boldsymbol{k}}, \boldsymbol{r}_c) \cdot \Big(\boldsymbol{I}_c(k, \hat{\boldsymbol{k}}, \boldsymbol{r}_c) + \boldsymbol{I}_L(k, \hat{\boldsymbol{k}}, \boldsymbol{r}_L)\Big), \tag{3.36}$$

where $\boldsymbol{I}_L$ represents the incoming fields from some other leaf level boxes in the vicinity.

It should be noted that each of the above mentioned three stages consists of many practical considerations. These are elaborated in full details in [66] along with some other important concepts regarding the efficiency and the complexity of MLFMA. In the next chapter, we demonstrate a newly developed method to deal with the low-frequency breakdown under N-MFIE formulation and then, we proceed on by introducing an amending factor to improve its accuracy.

Chapter 4

Amending Factor for Approximate Diagonalization of the Green's Function at Low Frequencies

4.1 Introduction

Recently, an approximate method for the diagonalization of the Green's function, which is stable at arbitrary short distances, has been developed by Ergül *et al.* [34]. In this method, scaled spherical functions are deployed to stabilize the plane-wave expansions, where the value of the scaling factor depends on the electrical size of the boxes. In this report, we use the same idea and introduce some correction factors to further decrease the error level. By invoking the ideal diagonal form of the Green's function, i.e.,

$$\tilde{g}(\boldsymbol{r}, \boldsymbol{r}') = \frac{e^{ik|\boldsymbol{r}-\boldsymbol{r}'|}}{ik|\boldsymbol{r}-\boldsymbol{r}'|} = \frac{e^{ik|\boldsymbol{w}+\boldsymbol{v}|}}{ik|\boldsymbol{w}+\boldsymbol{v}|} \tag{4.1}$$

$$= \frac{1}{4\pi}\int d^2\hat{\boldsymbol{k}}\; \tilde{\beta}(k, \hat{\boldsymbol{k}}, \boldsymbol{v})\; \tilde{\alpha}(k, \hat{\boldsymbol{k}}, \boldsymbol{w}), \tag{4.2}$$

where, $\tilde{\beta}$ and $\tilde{\alpha}$ are the diagonal shift and translation functions, respectively, we will use the scaled version of the conventional shift function, as proposed in equation (11) of [34], and try to find a "suitable" translation function. By "suitable" we aim to find an $\tilde{\alpha}$ function that reduces the overall error after integration on the unit sphere. The main reason for choosing the method proposed in [34] is that it can be easily implemented into existing MLFMA codes with minimum modifications.

4.2 Some Calculus

By using the infinite series representation of a scaled exponential function

$$\tilde{\beta}_s(k, \hat{\boldsymbol{k}}, \boldsymbol{v}, s) = e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}}{s}} \tag{4.3}$$

$$= \sum_{t=0}^{\infty}(i)^t(2t+1)j_t(\frac{kv}{s})P_t(\hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{v}}), \tag{4.4}$$

and inserting the summation into (4.2), we have

$$\frac{e^{ik|\boldsymbol{w}+\boldsymbol{v}|}}{ik|\boldsymbol{w}+\boldsymbol{v}|} = \frac{1}{4\pi}\int d^2\hat{\boldsymbol{k}} \left(\sum_{t=0}^{\infty}(i)^t(2t+1)j_t(\frac{kv}{s})P_t(\hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{v}})\right)\tilde{\alpha}_s(k, \hat{\boldsymbol{k}}, \boldsymbol{w}, s, v). \tag{4.5}$$

Note that $\tilde{\alpha}_s$ function has two extra parameters, i.e., s due to the scaling and v coming from non-ideal diagonalization. For the sake of simplicity, we consider a general form of the translation function as

$$\tilde{\alpha}_s(k, \hat{\boldsymbol{k}}, \boldsymbol{w}, s, v) = \sum_{t'=0}^{\infty} f_{t'}(\cdot)P_{t'}(\hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{w}}), \tag{4.6}$$

where $f_{t'}(\cdot)$ is assumed not to be a function of $\hat{\boldsymbol{k}}$. Using this assumption, and exchanging the summation and integration, we can write

$$\tilde{g}(\boldsymbol{r}, \boldsymbol{r'}) = \frac{1}{4\pi}\sum_{t=0}^{\infty}\sum_{t'=0}^{\infty}(i)^t(2t+1)j_t(\frac{kv}{s})\, f_{t'}(\cdot)\int d^2\hat{\boldsymbol{k}}\, P_{t'}(\hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{w}})\, P_t(\hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{v}}). \tag{4.7}$$

Based on an identity for the product of two Legendre functions over the unit sphere given in [63]

$$P_t(\hat{\boldsymbol{v}} \cdot \hat{\boldsymbol{w}})\delta_{tt'} = \frac{(2t+1)}{4\pi} \int d^2\hat{\boldsymbol{k}}\, P_{t'}(\hat{\boldsymbol{k}} \cdot \hat{\boldsymbol{w}})\, P_t(\hat{\boldsymbol{k}} \cdot \hat{\boldsymbol{v}}), \tag{4.8}$$

where $\delta_{tt'}$ is the Dirac delta function, we can further simplify (4.7) as

$$\tilde{g}(\boldsymbol{r}, \boldsymbol{r'}) = \sum_{t=0}^{\infty} (i)^t j_t(\frac{kv}{s})\, f_t(\cdot) P_t(\hat{\boldsymbol{v}} \cdot \hat{\boldsymbol{w}}). \tag{4.9}$$

Now, recalling the Gegenbauer's addition theorem for the scalar Green's function [62], which is based on an infinite series of the form

$$\frac{e^{ik|\boldsymbol{w}+\boldsymbol{v}|}}{ik|\boldsymbol{w}+\boldsymbol{v}|} = \sum_{t=0}^{\infty} (-1)^t (2t+1)\, j_t(kv)\, h_t^{(1)}(kw)\, P_t(\hat{\boldsymbol{v}} \cdot \hat{\boldsymbol{w}}) \tag{4.10}$$

and comparing with (4.9), we can easily find $f_t(\cdot)$ as

$$f_t(\cdot) = \frac{(i)^t (2t+1)\, j_t(kv)\, h_t^{(1)}(kw)}{j_t(\frac{kv}{s})}. \tag{4.11}$$

Hence, the final form of the needed translation function would be

$$\tilde{\alpha}_s(k, \hat{\boldsymbol{k}}, \boldsymbol{w}, s, v) = \sum_{t=0}^{\infty} (i)^t (2t+1) \left[\frac{j_t(kv)}{j_t(\frac{kv}{s})}\right] h_t^{(1)}(kw) P_t(\hat{\boldsymbol{k}} \cdot \hat{\boldsymbol{w}}), \tag{4.12}$$

where the term inside the bracket, as we expected, brings an extra parameter v into the list of arguments. Considering s as a scaling factor, we have few choices as follows.

- The choice $s = +1$ turns out to be the standard form of the diagonal translation function, which is unstable at low frequencies.
- The choice $s = -1$ turns out to be the same as above, with only difference that the summation over the unit sphere starts from the lower hemisphere as if $\hat{\boldsymbol{k}}$ is reversed, considering two simple identities, i.e., $j_t(-kv) = j_t(kv)(-1)^t$ and $(-1)^t P_t(\hat{\boldsymbol{k}} \cdot \hat{\boldsymbol{w}}) = P_t(-\hat{\boldsymbol{k}} \cdot \hat{\boldsymbol{w}})$ [62]. It is still unstable at low frequencies.

- The choice $s \neq \pm 1$ gives us an approximate diagonal form for $\tilde{\alpha}_s$ due to the fact that it is not independent of v. In the next section, we discuss about this approximation.

4.3 Some Approximations

Defining a as the box size in the lowest level of the octree structure, the low-frequency breakdown problem manifests itself for box sizes $a \leq \lambda/16$. For this size, the argument of the Bessel function inside the bracket in (4.12) is small enough, which means that we can utilize the small argument approximations in comparison to the order. Apparently, the existence of s as a scaling factor may be problematic using such an assumption, but as we will see later, we can avoid this situation by choosing proper values for s.

Using a mathematical software (e.g., MAPLE®) and by defining $F(k, v, s) = \left[\, j_t(kv)/j_t(\frac{kv}{s})\,\right]$, we compute the Maclaurin series of F up to order 4 about $kv = 0$ for different orders of the Bessel function from $t = 0$ to $t = 9$ as follows.

- $t = 0$

$$1 - \frac{1}{6}\,\frac{(s-1)\,(s+1)\,(kv)^2}{s^2} + \frac{1}{360}\,\frac{(s-1)\,(s+1)\,(3\,s^2-7)\,(kv)^4}{s^4} \tag{4.13}$$

- $t = 1$

$$s - \frac{1}{10}\,\frac{(s-1)\,(s+1)\,(kv)^2}{s} + \frac{1}{1400}\,\frac{(s-1)\,(s+1)\,(5\,s^2-9)\,(kv)^4}{s^3} \tag{4.14}$$

- $t = 2$

$$s^2 - \frac{1}{14}\,\frac{(s-1)\,(s+1)\,(kv)^2}{1} + \frac{1}{3528}\,\frac{(s-1)\,(s+1)\,(7\,s^2-11)\,(kv)^4}{s^2} \tag{4.15}$$

$$\vdots$$

- $t = 9$

$$s^9 - \frac{1}{42}\,s^7(s-1)\,(s+1)\,(kv)^2 + \frac{s^5}{81144}(s-1)\,(s+1)\left(21\,s^2-25\right)(kv)^4 \tag{4.16}$$

Note that the only term in these expansions that is independent of (kv) is s^t. Consequently, as the first approximation for equation (4.12), ignoring powers of (kv), we can write

$$\tilde{\alpha}_s(k, \hat{\boldsymbol{k}}, \boldsymbol{w}, s) = \sum_{t=0}^{\infty} (i)^t (2t+1) \left(s^t\right) h_t^{(1)}(kw) P_t(\hat{\boldsymbol{k}} \cdot \hat{\boldsymbol{w}}), \tag{4.17}$$

which is exactly the same expression as equation (6) given in [34]. It is worth mentioning that even though this independence with respect to v seems very useful, it imposes somehow a fixed amount of error regardless of the distance between the basis and testing boxes. This can be considered as the penalty of using (4.3) instead of its scaled infinite series given by (4.4). The reason for this behavior may be described by using the asymptotic formulas for the Hankel function due to small arguments. As we know, $h_t^{(1)}(z) = j_t(z) + i\, y_t(z)$, where j_t and y_t are spherical Bessel functions of the first and second kind, respectively. For small arguments, the real part of the Hankel function approaches to zero, but its imaginary part has an asymptotic behavior, which can be approximated by $-\frac{(2t)!}{2^t\, t!} z^{-(t+1)}$. Considering the scaling factor s^t in (4.17), it is readily seen that each term of this summation is a multiplication of three terms: (1) a fixed coefficient, (2) a normalized term of $(\frac{s}{z})^t$, and (3) an extra unnormalized term of $1/z$. This is important, because as we move to farther boxes or even to higher levels, we expect to see less error, but unfortunately the amount of error at larger distances due to larger translation ($\boldsymbol{w}$) vectors does not vanish. The graphical representation of this issue will be addressed in next sections.

As a better approximation for (4.12), we can include some other terms containing the higher even orders of (kv) in (4.13)–(4.16). However, we have to consider that any correction to $\tilde{\alpha}_s$ needs to be independent of basis and testing boxes, because we are going to reuse the same translation function among different boxes inside each level. To resolve this dilemma, we can normalize the second terms of the expansions in (4.13)–(4.16) with respect to s^t and approximate $F(k, v, s)$ as

- $t = 0 \longrightarrow s^0\, (1 - \frac{1}{6}\, \frac{(s-1)(s+1)(kv)^2}{s^2})$
- $t = 1 \longrightarrow s^1\, (1 - \frac{1}{10}\, \frac{(s-1)(s+1)(kv)^2}{s^2})$

$\vdots \qquad\qquad\qquad\qquad \vdots$

- $t = 9 \longrightarrow s^9\,(1 - \frac{1}{42}\,\frac{(s-1)(s+1)(kv)^2}{s^2})$

Now for small boxes, $\frac{(kv)^2}{s^2}$ will be small enough to dominate the second terms of above approximations inside the parentheses. Therefore, the coefficients of these terms (i.e., $\frac{1}{6}$ for $t = 0$ or $\frac{1}{42}$ for $t = 9$) are all approximated as $\frac{1}{6}$ for the sake of simplicity. It should be noted that a wide range of values with different combinations were tested and finally it is revealed that the best results can be obtained when $\frac{1}{6}$ is used. As a result, (4.12) is approximated as

$$\tilde{\alpha}_s(k, \hat{\boldsymbol{k}}, \boldsymbol{w}, s, v) \cong \left[1 - \frac{1}{6}\,\frac{(s-1)\,(s+1)\,(kv)^2}{s^2}\right] \sum_{t=0}^{\infty} (i)^t (2t+1) s^t h_t^{(1)}(kw) P_t(\hat{\boldsymbol{k}} \cdot \hat{\boldsymbol{w}}) \tag{4.18}$$

$$= \left[1 - \frac{1}{6}\,\frac{(s-1)\,(s+1)\,(kv)^2}{s^2}\right] \tilde{\alpha}_s(k, \hat{\boldsymbol{k}}, \boldsymbol{w}, s). \tag{4.19}$$

Writing this way helps us to break the translation function into two multiplicative terms. The first term depends on v and acts as a correction factor, while the second term is independent of v. Consequently, we can detach the correction factor from $\tilde{\alpha}_s$ function and recast it as a correction factor to the shift function ($\tilde{\beta}_s$). The only remaining issue is that, we have to take into account both of the aggregation and disaggregation parts simultaneously, which means that we have to divide the correction factor into non-equal parts, due to the fact that aggregation and disaggregation vectors are independent and can be different. To do this, we can consider another approximation as

$$\left[1 - \frac{1}{6}\,\frac{(s-1)\,(s+1)\left(k\sqrt{v_{agg}^2 + v_{dis}^2 + 2\,v_{agg}\,v_{dis}\,\cos\psi}\right)^2}{s^2}\right] \cong$$
$$\left[1 - \frac{1}{6}\,\frac{(s-1)\,(s+1)\,(kv_{agg})^2}{s^2}\right] \times \left[1 - \frac{1}{6}\,\frac{(s-1)\,(s+1)\,(kv_{dis})^2}{s^2}\right], \tag{4.20}$$

where v_{agg} and v_{dis} denote the aggregation and disaggregation vectors, respectively, and ψ is the angle between them which is an unknown parameter among the boxes interacting with each other. Note that in (4.20) two distinct terms concerning the product of v_{agg} and v_{dis} have been omitted due to the fact that they are small enough to be ignored without considerably affecting the overall results. First, the term related to $\frac{k^2}{s^2} v_{agg}\, v_{dis} \cos\psi$ and second, the much more smaller term consisting of $\frac{1}{36}(\frac{kv_{agg}}{s})^2(\frac{kv_{dis}}{s})^2$.

Now, by defining the correction factor as

$$C(k, s, \boldsymbol{v}) = \left[1 - \frac{1}{6}\,\frac{(s-1)\,(s+1)\,(k\,|\boldsymbol{v}|)^2}{s^2}\right], \tag{4.21}$$

we can rewrite (4.2) as

$$\tilde{g}(\boldsymbol{r}, \boldsymbol{r'}) \cong \frac{1}{4\pi} \int d^2\hat{\boldsymbol{k}}\, \left[\tilde{\beta}(k, \hat{\boldsymbol{k}}, \boldsymbol{v}_{agg}, s) C_{agg}(k, s, \boldsymbol{v}_{agg})\right] \tilde{\alpha}_s(k, \hat{\boldsymbol{k}}, \boldsymbol{w}, s) \\ \left[\tilde{\beta}_s(k, \hat{\boldsymbol{k}}, \boldsymbol{v}_{dis}, s)\, C_{dis}(k, s, \boldsymbol{v}_{dis})\right], \tag{4.22}$$

where C_{agg} and C_{dis} denote needed correction factors for aggregation and disaggregation operations.

4.4 Numerical Results

4.4.1 Single Level FMM

To have an estimation about the accuracy of all the approximations we went through, we start by computing the error coming from point-by-point interactions inside two far boxes within the leaf level. We use the well-known "one-buffer-box" scheme as usual to define far boxes in each level. Considering the fact that the far testing box may be any of the boxes along three main directions within a radius of $3a$ from the center of the basis box, we have to consider $(3+1+3)^3 - 3^3 = 316$ different combinations for $\tilde{\alpha}_s$ function, which is a bit large number to be considered and compared among the results.

To decrease the number of possible combinations, we note the following.

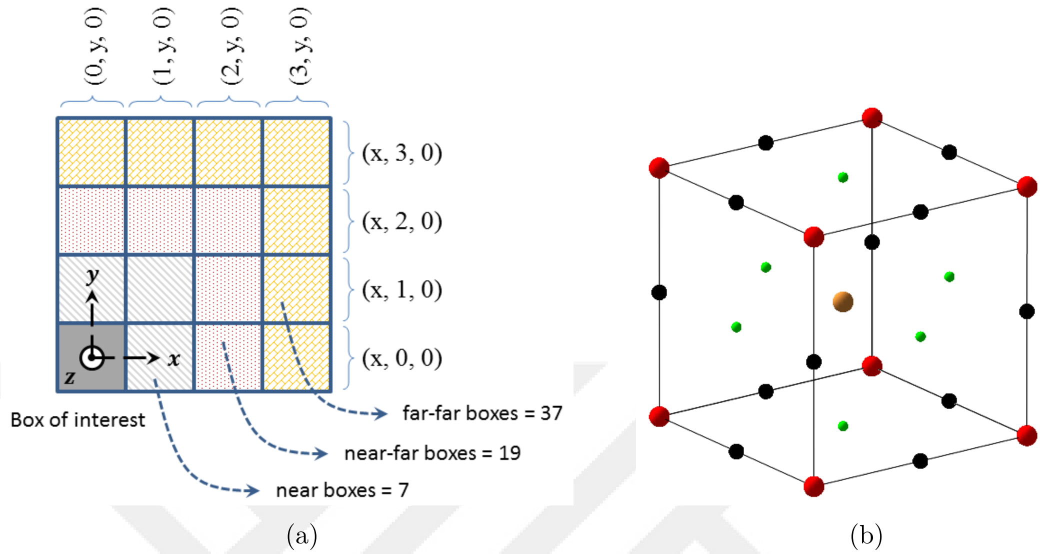


Figure 4.1: (a) 2D representation of different testing boxes with respect to the basis box of interest. Ordered triple sets are shown just for the xy plane. (b) 27 different points inside each box consisting of 8 at corners, 12 at mid-sides, 6 at mid-faces and 1 at the center of the box.

1. $P_n(\boldsymbol{k} \cdot \boldsymbol{w})$ is the only direction-dependent term in $\tilde{\alpha}_s$.

2. No matter how many number of harmonics (τ) we use to truncate the infinite series in $\tilde{\alpha}_s$ computation, we always use $\tau + 1$ sampling points in the θ direction (Gaussian-Legendre) and $2(\tau + 1)$ sampling points in the ϕ direction (uniform) [66]. Careful consideration of these directions reveals that, in the $\hat{\boldsymbol{k}}$ space over unit sphere, we have a complete symmetry with respect to three main planes, i.e., xy, xz and yz planes, which divide the whole sphere into 8 equal sections.

By considering these simple facts, one can readily conclude that it is enough to consider only those boxes that reside in the positive side of the axes of the coordinate system, i.e., just $(1 + 3)^3 - 2^3 = 56$ far boxes are enough, to have an estimation about different and possible directions affecting the $\tilde{\alpha}_s$ function. These boxes and their relative positions with respect to the basis box of interest, beside a simple labeling scheme to name them, is shown in Fig. 4.1(a).

A careful investigation regarding the addition theorem reveals that inside each box, the points at the vertices are prone to the highest level of error. Consequently, we choose 27 points in each box, consisting of one point at the center, 8 points at the corners, 12 points at the middle of the sides, and 6 points at the middle of the faces, as shown in Fig. 4.1(b). To have a precise information about the position of far boxes and also points inside each box in forthcoming numerical results, we decided to reorder the box numbers as follows (note that these box numbers will be used in upcoming figures).

1. Boxes numbered from 1 to 3, which belong to near-far boxes in Fig 4.1(a), are the closest ones to the basis box. More precisely, these are (2,0,0), (0,2,0) and (0,0,2) (not shown here) in three different directions. We will call them as group A.

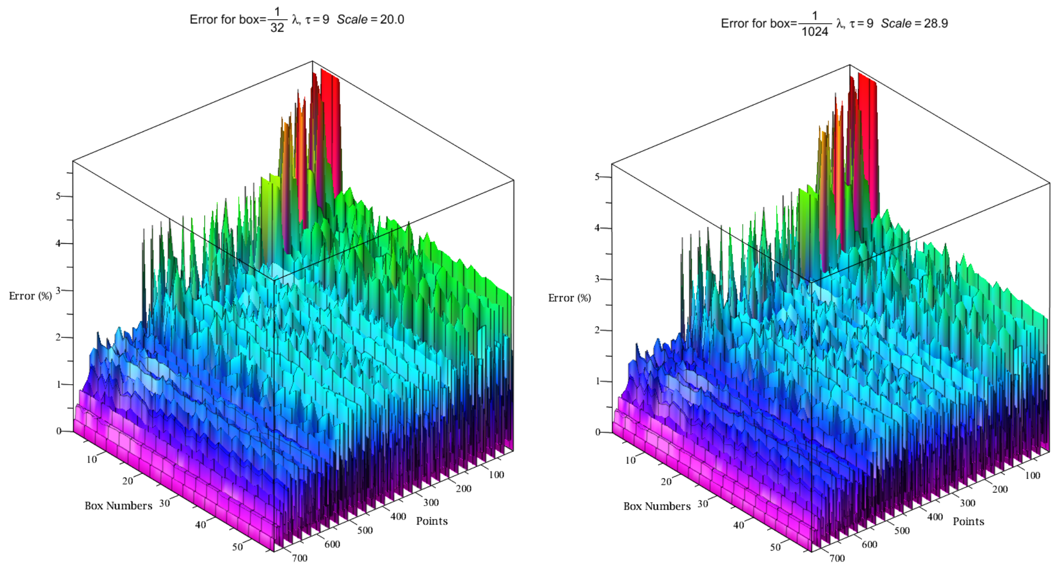
2. Boxes numbered from 4 to 9, which again belong to near-far boxes, are the next closest boxes comparing to the first group. These are (0,1,2),(0,2,1),(1,0,2),(1,2,0),(2,0,1) and (2,1,0). We will call them as group B.

3. Other boxes consisting of 13 near-far boxes and 37 far-far boxes are at the third level of closeness to the basis box, which we call them as group C.

4. Similar to boxes, first 8 points belong to vertices, from 9 to 20 belong to the middle of the sides, from 21 to 26 belong to the middle of the faces, and 27th point is at the center of the box.

First we try to verify the results of [34]. To do this, we use $\tilde{\beta}_s$ function of (4.3), $\tilde{\alpha}_s$ function of (4.17), and the scaling factors already given in the paper, which are repeated in Table 4.1 for more convenience. The simulation results for two box sizes, i.e., $a = \lambda/32$ and $a = \lambda/1024$ are shown in Fig. 4.2(a) and Fig. 4.2(b), respectively. Based on these results, we note the followings.

1. The maximum error is about 5.8% for $a = \lambda/32$ and 5.4% for $a = \lambda/1024$, corresponding to group A boxes and the first 64 points that belong to vertices.



(a) (b)

Figure 4.2: Percentage of error using (4.3) and (4.17) as shift and translation functions for 56 far boxes and 27 different points inside each box. Scales shown here are normalized to a/λ, where (a) $a = \lambda/32$ with $s = 20.0$ and (b) $a = \lambda/1024$ with $s = 28.9$.

2. Next maximum error is about 3.9% for $a = \lambda/32$ and 3.4% for $a = \lambda/1024$, corresponding to group B boxes and the points located at the middle of the sides.

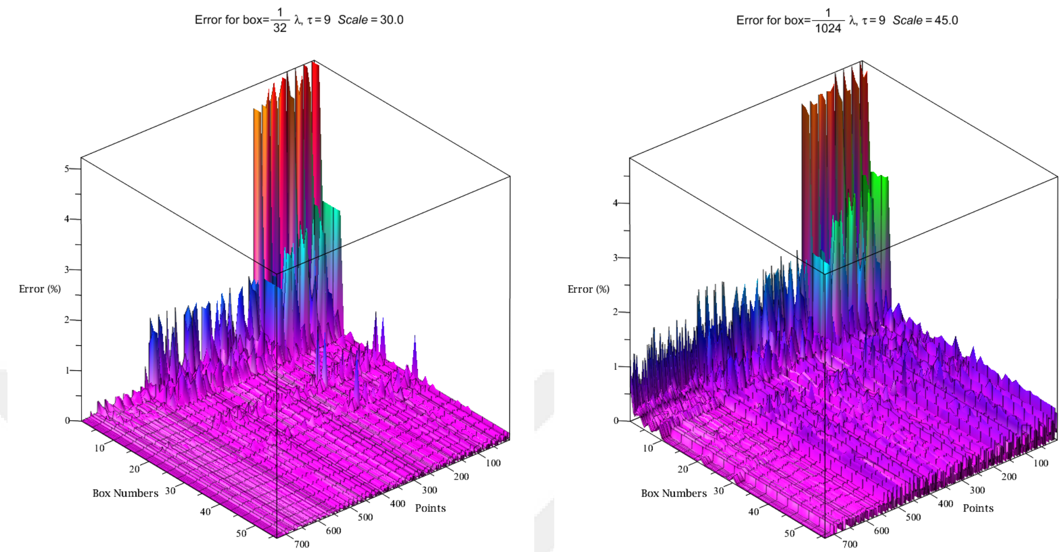
3. The scales shown in the pictures are given by the authors as the best ones with a minimum achievable worst-case error.

4. The maximum error area (in red) is limited to a portion of the plot that belongs to the first 9 boxes and the first 64 points (i.e., group A + B boxes and 8 points of vertices).

5. For group C boxes, the amount of error doesn't change so much. On the other hand, no matter how much we are close to the basis box or far from it, the error is almost fixed around 2.7% $\mapsto$ 3.3%. This effect, as briefly mentioned in the previous section, is mostly because of using scaled $\tilde{\alpha}_s$ function.

6. Lower amount of errors correspond to all group of boxes with points at the middle of the faces.

7. Minimum errors, as we expected, belong to the center of the boxes. These



(a) (b)

Figure 4.3: Percentage of error using (4.22) for 56 far boxes and 27 different points inside each box. Scales shown here are normalized to a/λ, where (a) $a = \lambda/32$ and (b) $a = \lambda/1024$.

are in pink color and they are repeated every other 27 points in between, on the axis labeled by "Points". Note that the interaction between 27th point of each box with 27th point of another box (center-to-center interaction) is illustrated by 729 (27×27)th point in the plot (the end point).

Exactly the same simulation has been carried out with correction factor applied to the system using (4.22). The results are shown in Fig. 4.3 for some optimized values of scaling factor s. The procedure to find these optimal values are described at the end of this subsection after defining an error gauge.

We can conclude a few facts comparing these two plots with their counterparts.

1. The maximum error is reduced by 10% corresponding to group A of boxes and the first 64 points, which belong to the vertices.

2. The next maximum error is reduced by more than 15%, corresponding to the group B of boxes and the points that are located at the middle of the sides.

3. As we expected, the maximum error area (in red) again is limited to a portion

of the plot that belongs to the first 9 boxes and the first 64 points (i.e., group A + B boxes and 8 points of vertices).

4. For group C boxes, we have more than 90% decrease in the error level, which means that the contribution of farther boxes in the overall error is negligible, which was not true in previous plots.

Table 4.1: Optimal Scaling-Factor Values for Minimum Worst-Case Errors in The Approximate Diagonalization of The Green's Function [34].

Box Size (a)	Scaling Factor (s)	$s\lambda/a$
$\lambda/32$	6.26E-1	20.0
$\lambda/64$	4.03E-1	25.8
$\lambda/128$	2.19E-1	28.0
$\lambda/256$	1.12E-1	28.7
$\lambda/512$	5.63E-2	28.8
$\lambda/1024$	2.82E-2	28.9
$\lambda/2048$	1.41E-2	28.9
$\lambda/4096$	7.04E-3	28.8
$\lambda/8192$	3.52E-3	28.8
$\lambda/16384$	1.76E-3	28.8
$\lambda/32768$	8.81E-4	28.9

By considering these results, we have to define an error gauge as a figure of merit to be able to compare these and forthcoming results. We can define the volume under these "Error" plots weighted by proper coefficients and normalized to the number of sampling points, as an error gauge. To be more precise, we have to apply some weights to these points as they do not have equal contributions in the overall error.

- 3 boxes among 56 have the maximum amount of errors. These are group A boxes.
- 6 boxes among 56 have the next maximum errors. These are group B boxes.
- 47 remaining boxes are middle-far or far-far boxes.

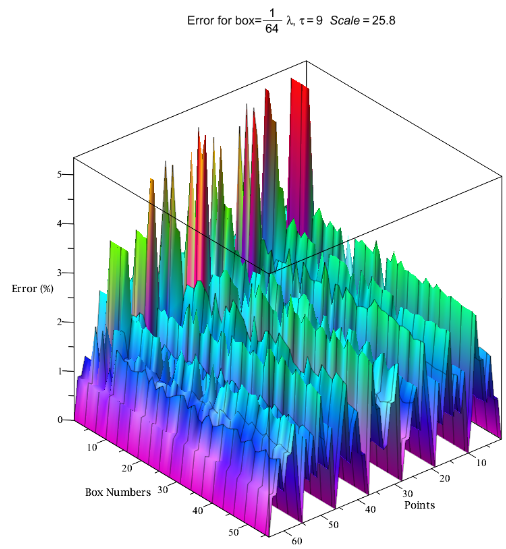
An important fact that should be considered is that, when we deal with real objects with an overly-meshed scenario, most of the basis functions inside the nearby boxes have rather flat surfaces or they possess very high radius of curvature, otherwise the mesh density around this area is not enough to capture the details of the object. Consequently, for most of the cases in the low-frequency regime, we are facing with middle-far and far-far boxes lying over nearly flat surfaces. This is promising because then we have just 2 boxes among 3 for maximum error and 4 boxes among 6 for the next maximum errors, while the rest of the boxes are far enough. Considering all of these facts, we define our error gauge as

$$\begin{aligned} Err = \tfrac{1}{\text{points}\times\text{boxes}} \times \big(&\tfrac{3}{56}\,[\text{volume under the error plot for group A boxes } (1 \mapsto 3)] \\ &+ \tfrac{6}{56}\,[\text{volume under the error plot for group B boxes } (4 \mapsto 9)] \\ &+ \tfrac{47}{56}\,[\text{volume under the error plot for group C boxes } (10 \mapsto 56)]\big), \end{aligned} \tag{4.23}$$

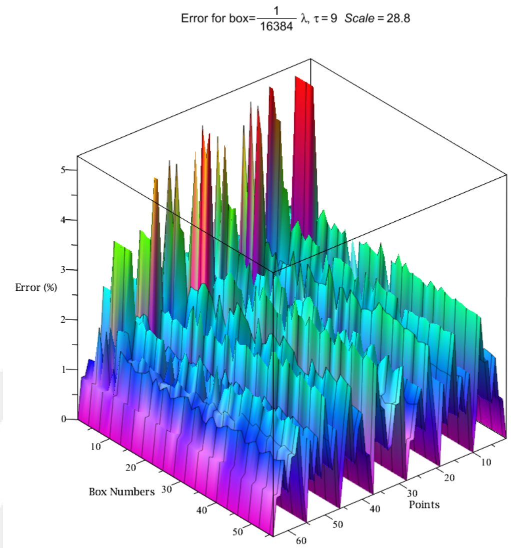
which is somehow a prorated percentage of error per box per interaction. Using this error gauge for the results of Fig 4.2(a) and (b) we get $Err = 0.712\%$ and $Err = 0.561\%$ respectively, and for Fig. 4.3(a) and (b), we get $Err = 0.030\%$ and $Err = 0.109\%$, which clearly shows that applying the correction factor effectively decreases the amount of error.

Based on our observations regarding sensitive boxes, we limit the number of points to those located at the vertices. This helps us to have a closer look regarding the maximum amount of error. Obviously, those points belonging to the middle of the sides and faces as well as the center of the box have less contribution in the total error. For example, using 56 boxes each with 8 points at the vertices, the error plots are shown in Fig. 4.4(a) and (b) for $a = \lambda/64$ and $a = \lambda/2^{14}$, respectively. The corresponding errors are $Err = 0.888\%$ and $Err = 0.852\%$, respectively. As it is clear from the picture, they are almost the same with less than 1% difference in between. The same types of plots are shown in Fig. 4.5(a) and (b) after applying the proper corrections.

It should be mentioned that by minimizing the error gauge of (4.23) for different box sizes, the new optimal values for the scaling factor may be calculated. These optimal values are given in Table 4.2.



(a)



(b)

Figure 4.4: Percentage of error using (4.3) and (4.17) as shift and translation functions for 56 far boxes and 8 points at vertices inside each box. Scales are normalized to a/λ, where (a) $a = \lambda/64$ and (b) $a = \lambda/2^{14}$.

4.4.2 Two Level FMM

We apply the proposed correction for two levels using the same far boxes as before. The important fact to bear in mind is that the box sizes in this higher level are doubled, each of which containing eight smaller boxes.

Again, we may have 56 testing boxes (groups **A**,**B** and **C**[1]), each containing 8 leaf-level smaller boxes, each of which with 27 points that are interacting with another 27 basis points located inside 8 leaf-level boxes.

Based on the elucidation given above and our numerical experiments regarding the behavior of the addition theorem, we realized that utilizing a wisely prepared labeling scheme can effectively reduce the number of needed interactions to be computed. Based on a few trial and errors, if we label eight different boxes in each larger box exactly the same way as we label the points at eight corners of the leaf-level boxes, then considering only these eight corner points will be completely

[1]We use boldface letters to distinguish between groups A,B and C at the leaf-level and **A**,**B** and **C** at the higher levels.

Table 4.2: New Optimal Scaling-Factors for Minimum Worst-Case Errors Using Correction Factors.

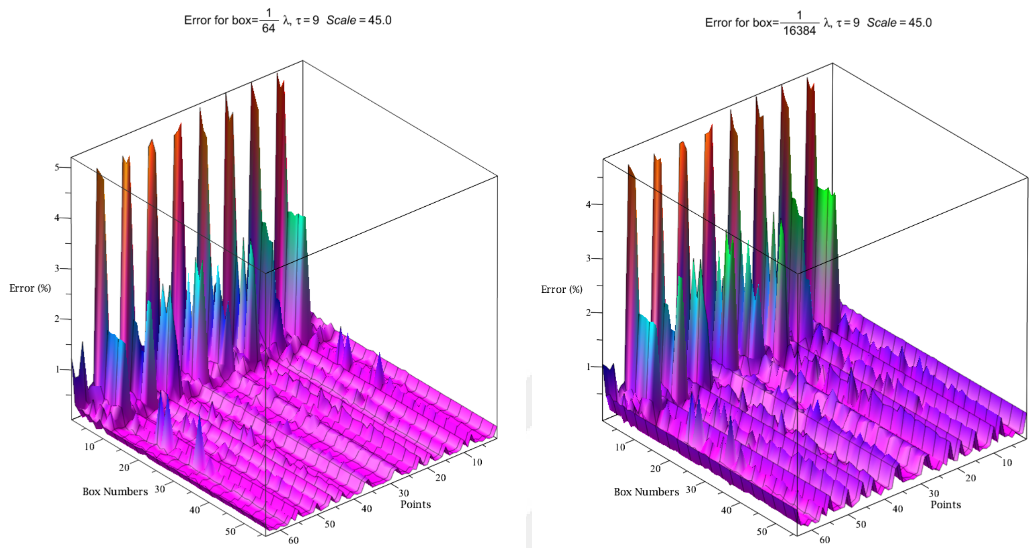
Box Size (a)	Scaling Factor (s)	$s\lambda/a$
$\lambda/32$	6.26E-1	16.65
$\lambda/64$	4.03E-1	19.25
$\lambda/128$	2.19E-1	19.96
$\lambda/256$	1.12E-1	20.18
$\lambda/512$	5.63E-2	20.22
$\lambda/1024$	2.82E-2	20.24
$\lambda/2048$	1.41E-2	20.24
$\lambda/4096$	7.04E-3	20.25
$\lambda/8192$	3.52E-3	20.26
$\lambda/16384$	1.76E-3	20.26
$\lambda/32768$	8.81E-4	20.26

enough to have a rough estimation of the maximum error, as well as the error distribution among the boxes at different locations.

4.4.2.1 Without Correction

To test the proposed scheme, we use the scaled versions of the shift and translation functions as before, i.e., (4.3) and (4.17), to have an estimation about the error level without using any correction factor. As mentioned previously, the group of near-far boxes (group **B**) are prone to have the maximum errors among the others. We choose $(2, 0, 1)$ (see Fig. 4.1(a)), as it is one of the worst cases among the leaf-level testing boxes. Fig. 4.6(a) illustrates the percentage of error for $\tau = 9$ and $a = \lambda/2^{10}$ with a given scaling factor of $s = 28.9$. Fig. 4.6(b) is the top-view of part (a) to have an overview about how the error has spread out over the boxes of interest.

As stated before, due to the above-mentioned labeling scheme, the group of maximum errors are almost always located diametrically. Here we can see 13.56% error as the worst case. To identify the box number (or equally, point number),



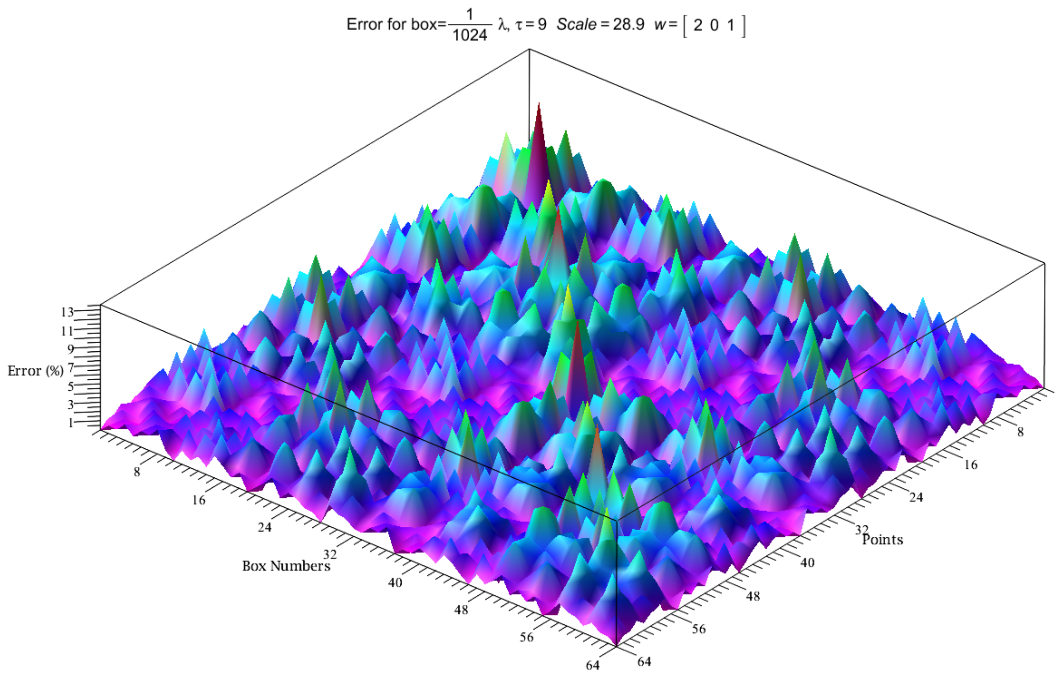
(a) (b)

Figure 4.5: Percentage of error using (4.22) for 56 far boxes and 8 points at vertices inside each box. Scales are normalized to a/λ, where (a) $a = \lambda/64$ and (b) $a = \lambda/2^{14}$.

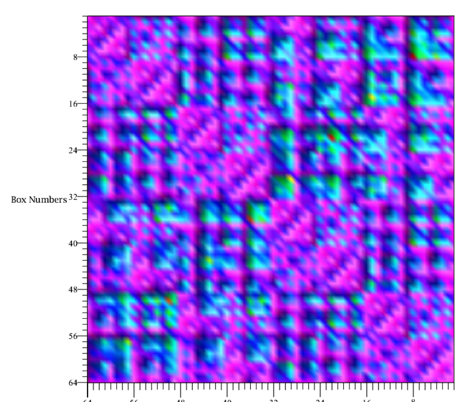
two simple formulas may come in handy:

$$\text{basis box} = \lfloor \frac{\text{Box Number} - 1}{8} \rfloor + 1$$
$$\text{testing box} = \text{Box Number} - 8 \times \lfloor \frac{\text{Box Number} - 1}{8} \rfloor.$$

It is important to emphasize that, even though we are dealing with larger translation distances, the error level is a bit higher than expected. Also note that the distance between basis and testing boxes is not an important parameter to delimit the error. To verify this claim, we choose the last testing box, i.e., (3,3,3) and reproduce a similar plot as shown in Fig. 4.7. Here the maximum error decreases only by 22% taking into account that we are at the farthest possible box with respect to the basis box. It can be seen that the maximum error is still located around the diagonal points of the plot.



(a)



(b)

Figure 4.6: Percentage of error for two-level FMM based on (4.3) and (4.17) using 8 boxes, each with 8 corner points for both of the basis and testing boxes. Scales shown here are normalized to a/λ. (a) The translation vector $\boldsymbol{w}$ is chosen to be the worst case for $a = \lambda/1024$. (b) Top view of the same plot to distinguish 8×8 different basis and testing boxes.

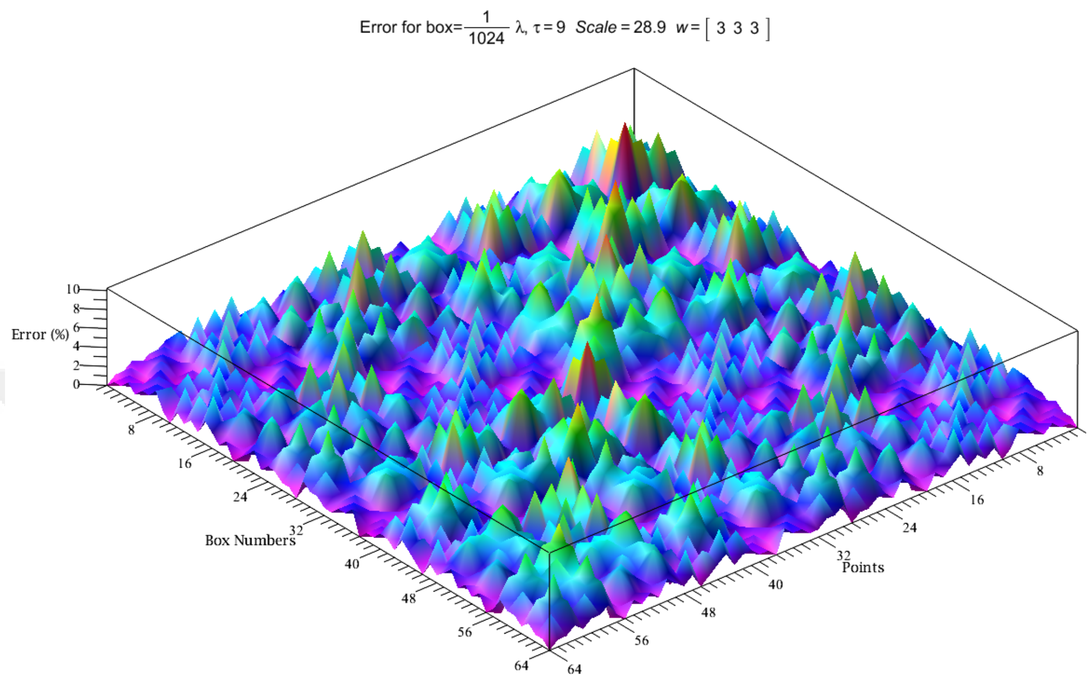


Figure 4.7: The same plot as Fig. 4.6 except for farthest testing box, i.e., $[3, 3, 3]$.

4.4.2.2 With Correction

By utilizing the proposed correction factor given by (4.22), to be applied in both of the levels, we investigate the potential of this correction. We expect to get two simultaneous results:

1. Lower errors for near-far and middle-far boxes (group **A** & **B**).
2. Much lower errors for the farther boxes at the higher levels.

Based on our simulations, the optimum value for the scaling factor s is a matter of optimization. Its optimal value somehow depends on the relative direction of $\boldsymbol{w}$ vector with respect to the basis and testing boxes and also on the distance between these two, which in turn, depends on the current box size. Dependency on the direction of the translation vector $\boldsymbol{w}$ is due to the fact that we ignore the angle between the aggregation and disaggregation vectors, i.e., $\cos(\psi)$ in (4.20). Considering all of these inconsistent parameters, we have two choices as follows.

1. Using the same scaling factor s for all of the levels, optimized for the overall least error.

2. Using specific scaling factor for each level, aiming to minimize the error in a systematic manner.

Actually the former is the normal way of applying the correction factor without any increase in complexity order, i.e., $\mathcal{O}(N\ \log N)$, but the later, increases the complexity to $\mathcal{O}(N\ \log^2 N)$, because each correction inside the individual levels has to be carried out with a proper scaling starting from the basis box up to the desired testing box at the higher levels. Before applying these correction factors, we have to modify the error gauge defined in (4.23). This is mainly because of some noticeable changes in the pattern of error plots that we are going to use hereafter for comparison purposes. Despite the fact that we have to consider all of the 56 different far boxes to define a fair and solid error gauge, this is not practically possible due to the huge number of combinations as the number of levels increases. So it suffices to use a few known worst cases that would be group **B** boxes as discussed before.

By removing the weighting coefficients in (4.23) and computing the volume under the error plot, we may redefine the error gauge as

$$Err(\%) = \frac{1}{\text{points} \times \text{boxes}} \times (\text{volume under the error plot})\,. \tag{4.24}$$

Applying this error gauge for plots shown in Fig. 4.6 and Fig. 4.7 ends up with 2.106% and 2.045%, respectively which seems to be quite high.

4.4.2.3 Correction: Based on Multilevel Optimization

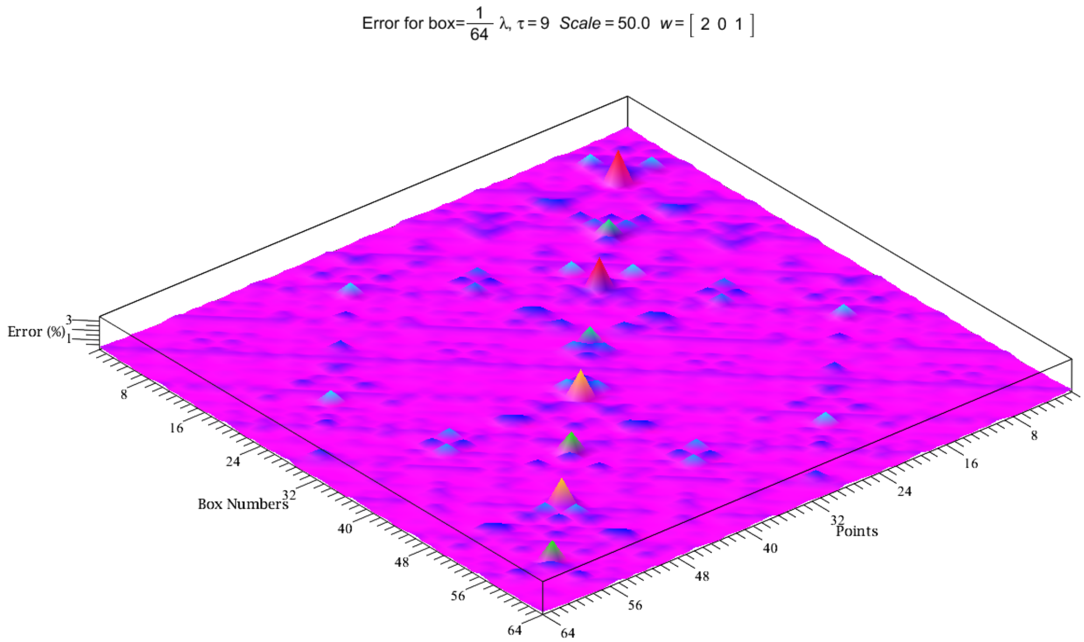
In this approach, we choose a proper scaling factor to optimize the response of this method as a whole. Although this is not the best approach for a minimum error, our simulations clarify that its outcome is quite satisfactory, comparing to its uncorrected counterpart. Based on a wide range of simulations, it is revealed that s should be a function of the electrical size of the boxes, where $s = 50\,a/\lambda$ seems to be a good choice for this case. The error plots for $a = \lambda/64$ and $a = \lambda/2^{10}$ regarding one of the worst cases are depicted in Fig. 4.8(a) and (b), respectively. The respective error gauges are $Err = 0.415\%$ and 0.164%. Beside the fact that the latter is 12.8 times less than the corresponding error gauge referring to Fig. 4.6, the maximum error is also 2.6 times less than it.

So far we have shown that the correction factor can satisfy our first expectation already declared in the previous section. To illustrated whether it satisfies our second expectation, we attempt to show the effect of applying the correction factor to the farthest testing box, exactly similar to the case shown in Fig. 4.7. The resulted plot, shown in Fig. 4.9 reveals that, not only the maximum error level is reduced by about 75%, but also the error gauge shows 80% reduction from $Err = 2.045\%$ to $Err = 0.395\%$.

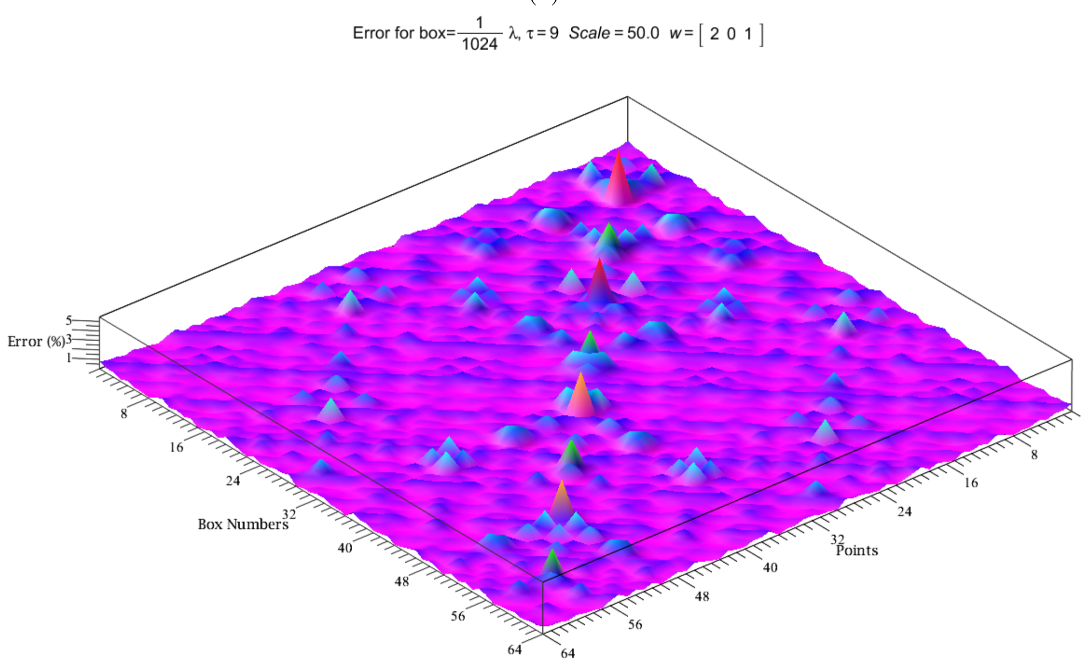
4.4.2.4 Correction: Based on Level-Wise Optimization

Very similar to the multilevel optimization method, we try to find proper scaling factors for each and every level individually based on the corresponding box sizes. These factors are chosen based on a wide range of numerical simulations due to the fact that we cannot find an analytic optimum point as a function of box size. Repeating the same scenario for different levels, we can compute and save an appropriate lookup table, which can be used for other simulations as a reference table.

As stated previously, using dedicated scaling factors for each level increases the complexity. It stems mainly from the fact that for each level, the whole cycle of the testing operation, i.e., aggregation, translation, and dissaggregation, along with their corresponding scaling and corrections, has to be carried out exactly the same way as the other levels. It simply means that we never use aggregated values of the lower levels at the higher levels. Hence, every testing cycle starts from a group of basis boxes and ends up with the desired testing boxes. This simple fact guaranties that previously discussed single level FMM (Section 4.4.1) works fine



(a)



(b)

Figure 4.8: Percentage of error for two-level FMM based on a global optimization and considering one of the worst cases. Scales shown here are normalized to a/λ, where (a) $a = \lambda/64$ and (b) $a = \lambda/1024$.

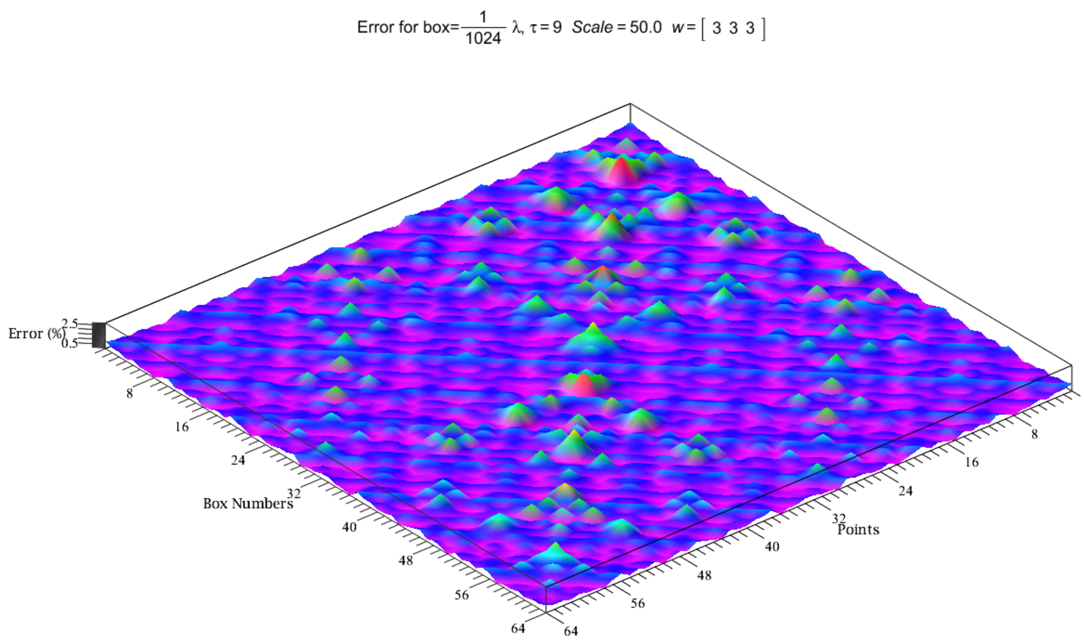


Figure 4.9: Percentage of error for two-level FMM based on global optimization by considering the farthest testing box, i.e., $[3, 3, 3]$. Compare with Fig. 4.6.

for multilevel FMM but carries the penalty of slightly increased complexity from $\mathcal{O}(N\ \log N)$ to $\mathcal{O}(N\ \log^2 N)$.

We have to emphasize that the level-wise optimization is revealed to be a better choice and it is used in the context of IL-MLFMA due to the following reasons.

1. As mentioned in the previous section, a proper choice for the scale factor s is a function of $\frac{a}{\lambda}$ and unfortunately, this function is not linear.

2. Using MLFMA with a few number of FMM levels, it is possible to find an almost optimum scale factor to optimize the overall correction process based on a multilevel optimization scheme. However, such a universal scale factor depends on the average size of the boxes, which in turn depends on the size of the object itself. This simply means that we have to take care of both the object size and the total number of levels to find an optimum value for s, may be based on a large lookup table or an object-dependent separate pre-processing routine, both of which are not feasible.

3. As we will see in the next chapter, 5 up to 12 (and even higher) levels

of IL-MLFMA is quite normal to solve some medium-sized canonical or even real-life EM problems. Consequently, finding a universal scale factor is obviously not efficient or even practical.

4.5 Corrected Version of the MFIE Formulation

In order to implement a wide-band MLFMA based on the N-MFIE formulation (see Section 2.4.2), we need to replace the high-frequency stable diagonalized Green's function with the so called "full-frequency" stable, approximate version of the digonalized Green's function. Both of these expansions are repeated here for convenience, i.e.,

$$\tilde{g}(\boldsymbol{r}, \boldsymbol{r'}) = \frac{e^{ik|\boldsymbol{w}+\boldsymbol{v}|}}{ik|\boldsymbol{w}+\boldsymbol{v}|} = \frac{1}{4\pi}\int d^2\hat{\boldsymbol{k}}\ \tilde{\beta}_s(k, \hat{\boldsymbol{k}}, \boldsymbol{v})\ \tilde{\alpha}_s(k, \hat{\boldsymbol{k}}, \boldsymbol{w}), \tag{4.25}$$

$$\begin{aligned}\tilde{g}(\boldsymbol{r}, \boldsymbol{r'}) \cong \frac{1}{4\pi}\int d^2\hat{\boldsymbol{k}} \left[\tilde{\beta}_s(k, \hat{\boldsymbol{k}}, \boldsymbol{v}_{agg}, s) C_{agg}(k, s, \boldsymbol{v}_{agg})\right] . \tilde{\alpha}_s(k, \hat{\boldsymbol{k}}, \boldsymbol{w}, s). \\ \left[\tilde{\beta}_s(k, \hat{\boldsymbol{k}}, \boldsymbol{v}_{dis}, s)\, C_{dis}(k, s, \boldsymbol{v}_{dis})\right],\end{aligned} \tag{4.26}$$

where $\tilde{\beta}_s$ and $\tilde{\alpha}_s$ are defined in (4.3) and (4.17), respectively. To have a more compact representation of the correction factor, C can be written as

$$C(k, s, \boldsymbol{v}) = \left[1 - \frac{1}{6}\,\frac{(s-1)\,(s+1)\,(k\,|\boldsymbol{v}|)^2}{s^2}\right] = 1 + q\,|\boldsymbol{v}|^2 = 1 + q\,\boldsymbol{v}\cdot\boldsymbol{v}, \tag{4.27}$$

where

$$q = -\frac{1}{6}\,\frac{(s-1)\,(s+1)\,k^2}{s^2} \tag{4.28}$$

and $\boldsymbol{v}$ may be any of the aggregation or disaggregation vectors (i.e., $\boldsymbol{v}_{agg}$ or $\boldsymbol{v}_{dis}$). To derive the corrected version of the N-MFIE formulation, we can use the definition of $\mathcal{K}_{\mathrm{PV}}\{\boldsymbol{X}\}(\boldsymbol{r})$ operator in (2.62) and write

$$\begin{aligned}\boldsymbol{K}^{N}_{\mathrm{Cor}}[m,n] &= \int d\boldsymbol{r}\boldsymbol{t}_m(\boldsymbol{r})\cdot\left[\hat{\boldsymbol{n}}\times\int_{\mathrm{PV}} d\boldsymbol{r}'\boldsymbol{b}_n(\boldsymbol{r}')\times\nabla' g(\boldsymbol{r},\boldsymbol{r}')\right]\\ &= -\int d\boldsymbol{r}\left[\boldsymbol{t}_m(\boldsymbol{r})\times\hat{\boldsymbol{n}}\right]\cdot\int_{\mathrm{PV}} d\boldsymbol{r}'\boldsymbol{b}_n(\boldsymbol{r}')\times\nabla g(\boldsymbol{r},\boldsymbol{r}').\end{aligned}\tag{4.29}$$

To calculate the gradient of the newly defined Green's function with respect to the non-primed coordinates, we have to notice that only $\boldsymbol{v}_{dis}=\boldsymbol{r}-\boldsymbol{r}_m$ is a function of $\boldsymbol{r}$ and hence,

$$\begin{aligned}\nabla\tilde{g}(\boldsymbol{r},\boldsymbol{r}') \cong \frac{1}{4\pi}\int d^2\hat{\boldsymbol{k}}&\left[e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}_{agg}}{s}}(1+q\,\boldsymbol{v}_{agg}\cdot\boldsymbol{v}_{agg})\right]\tilde{\alpha}_s(k,\hat{\boldsymbol{k}},\boldsymbol{w},s)\\ &\nabla\left[e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}_{dis}}{s}}(1+q\,\boldsymbol{v}_{dis}\cdot\boldsymbol{v}_{dis})\right].\end{aligned}\tag{4.30}$$

The last gradient can be carried out and simplified as

$$\nabla\left[e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}_{dis}}{s}}(1+q\,\boldsymbol{v}_{dis}\cdot\boldsymbol{v}_{dis})\right] = e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}_{dis}}{s}}\left[\frac{ik\hat{\boldsymbol{k}}}{s}\,q_1+2q\,\boldsymbol{v}_{dis}\right],\tag{4.31}$$

where

$$q_1 = 1+q\,|\boldsymbol{v}_{dis}|^2.\tag{4.32}$$

Inserting (4.31) in (4.30) and taking into account that $g=\tilde{g}\frac{ik}{4\pi}$, we have

$$\begin{aligned}\boldsymbol{K}^{N}_{\mathrm{Cor}}[m,n] = -\frac{ik}{(4\pi)^2}&\int d\boldsymbol{r}\left[\boldsymbol{t}_m(\boldsymbol{r})\times\hat{\boldsymbol{n}}\right]\cdot\int d\boldsymbol{r}'\boldsymbol{b}_n(\boldsymbol{r}')\times\\ &\int d^2\hat{\boldsymbol{k}}\left[e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}_{agg}}{s}}(1+q\,|\boldsymbol{v}_{agg}|^2)\right]\tilde{\alpha}_s\left[e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}_{dis}}{s}}(\frac{ik\hat{\boldsymbol{k}}}{s}\,q_1+2q\,\boldsymbol{v}_{dis})\right].\end{aligned}\tag{4.33}$$

By exchanging the order of the integrals and using the vector identity $\boldsymbol{A}\cdot\boldsymbol{B}\times\boldsymbol{C} = -(\boldsymbol{C}\times(-\boldsymbol{A}))\cdot\boldsymbol{B}$, we can rewrite the same expression as

$$\begin{aligned}
\boldsymbol{K}^{N}_{\text{Cor}}[m,n] =& \frac{ik^2}{(4\pi)^2}\int d^2\hat{\boldsymbol{k}}\int d\boldsymbol{r}\int_{\text{PV}} d\boldsymbol{r'}\Big[\frac{i\hat{\boldsymbol{k}}}{s}\,q_1 + \frac{2q}{k}\,\boldsymbol{v}_{dis}\Big]\times[\hat{\boldsymbol{n}}\times\boldsymbol{t}_m(\boldsymbol{r})]\cdot \\
& \boldsymbol{b}_n(\boldsymbol{r'})\left[e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}_{agg}}{s}}(1+q\,|\boldsymbol{v}_{agg}|^2)\right]\tilde{\alpha}_s\left[e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}_{dis}}{s}}\right] \\
=& \frac{ik^2}{(4\pi)^2}\int d^2\hat{\boldsymbol{k}}\int d\boldsymbol{r}e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}_{dis}}{s}}\Big(\frac{i\hat{\boldsymbol{k}}}{s}\,q_1 + \frac{2q}{k}\,\boldsymbol{v}_{dis}\Big)\times\Big(\hat{\boldsymbol{n}}\times\boldsymbol{t}_m(\boldsymbol{r})\Big) \\
& \tilde{\alpha}_s\cdot\left[\int_{\text{PV}} d\boldsymbol{r'}e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}_{agg}}{s}}(1+q\,|\boldsymbol{v}_{agg}|^2)\,\boldsymbol{b}_n(\boldsymbol{r'})\right].
\end{aligned} \tag{4.34}$$

Now by defining corrected versions of the radiation and receiving functions as

$$\boldsymbol{S}^{\mathcal{K},\text{N}}_{n,\text{Cor}}(\boldsymbol{k},\boldsymbol{r}_n) = \int d\boldsymbol{r'}e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}_{agg}}{s}}(1+q\,|\boldsymbol{v}_{agg}|^2)\,\boldsymbol{b}_n(\boldsymbol{r'}) \tag{4.35}$$

$$\boldsymbol{R}^{\mathcal{K},\text{N}}_{m,\text{Cor}}(\boldsymbol{k},\boldsymbol{r}_m) = \int d\boldsymbol{r}\,e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}_{dis}}{s}}\Big[\frac{\hat{\boldsymbol{k}}}{s}\,(1+q\,|\boldsymbol{v}_{dis}|^2) - i\frac{2q}{k}\,\boldsymbol{v}_{dis}\Big]\times\Big[\hat{\boldsymbol{n}}\times\boldsymbol{t}_m(\boldsymbol{r})\Big], \tag{4.36}$$

we can rewrite the last expression for $\boldsymbol{K}^{N}_{\text{Cor}}$ in a more compact and elegant form as

$$\boldsymbol{K}^{N}_{\text{Cor}}[m,n] = (\frac{ik}{4\pi})^2\int d^2\hat{\boldsymbol{k}}\,\boldsymbol{R}^{\mathcal{K},\text{N}}_{m,\text{Cor}}(\boldsymbol{k},\boldsymbol{r}_m)\,\tilde{\alpha}_s(k,\hat{\boldsymbol{k}},\boldsymbol{w},s)\;\cdot\;\boldsymbol{S}^{\mathcal{K},\text{N}}_{n,\text{Cor}}(\boldsymbol{k},\boldsymbol{r}_n). \tag{4.37}$$

Note that these new radiation and receiving functions reduce to the previously defined functions in (3.29) and (3.30) when we deal with conventional MLFMA without any scaling or correction, i.e., $s=\pm1 \Rightarrow q=0\,\&\,q_1=1$.

4.6 Comparison Among Three MLFMA Versions

A perfect electric conductor (PEC) sphere of radius R=0.3 m ($\lambda/250$) at f = 4.0 MHz with a uniform mesh consisting of 984 triangles (1476 RWG functions) is used to investigate the accuracy of three MLFMA versions, i.e., the conventional MLFMA, the scaled-MLFMA, and its corrected version using (4.34)–(4.37). In all simulations, second-order Gaussian quadrature is used in the integrations over the surfaces of the triangles to capture rapid field variations in near-field computations. Regarding the far-field computations, the first-order Gaussian quadrature is used to carry out the required interactions. Furthermore, the solutions are obtained without any preconditioning using the generalized minimal residual (GMRES) iterative solver with error tolerance of 10^{-3} and a restart value of 100, while the maximum number of iterations is fixed at 1000.

Results obtained from these three versions are compared to that of the Mie-series approach, which is an analytic method for scattering from PEC sphere problems. To assess the accuracy, the relative root-mean-square (RMS) error is used, given by

$$\text{Relative RMS Error} = \sqrt{\frac{\sum_{i=1}^{N} \left|\boldsymbol{E}_{\theta_i}^{\text{Sim}} - \boldsymbol{E}_{\theta_i}^{\text{Mie}}\right|^2}{\sum_{i=1}^{N} \left|\boldsymbol{E}_{\theta_i}^{\text{Mie}}\right|^2}}, \tag{4.38}$$

where $\boldsymbol{E}_{\theta_i}^{\text{Sim}}$ and $\boldsymbol{E}_{\theta_i}^{\text{Mie}}$ are the dominant θ component of the far-zone electric field obtained from the MLFMA simulations and the Mie-series solution, respectively.

Fig. 4.10 depicts the far-zone electric field as a function of the bistatic angle, along with the Mie-series solution, where 0° and 180° correspond to the backscattering and forward-scattering directions, respectively. Field values around the forward-scattering direction, i.e., from 170° to 190°, are further focused in an inset. The relative RMS errors for these results are 6.51% for the conventional MLFMA, 2.35% for the scaled MLFMA, and 2.28% for the corrected version of the scaled MLFMA, respectively. It is seen that there is a little improvement due to the use of higher order corrections applied to the scaled MLFMA, and hence, this formulation is used to implement the IL-MLFMA in the next chapter.

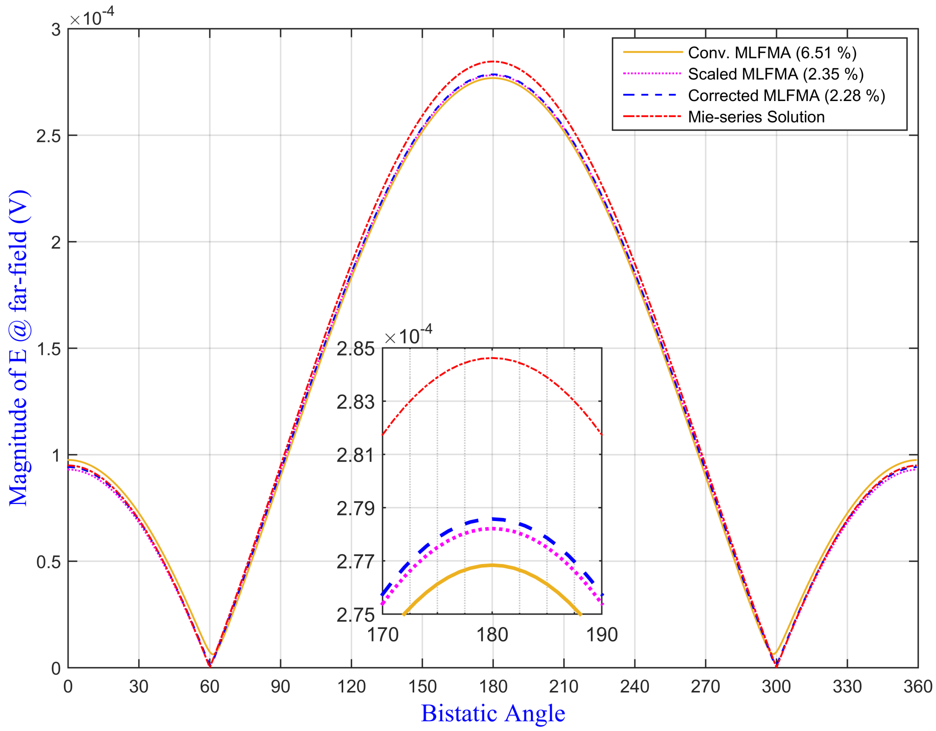


Figure 4.10: A comparison among three versions of MLFMA, i.e., the conventional MLFMA, the scaled-MLFMA, and its corrected version using the Mie-series solution as the analytic (exact) solution. Field values around the forward-scattering direction, i.e., from 170° to 190°, are further focused in an inset, where a small improvement can be seen in the corrected version.

Chapter 5

Incomplete-Leaf MLFMA and its Implementation

5.1 Introduction

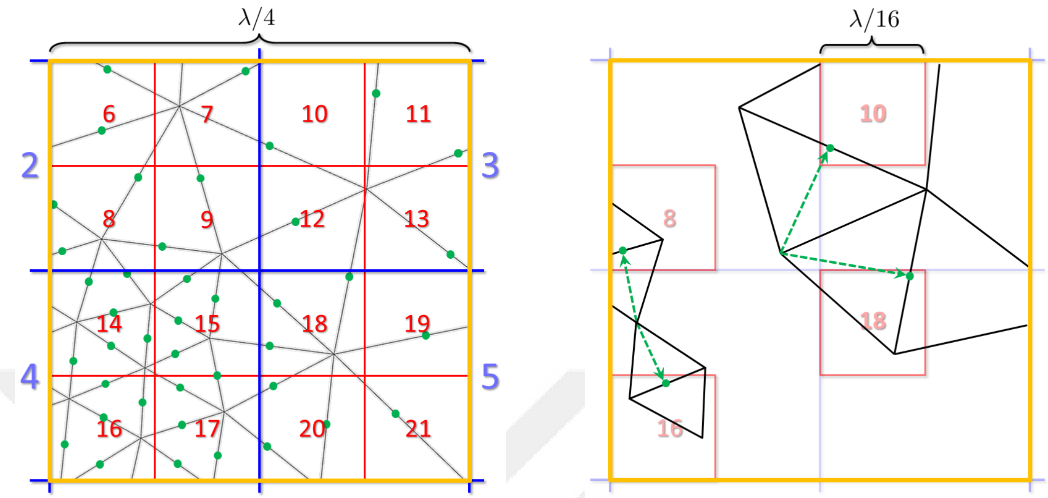
In this dissertation, we modify the tree structure of the conventional MLFMA, and deploy the approximate method for the diagonalization of the Green's function which is discussed in the previous chapter to handle the low-frequency problems of MLFMA. Thus, a wideband MLFMA, capable of handling multiscale electromagnetic problems regarding closed-surface PEC objects is presented. It is worth mentioning that, even though the previously discussed corrected/scaled version of MLFMA is capable of handling low-frequency problems that usually appear in a wide frequency spectrum, there are serious accuracy/efficiency issues regarding the multiscale meshes originating from the fact that available methods (including the conventional MLFMA) deploy fixed-size boxes at each and every level of the corresponding tree structure.

In multiscale problems, very dense meshes over the entire geometry can be used to possess the fine details, so that accurate results can be obtained using fixed-size boxes. However, such an attempt is a brute-force solution with a huge number of unknowns requiring excess amount of memory in addition to being extremely inefficient even when large numbers of levels are employed.

A better and widely preferred approach is to use nonuniform meshes to discretize multiscale problems. In this approach, one may use large boxes together with lower number of levels to maintain the desired accuracy. Unfortunately, in this case, an $\mathcal{O}(N^2)$ complexity is inevitable in near-field computations due to boxes enclosing locally overmeshed regions. On the other hand, one can use smaller boxes with some deep levels of MLFMA to avoid the $\mathcal{O}(N^2)$ complexity. However, leaf level boxes at deep levels are too small to hold basis/testing functions. Parts of basis/testing functions may be located in different boxes and some of these boxes may even be in the far-zone of each other based on the one-buffer-box criterion that is frequently used in MLFMA implementations. Hence, critical computational errors contaminate the accuracy. Furthermore, this error becomes more significant as the multiscale factor increases.

Therefore, in this dissertation for multiscale problems, in addition to using variable mesh sizes, variable box sizes are used so that both $\mathcal{O}(N^2)$ complexity coming from the near-field calculations of overcrowded boxes and the computational errors arising from the protrusion of basis/testing functions are eliminated.

To have a clear idea about the imposed errors and the necessity of introducing the novel incomplete-leaf concept to overcome them, we consider a flat two-dimensional mesh with a multiscale factor of 4 in Fig. 5.1(a). The size of the largest box shown in the figure is assumed to be $\lambda/4$. Hence, after two consecutive clustering (i.e., recursively bisecting the boxes in both directions), one reaches to the minimum box size of $\lambda/16$. To demonstrate the borders of some typical boxes in three consecutive MLFMA levels, different colors are used in Fig. 5.1, where the largest (yellow) box casts as the top level, four (blue) boxes numbered from 2 to 5 as the next lower level, and finally, 16 small (red) boxes numbered from 6 to 21 as the second lower level. Midpoints of all edges are labeled with small (green) circles to track the edge-based RWG functions. Based on the three levels and the boxes involved, one can observe two important issues that are interfering with each other.



(a) (b)

Figure 5.1: A demonstration of computational errors due to multiscale meshes. (a) Recursive clustering of a multiscale mesh. (b) Some problematic interactions between touching RWG functions that are interpreted as far-field in the given clustering scheme.

5.1.1 Considering Box Sizes

- The largest box with a size of $\lambda/4$ contains 40 edges (RWGs). It costs $40^2 = 1600$ near-field interactions by itself, and some others with possible nearby boxes.
- The lower-level boxes contain 5 to 19 RWGs. They involve 25 to 361 near-field interactions by themselves and some others with nearby boxes. But those boxes involve large errors in far-field interactions either due to their small size ($\lambda/8$) or due to those triangles sticking out of the boxes.
- The lowest level boxes contain 1 to 6 RWGs. They involve 1 to 36 near-field interactions by themselves and some others with nearby boxes. Considerable errors regarding far-field interactions occur, mainly due to very small box sizes ($\lambda/16$) and severe protrusions.

Fortunately, deploying the corrected/scaled MLFMA introduced in the previous chapter or some other broadband solvers [29],[32],[23] can effectively eliminate, or at least alleviate box-size-related inaccuracies to some limited extent. This limit is governed by the size of the RWG functions in such a way that after some critical point, we may not have a complete RWG function embedded inside the box of interest. On the other hand, some or even many of those halved boxes are so small that they can hardly enclose the functions, no mater how powerful and accurate our computational routines are. Note that this limit has a close relation with the multiscale factor and the distribution of box populations in each and every level. Obviously, uniform or mostly uniform meshes do not suffer from such sort of problems.

5.1.2 Considering Box Populations

- The largest box contains almost 40 RWG functions, but only 17 (=42%) of them are completely inside the box.
- The lower-level boxes contain 5 to 19 RWGs, but only the 4th box contains 6 (=30%) complete functions.
- The lowest-level boxes contain 1 to 6 RWGs, and none of them contains a complete RWG function.

This simply means that, by increasing the number of levels over multiscale meshes possessing high multiscale factor, a new error source is introduced regarding the far-field computations. To have an intuition about this effect, consider two boxes, i.e., 18 and 10, that are independently depicted in Fig. 5.1(b) for further clarification. Considering the level where the red boxes are concerned, these two boxes are supposed to be far-field boxes based on the one-buffer-box criterion. Hence, we attempt to use the addition theorem to test the radiated field of an RWG function located inside the 18^{th} box over any RWG function located inside the 10th box. However, both RWG functions have a common node inside the 9th box and even a common edge partially inside the 12^{th} box (the buffer box), which means, they are in fact practically near to each other.

At this stage, we note that if we consider only the edge centers as the source and test points, they completely satisfy the necessary condition needed for far-field assumption and totally fulfills the usage of the addition theorem (or may be its corrected/scaled version). But considering these midpoints as representatives of their corresponding RWG basis and testing functions, this assumption is not valid anymore, and their interactions are prone to a considerable error. The same argument holds for the other RWG functions that are protruded out of their corresponding boxes. Increasing the box size effectively diminishes the error, but on the other side, it dramatically increases the computation time due to increased number of RWGs inside the boxes leading to $\mathcal{O}(N^2)$ complexity in self and near interactions of boxes. Obviously, for real-life large multiscale problems with millions of unknowns, we have no other choice than increasing the number of levels to run away from quadratically increasing number of near-field interactions. It should be mentioned that this problem becomes worse for three-dimensional structures (and the worst case happens for the VIEs via volumetric discretization, which will be covered in the next chapter).

5.2 Incomplete-Leaf Tree Structure

The aforementioned deficiencies arising from large RWG functions inside relatively small boxes can effectively be eliminated by totally modifying the whole clustering strategy. In the new strategy there is no need to divide all the boxes across a given level to create the next level, therefore, some boxes with a lower RWG population may stay at an upper level while those with a higher population can be further divided into smaller boxes. While such a clustering strategy seems trivial, its implementation needs detailed investigation of box-to-box interactions and their careful organizations for an efficient and accurate solver. Note that some other alternative options regarding this clustering scenario could be used. Among them are the RWG population, the size or aspect ratio of the basis/testing functions, and the position of the integration points inside the triangles.

In this dissertation, we present an implementation, where the boxes with RWG populations exceeding some predetermined threshold (that may be a function of the corresponding level) are split into smaller boxes. This is achieved by letting the program to start from an arbitrary level, which is mainly the second level (i.e., the MoM level), and continuing the clustering process as far as it takes, such that none of the leaf boxes (those boxes at the very end of any branch) are overpopulated. However, it is also possible to start by a uniform clustering for a few initial levels, and then continue with a nonuniform clustering after a given level. This is useful to prevent some leftover RWG functions at the corners of some boxes (to be explained later). Finally, via a semi-automatic process with a kind of end truncation, the program continues the division process up to a maximum allowed level. In this scenario, some of the last level boxes may be overcrowded. This is mainly to prevent a situation that the very last level boxes are not capable of enclosing the triangles any more. This is very similar to our 2D example even in the topmost level. Note that, at topmost level indicated by the yellow box (#1), we have a fairly high population, say 40 RWG/box, but strictly speaking, only 42% of them are complete without any protrusion.

It should be emphasized that the novel IL tree structure can be reduced to a traditional one by simply assigning unity population threshold for all higher levels except the leaf level. Then the program starts from the second level and continues the division process equally everywhere up to a maximum allowed level (though some of the last level boxes may be overcrowded), and hence, the conventional MLFMA is recovered.

5.2.1 Box Populations

To determine how deep the program has to continue the partitioning process and to make sure that none of the boxes inside any of the levels has a population more than desired, we have to perform two simultaneous tasks. We have to recursively slice the object in three dimensions and then, take the population record of those non-empty boxes. This is somehow memory consuming and also CPU demanding operation at the higher levels, and should be carried out carefully. For example at the 10th level, each side of the cube that encloses the object has to be divided into $2^{10-1} = 512$ smaller boxes, which means that at each of the 512 layer of the sliced object, we have $512 \times 512 \simeq 262K$ boxes and we have to check all of the RWG functions to see which one belongs to which box. Taking into account that we have no prior information about the box populations, we have to consider all of the boxes across 512 layers, which simply leads us to $512 \times (512 \times 512) \times 8$(Byte) $\simeq 1$ GBytes of memory. Note that we need the population information for each non-empty box to decide whether it is an overcrowded box or not.

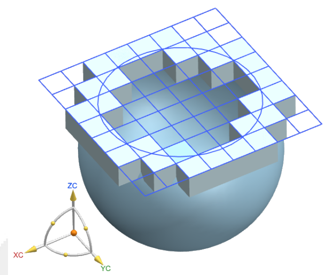


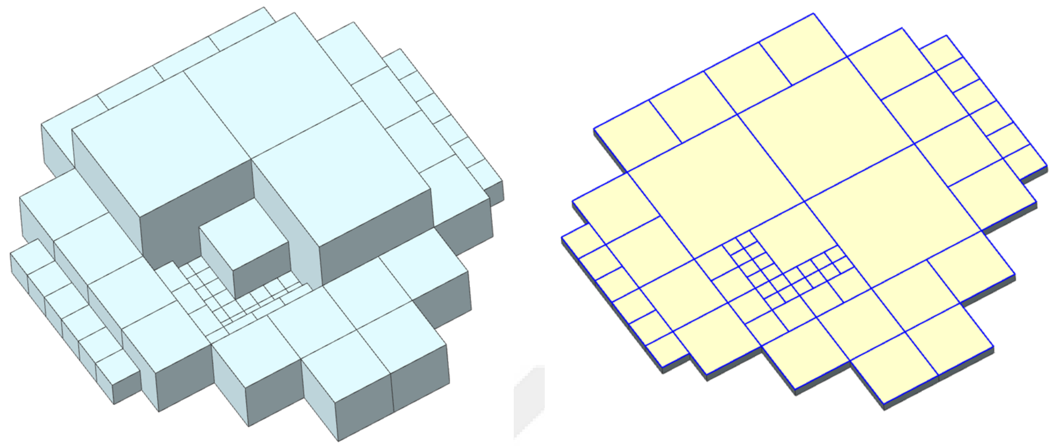
Figure 5.2: Using a sparse matrix to represent each slice of the object.

As a remedy, we can utilize the "sparse" facility in conjunction with the "cell" structure of MATLAB (even though they slightly reduce the efficiency of the the algorithm) to alleviate this bottleneck. Each element of a cell structure in MATLAB can hold an independently separate sparse matrix, which can be associated with each layer along the z axis. This procedure, partially shown in Fig. 5.2, is extremely useful because empty layers act as empty elements inside the cell structure, and empty boxes in each layer are not saved in the sparse representation of that layer, hence, saving huge amount of memory.

5.2.2 The New Tree Structure

To define some new concepts, establish new rules and algorithms in a structured manner, and to be able to implement the IL-tree structure efficiently and accurately, we commence by wisely prepared 2D pictures and then generalize them to their 3D versions. Fig. 5.3(a) illustrates six-level nonuniform clustering for an object (without showing the object itself) that might be electrically large and possess very fine local features. Its 2D cross sectional view is shown in Fig. 5.3(b). The potential benefits regarding this kind of 2D illustration are as follows.

- Although we cannot see the real object, it is easy to imagine its mesh



(a) (b)

Figure 5.3: A sample of forthcoming 2D pictures representing how they are created from corresponding 3D structures. (a) 3D structure of a typical object after 6 levels of partitioning into proper boxes. (b) 2D cross section of the 3D frame.

density since none of the boxes have more than n RWGs, where n is a predetermined value (e.g. 50). This is useful due to the fact that, each box has the same computational cost (regardless of its size) and hence, one can have an educated guess about the total computational cost of the whole process.

- By looking at a 2D cross section, we may acquire enough information to imagine the relative sizes of the RWG functions at different levels. Obviously, large boxes on top (i.e., first couple of) levels contain relatively large RWG functions, and small boxes at deep levels may contain extremely small RWG functions.
- It is considerably easier to extend the new processing algorithms from 2D cases to 3D ones just by generalizing the rules.

To handle such a nonuniform clustering scheme, we need a threshold, which may be fixed or a function of level. Having this threshold as the maximum RWG functions per box, three different types of boxes should be defined as listed in Table 5.1.

Table 5.1: Different Box Definitions for the Modified Tree Structure

Box type	Definition and Properties
Overcrowded Box (OCB)	A box with a population equal to or more than the given threshold. • An overcrowded box cannot be located at the last level unless we use the semi-automatic process discussed in Section 5.2.
Truncated Box (TB)	A box with a population less than the given threshold such that it should be the end box along the corresponding ramified branch. • A truncated box may exist at any level. • There must be no other boxes branching from truncated boxes. • For semi-automatic process, the last level overcrowded boxes will be considered as truncated ones.
Pruned Box (PB)	A virtual box, which was supposed to be one of the extensions for a truncated box. • It is not possible to have a pruned box at the second level.

Hereinafter in this dissertation, we use OCB, TB, and PB abbreviations instead of their long names, i.e., “OverCrowded Box(es)”, “Truncated Box(es)”, and “Pruned Box(es)”, respectively. Furthermore, to identify these boxes inside the program, we will use positive population for both of the OCBs and PBs and negative population for TBs. Obviously positive values used for both OBs and PBs will not be confusing due to the fact that there is a clear distinction between these two: the population for an OCB is always higher than the a given threshold while for a PB, it is always less than it.

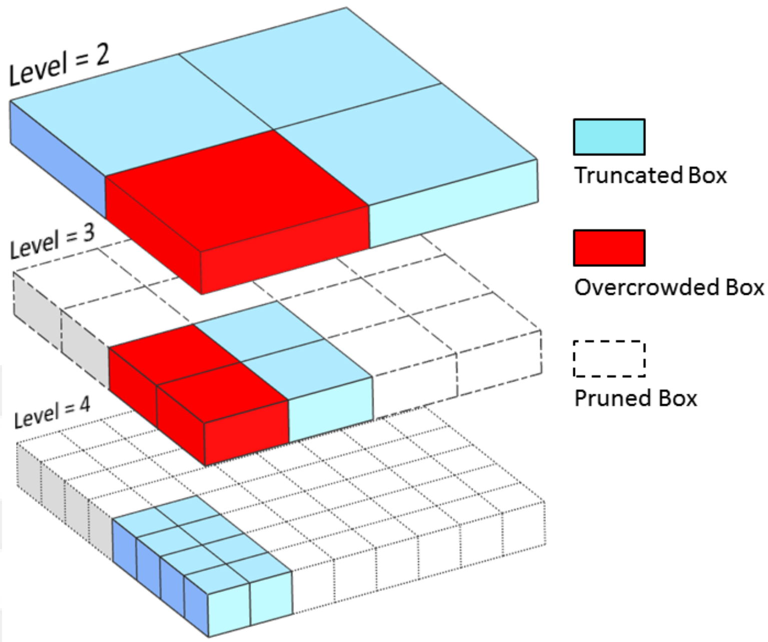
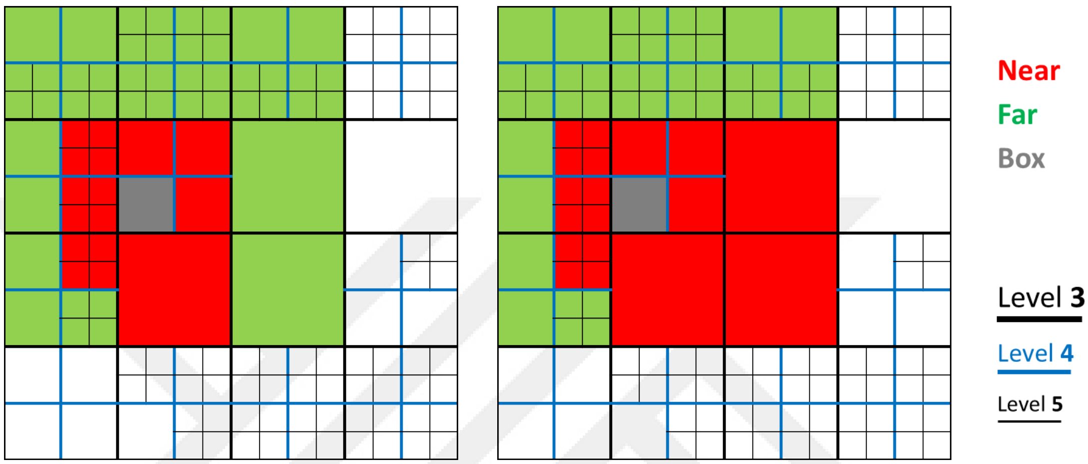


Figure 5.4: A typical IL tree structure. Note that the height of the boxes are not in scale for the second and third levels (to save space).

Imagining the tree structure based on these new definitions, one can easily deduce that the truncated boxes are actually new "leaves" but with distributed positions across the whole tree structure. Using this visual concept, some or many of the branches are not complete ones, leading us to the novel IL tree structure, and hence, "Incomplete-Leaf" MLFMA (IL-MLFMA). Fig. 5.4 illustrates a typical four-level IL tree structure for a thin object (without taking into account the correct height of the boxes), where OCBs, TBs, and PBs can clearly be seen.

5.2.3 Two Possible Scenarios

Since an IL tree structure is different than a traditional one used by the conventional MLFMA, near-box and far-box concepts are required to be redefined. In addition, some ground rules must be constructed to determine the near and far boxes from the accuracy and efficiency point of views. Hence, possible two different 2D scenarios to determine the near and far boxes in an IL tree structure are illustrated in Fig. 5.5, where the small gray box at the middle is the testing box and the surrounding red (dark grey) and green (light gray) boxes denote near and far basis boxes, respectively. The borders for the third to fifth levels are shown in the legend.



(a) (b) (c)

Figure 5.5: Two different scenarios to define near and far boxes, shown by red (dark grey) and green (light gray) boxes, respectively. (a) Near boxes based on touching boxes, (b) near boxes based on either touching boxes or larger near boxes at higher levels. Far boxes are defined as the rest of the boxes inside a proper cuboid of size $6 \times 6 \times 6$ at the corresponding level.

In the scenario shown in Fig. 5.5(a) only touching boxes with equal or larger sizes are labeled as near boxes. However, when the second scenario shown in Fig. 5.5(b) is considered, unclustered near boxes of all the upper-level parents, if exist, are added to those boxes found in the first scenario. For example, additional two level-3 large (i.e., unclustered) boxes should also be considered as near boxes for the parent of the shaded box. Then, in both scenarios, far boxes are determined by the rest of the boxes that reside inside a proper $6 \times 6 \times 6$ cuboid of the same size (of the gray box). A detailed comparison between these two scenarios reveals that, although the second scenario is more complicated to implement, it is more superior than the first one due to the following three reasons:

1. For the scenario shown in Fig. 5.5(a), consecutive interpolation and anterpolation operations are needed for both aggregation and disaggregation processes within MVM routines. The reason is that, there are a few larger,

some equal, and some other smaller far boxes for which, the far-field interactions should be calculated using FMM. Consequently, for these boxes at the very same level (i.e., level 4 refering to Fig. 5.5) one needs an anterpolation, a direct calculation, and an interpolation, respectively, each followed by a proper translation to carry out the far-field interactions correctly. These extra computations based on FMM (and not MoM) bring in undesired additional errors. On the other hand, using different-sized boxes and the one-buffer-box scheme pushes the limits of the addition theorem, if the buffer box is smaller than any of them. It should be emphasized that these interpolation/anterpolation operations bring additional challenges in paralellization.

2. In the conventional FMM or MLFMA, due to the uniform size of the leaf-level boxes, x, y, and z components of translation vectors are integer multiples of the leaf-level box size. Assume that the triplet (m, n, p) represents the multiples and a is the smallest box size at the very last level. Obviously any translation vector can be written as $\bar{\boldsymbol{W}} = a(m\hat{x} + n\hat{y} + p\hat{z})$, where $0 \leq m, n, p \leq 3$. Once the translation vector for a triplet is calculated, it can be reused everywhere with the same relative vector between two far boxes. Hence, there is no need to calculate it for each pair of far interactions. However, in the scenario given in (a), as mentioned in the first item (above), we may have larger boxes as well as smaller boxes at the same level (also from upper and/or lower levels) to be considered as far boxes. Clearly, far boxes may be anywhere in the 3D space and the three components of the translation vectors are not integer multiples of a. Because the size of the smaller boxes are among $a/2, a/4, a/8$ and smaller downward to the last level, and at the same time larger boxes are among $2a, 4a, 8a$ and wherever it takes upward. Hence, we may have fractional multiples of translation vectors, referenced to the size of the boxes at a given level. This is not efficient, since the number of distinct translation vectors would be at the same order of the number of far-field interactions. Note that this degrades the overall performance, but still doable.

3. For the scenario shown in Fig. 5.5(b), larger boxes are treated via MoM

computations leading to higher accuracy without any of above mentioned issues.

Two important issues should also be noted at this point.

1. Although the scenario shown in Fig. 5.5(b) has more near-field interactions than the scenario in Fig. 5.5(a) (so that it improves the accuracy but leads to a higher computational cost), the total number of near-field interactions are controllable [and in fact $\mathcal{O}(N)$] as there is no practical limit regarding the needed levels. This will be addressed in the numerical results, where we discuss the complexity of the proposed algorithm.

2. The total number of interacting boxes are always less than the conventional MLFMA due to the pruned boxes that were supposed to branch from TBs across the levels.

5.2.4 Near, Pseudo-Near, and Far Boxes

In many multiscale problems that have to be meshed nonuniformly with large multiscale factors, one can see some special boxes that belong to upper level parents, and are located farther than some of the far boxes at the same level based on the scenario shown in Fig. 5.5(b). These boxes, here after to be called "pseudo-near" boxes, are defined as non-touching TBs located at any upper level that have the following three properties: (a) They are larger than the box of interest (the gray box). (b) They can be classified as touching near boxes for the parent or any of the grandparents of the box of interest. (c) They cannot be smaller than the size of the parent or the grandparent to be compared in (b). Such a definition justifies its appellation and helps us to distinguish these boxes from the real near boxes that touch each other. Note that, touching boxes may be smaller, equal, or larger than the testing box, but pseudo-near boxes are always larger, because they usually contain larger basis functions.

The importance of this pseudo-near concept can be displayed best by once

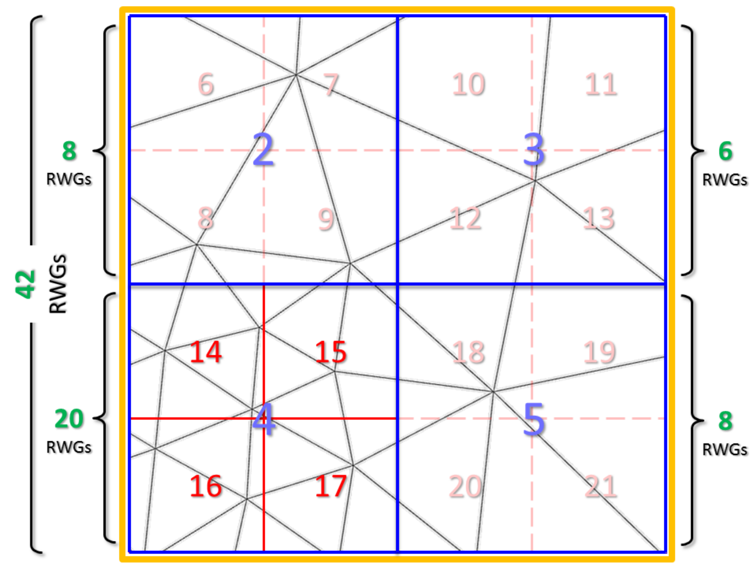


Figure 5.6: Applying IL tree clustering on the multiscale mesh of Fig. 5.1. The numbers given next to the curly braces show RWG populations of the corresponding boxes. Only the largest box and the 4th box are divided into four smaller boxes.

again considering Fig. 5.1(a). By clustering the same mesh using the proposed IL tree structure for a given population threshold (e.g. 20), Fig. 5.6 is obtained, where the numbers near the curly braces show the RWG populations. Only the largest box and the 4th box are divided into four smaller boxes, namely, 2 to 5 and 14 to 17, respectively, since their populations are larger than or equal to the given threshold. Considering the small box labeled as 17 and applying the aforementioned ground rules used to determine the near and far boxes, apart from the trivial near boxes 14, 15, 16, and 5, there are two larger pseudo-near boxes labeled as 2 and 3. When the large RWG functions extended into the 2nd and 3rd boxes are considered, most of the RWG functions inside the 17th box are connected to them justifying our rule that all RWG functions inside a pseudo-near box of a testing box, should indeed have near-field interactions with the RWG functions inside the testing box regardless of the distance between the testing box and the pseudo-near box.

The final reason about why we have to treat the pseudo-near boxes using MoM computations is that, they usually contain larger (and sometimes the largest) basis functions all over the surface of the object. As the basis function becomes larger, its radiation intensifies and creates sharper field variations at rather farther distances (and may be at the other side of the object). This means that assuming a slow variation and using a single point integration to calculate the interactions (as is the case for any far-field interaction in the context of FMM) is not valid and accurate anymore.

It is very important to emphasize that, such a treatment, i.e., dealing with some of the far boxes as if they are near boxes, and hence, computing MoM-based interactions instead of FMM-based interactions, do not imperil the robustness and reliability of the proposed method in comparison to the conventional MLFMA. As a corollary fact, we can strongly claim that the robustness and reliability of the proposed method are inherited from the conventional MLFMA.

In the upcoming sections, it is assumed that we know the total number and exact positions of non-empty boxes, as well as the corresponding RWG contents, total number of levels satisfying our population criteria, detailed population statistics at all levels, and finally the full list of far-field and near-field boxes based upon the conventional MLFMA definitions [66].

5.2.5 Construction of a List of Near and Pseudo-Near Boxes

Construction of a list of near-field interactions containing all of the near and pseudo-near boxes of a given testing box is an important aspect of the proposed method. To develop an effective and flexible algorithm, it is enough to carefully use the full properties of what we have defined so far. Interestingly, there is no need to consider any PB or OCB. No need for any PB, because the parent or one of the grand parents of any PB is a TB, which is already considered somewhere at a level above the current one. Also there is no need to consider any OCB, because of a simple fact, which reads "if two boxes are near or pseudo-near to each other for any reason, this relationship is mutual, regardless of their relative sizes". Consequently, if inside any of the levels, an OCB is near or pseudo-near to any given TB, then all the smaller sub boxes inside that OCB are also considered as the pseudo-near boxes with respect to the given TB.

From the programming point of view, this simply means that, the very same TB (which is larger and belongs to one of the higher levels) should be added to the so called "near list" of all the smaller sub boxes (which reside inside the OCB and belong to some lower levels) and the similar operation must be repeated across all the levels from top to bottom. This eliminates the need to find out which one of the larger boxes at the higher levels are near or pseudo-near to a small box located at a lower level. To fulfill this claim, we wrote a MATLAB function, or namely "ContentBoxes", that is capable of traversing the traditional tree structure in order to find all the truncated contents (i.e., TBs) inside a given OCB. This simply means that, "ContentBoxes" can find the leaf boxes inside a given branch of the tree.

The pseudo-code shown in Algorithm 1 is used to construct the near list for each level. Note that a positive box number is used for an OCB and a negative box number is used for a TB. Hence, the '−' sign used in lines 6 and 11 and the '+' sign used in the 8th line are important, since they facilitate discriminating OCBs from TBs within the list. By putting all together, it is possible to develop a proper algorithm capable of filling the near-field sparse matrix by computing the corresponding RWG interactions for our new tree structure. The algorithm is briefly illustrated in Algorithm 1 by means of pseudo-codes. At a first glance, it is clear that it consists of six successive **for** loops, which means that it takes a long time to be computed. To accelerate this important issue, one has to vectorize the codes as much as possible and use the high-speed compiled codes (known as **M**ATLAB-**EX**ecutable or **MEX** codes) to speed up the whole process. This was exactly why we were trying to develop some practical techniques capable of handling sparse matrices inside the MEX codes. It is worth mentioning that the resulting "near list" is actually a highly-sparse matrix. This sparsity completely depends on the population statistic, operating frequency and some other geometrical details, but with a limit for the maximum number of near boxes commonly known as 27.

Algorithm 1: Construction of IL Near List

```
Input: given level, traditional near list
Output: IL.NearList
TB.List := List of TBs within the given level
foreach TestingBox ∈ TB.List do
    Trad.NearList := traditional near list for TestingBox
    foreach BasisBox ∈ Trad.NearList do
        if BasisBox ∈ TBs then
            put −BasisBox into IL.NearList of TestingBox
        else if BasisBox ∈ OCBs then
            put +BasisBox into IL.NearList of TestingBox
            Trunc.SubBoxes := ContentBoxes (BasisBox)
            foreach SubBox ∈ Trunc.SubBoxes do
                put −TestingBox into IL.NearList of SubBox
            end
        end
    end
end
```

For example, a sphere of radius $R = 700$ mm, meshed with an average size of 5 mm under a population threshold of 50 RWG/box at $f = 500$ MHz, needs 100 K boxes to be processed. Total required memory to store the whole matrix is no more than 22 MB, which is negligible. But, considering the fact that we need few dozens of different indices to keep the needed data, the memory easily exceeds 200 MB even for such a small example with an electrical length of approximately λ. Obviously, for larger objects, we need tens of GBs of memory, and therefore, we have to take care of these wasted memories regarding highly-sparse matrices. Later on in this dissertation, we will cover this issue, i.e., how to squeeze and compress such sort of sparse matrices in order to efficiently use the memory without slowing down the computations.

5.2.6 Construction of a List of Far Boxes

Another important aspect of the proposed method is to construct a list of far-field interactions for a given testing box, where the same rules used in the conventional MLFMA in order to determine the boundaries of the far boxes are deployed. To create a so called "far list" for each level and based upon the existing information obtained from the traditional tree structure, we invoke the same policy regarding negative and positive box numbers. One more time, we have to emphasize that there is no need to deal with any pruned box, neither as a box to be processed, nor as a far box to be registered, exactly based on the similar reasons we already discussed for the "near list".

The main difference between the "far list" and the "near list" algorithm stems from the fact that, none of the larger boxes (belonging to the higher levels) in the neighborhood of a smaller box (belonging to a lower level) can act as a far-field box. Note that this neighborhood approach is precisely defined within the context of conventional MLFMA. To clarify this important rule, we should carefully review Fig. 1.1(a) one more time, even though it is a 2D example. Considering Level 5, which is a complete demonstration of near and far boxes, we realize that only dark blue (gray) boxes can be considered as far-field boxes to the red (dark gray) one. The boundary is governed by a simple rule: Move at least 2 and at most 3 boxes in all tree directions and stop at the boundaries of either the object itself or one of the parent boxes around, whichever comes first. Using this approach, there will be no chance for any larger box in the vicinity to act as a far box.

A corollary fact is that, we can reuse our "near list" algorithm by considering two facts: (a) Unlike the "near list", we should process the OCBs in addition to the TBs because we need to aggregate the radiation of the smaller sub boxes inside OCBs. (b) We must start from the third level, where the far-field interactions make sense. Putting all together, we can write another compact algorithm to fill-in the needed "far list" as shown in Algorithm 2.

Algorithm 2: Construction of IL Far List

```
Input: given level, traditional far list
Output: IL.FarList
Joint.List := List of TBs ∪ OCBs within the given level
foreach TestingBox ∈ Joint.List do
    Trad.FarList := traditional far list for TestingBox
    foreach BasisBox ∈ Trad.FarList do
        if BasisBox ∈ TBs then
            put −BasisBox into IL.FarList of TestingBox
        else if BasisBox ∈ OCBs then
            put +BasisBox into IL.FarList of TestingBox
        end
    end
end
```

Again, we have to be careful about the "far list" from the memory point of view. A simple computation based on the given rules (for far-field boundaries) reveals that the number of boxes can reach up to $(2(\text{backward}) + 3(\text{forward}) + 1(\text{itself}))^3 - 3^3(\text{near}) = 189$ boxes. By considering our previous example with approximately 100 K boxes, we need at least 150 MB of memory just to hold the required "far list" data. This is another evidence for the necessity of using sparse matrices to optimize the memory structure for large simulations.

5.3 Pictorial Examples

Based upon the 2D equivalent cross section discussed in Section 5.2.2, one can apply the newly developed algorithms to find near, pseudo-near, and far boxes for a given testing box in order to construct the IL tree structure. Here, the target is to find two lists of boxes: a "near list" containing a list of boxes in which we have to carry out direct near-field computations and a "far list", as a list of boxes in which we should carry out the far-field computations. An important concept to bear in mind is that, the "near list" algorithm works based on a hierarchy implied

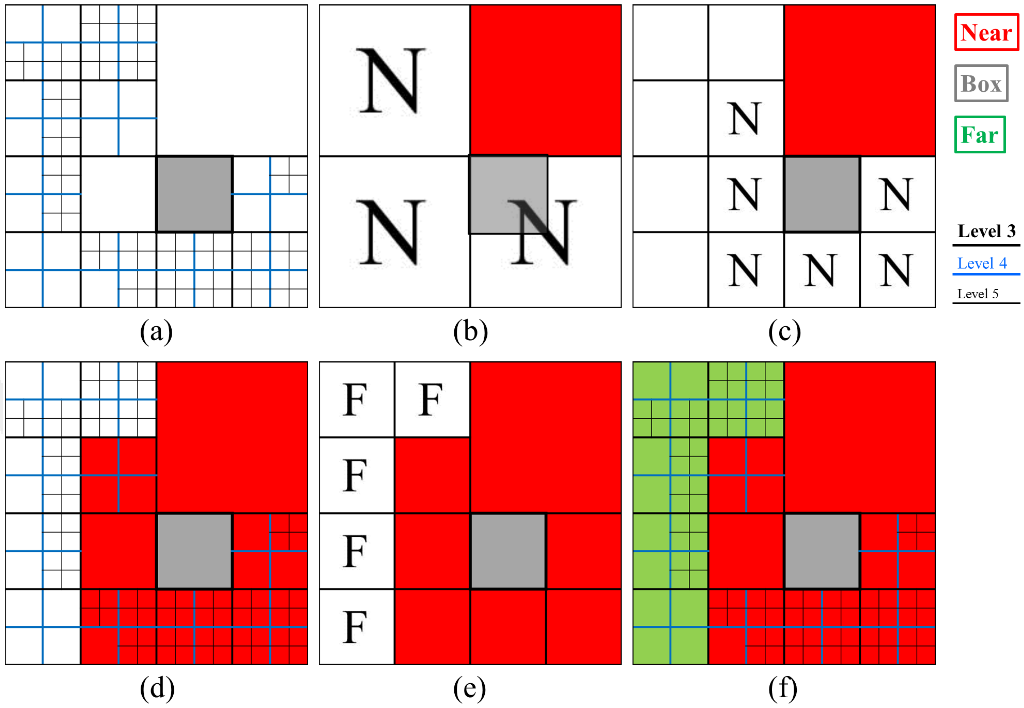


Figure 5.7: A step-by-step pictorial approach to determine both near and far boxes for a given testing box inside a typical IL tree structure. (a) The box of interest in gray. (b) The first near box. (c) A group of next level near boxes. (d) All near boxes together regardless of size or level. (e) Far boxes at the same level. (f) All far (and near) boxes regardless of size or level. In all of these pictures, higher level boundaries always coincide with the lower level boundaries and consequently cover their designated color or thickness. Note that near and far boxes are shown in dark gray (red) and light gray (green), respectively.

by the 9th line of the Algorithm 1, which means that one has to first process the larger boxes at the upper levels, then process the next level smaller boxes and so on. Note that this is not necessary for the far list algorithm. Getting back to the figure, our observations are as follow. To start with, we consider a simple case shown in Fig. 5.7(a), where the given testing box is in gray.

1. The only TB in level 2 is shown in red (dark gray) in Fig. 5.7(b). All three near boxes of this TB in this level, which are OCBs, are labeled by the letter "N". However, the gray box (i.e., the testing box) is the sub box of only one of them. Hence, the remaining OCBs are not our interest and the TB is the first member of our near list.

2. Considering level 3, Fig. 5.7(a) also shows that there are only two TBs in

this level, which are near to each other. One of them is the testing box, and the other one is the second member of the near list. However, referring to Fig. 5.7(c), boxes labeled with “N” are also the members of the near list at this level. Fig. 5.7(d) illustrates all the near boxes including level-wise details.

3. For the far boxes, we must move at least two and at most three boxes from the given test box. However, we also need to stop at the object boundaries. Fig. 5.7(e) and Fig. 5.7(f) show all the far boxes labeled with the letter “F” and the final structure including all details, respectively.

A more complicated and peculiar structure, shown in Fig. 5.8(a), is considered next, where the same strategy is followed.

1. Starting from level 2, the only TB box is shown in red (dark gray) in Fig. 5.8(b). As before, all three near boxes to this TB of this level are labeled by “N”, where the test box is inside one of them. Similar to the first example, the other boxes labeled with “N” are OCBs, and are not considered at this level. Hence, the large red (dark gray) box is the first member for the near list as a pseudo-near box.

2. Referring back to Fig. 5.8(a), we can see three level-3 TBs labeled as 1 to 3 in Fig. 5.8(c). Neither box 1 nor box 2, with opposite cross-hatched shaded regions, may have near-field interactions with the small gray box. Proceeding to Fig. 5.8(d), the box labeled as 3 is the only level-3 box having near-field interaction with the given test box, thus, it will be the member of the near list again as a pseudo-near box.

3. There is only one level-4 TB near to the test box, as shown in Fig. 5.8(e). In a similar fashion to the previous cases, this TB is also a member of the near-list as a near box. Considering the level of the test box, the touching seven boxes are the members of the near list. Fig. 5.8(f) shows all near and pseudo-near boxes.

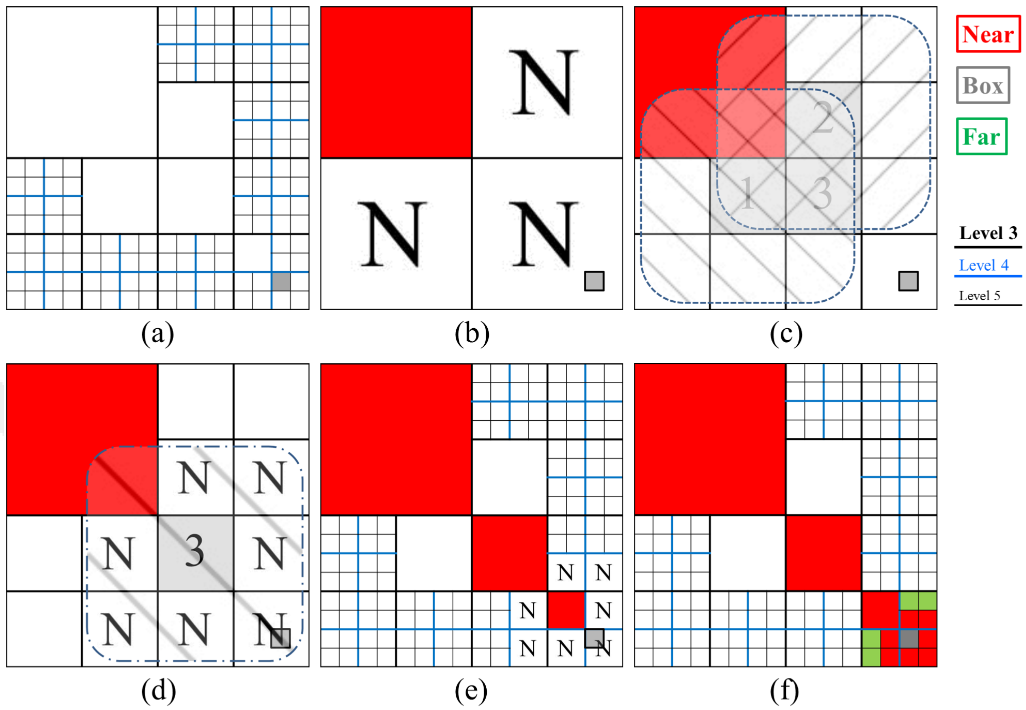


Figure 5.8: Another pictorial example to determine the near, pseudo-near, and far boxes inside a symmetric IL tree structure. (a) The box of interest in gray. (b) The first pseudo-near box. (c) None of two boxes labeled by 1 and 2 can cover the gray box. (d) Only 3rd box is pseudo-near to gray box. (e) The same procedure but with the next level smaller boxes. (f) All the near, pseudo-near, and far boxes in one shot.

4. To find far boxes, if we move down or right, object boundary comes first, however, if we move up or left, the level-4 box boundaries come first. Hence, after excluding all near boxes, we end up with the indicated far boxes as also illustrated in Fig. 5.8(f).

To clarify the philosophy behind this new concept, lets review Fig. 5.1(a) from another perspective, but somehow related to the multiscale issues we already discussed at the beginning of this chapter. Considering the small red box labeled 20 and following the same strategy to find the “near list”, we find out that beside trivial ones, i.e., 15, 17, 18, 19, and 21, we have two farther but larger pseudo-near boxes labeled as 2 and 3. Investigating the large RWG functions inside 2 and 3, one can easily realize that they are very close to the RWG function located inside box 20. Actually this situation is very similar to what is illustrated in part (b) of the same figure. The same reasoning goes to smaller RWG functions inside box

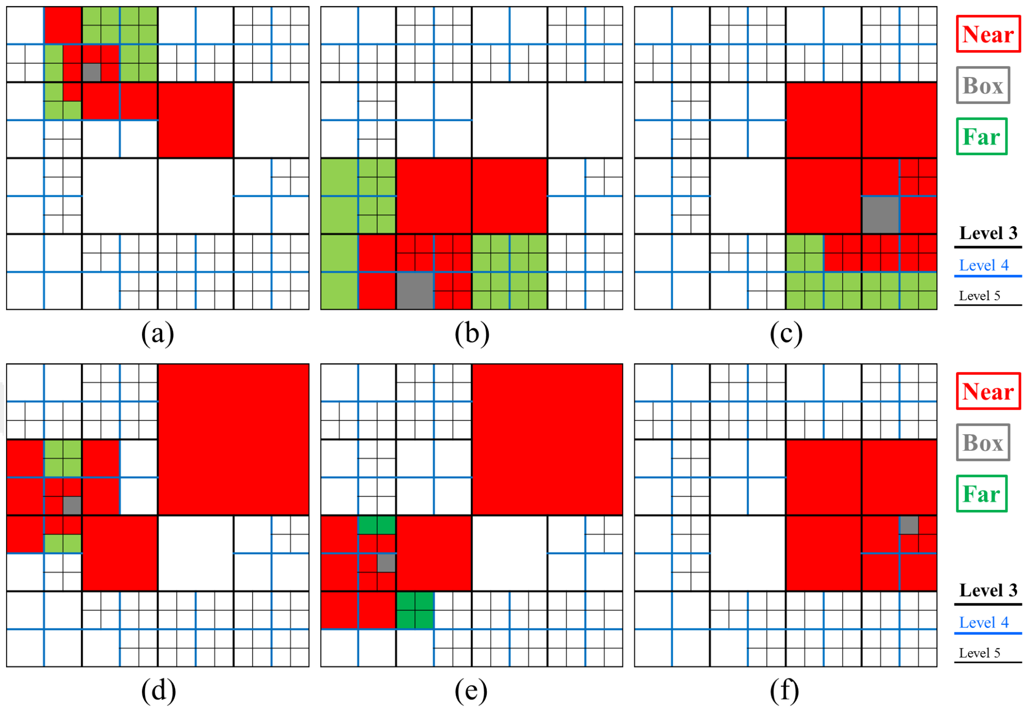


Figure 5.9: For a nonuniform clustering, six examples illustrating the near, pseudo-near, and far basis boxes regarding a given testing box. In each sub-figure, different testing box is considered. Note that in (f), there is no far box due to some large and very large nearby basis boxes.

16 where all three larger boxes 2, 3 and 5 are considered to be pseudo-near boxes. Generalizing to the higher levels and also larger boxes, we can always find such sort of faraway pseudo-near boxes.

More complicated examples are shown in Fig. 5.9 without any step-by-step details about how to apply the ground rules and the related strategies. Note that in Fig. 5.9(f) there is no far box regarding the given gray box, mainly because of the large and very large nearby basis boxes. Obviously this is not an abnormal condition due to the fact that, after aggregation from level 5 (the gray box) up to level 4 (the blue boxes) or even higher, we will have lots of far-field boxes to test on.

5.4 Construction of Near-field Sparse Matrix

It is well-known that the impedance matrix regarding near interactions is highly sparse. Even though we can use sparse facilities embedded in MATLAB (or any other programming language that we use) to store data, we rarely use them. This is simply because sparse-oriented commands in MATLAB (or scripts in any other language) are generally slow since they have to track the non-zero elements in every operation, which takes some time.

On the other hand and regardless of this retardation, MATLAB does not support the use of sparse matrices inside the MEX codes. To circumvent this issue, one may use one of the following methods, which are somehow similar to each other:

1. The first method is suitable for element-wise random access read and write. In this method, all sparse data are embedded inside a long column (or row) vector, and then, by means of another wisely prepared index, casting as a kind of address pointer, one can compute and then access the exact positions of the matrix elements. Near interaction between ith basis function and jth testing function, where i and j belong to a predetermined set of n RWG functions, is a good example of this group. In this section (and later on inside the near-field MVM routines), we use this method to compute and store the near-field interactions as a sparse matrix.

2. Second method is suitable for block-wise read and write. In this method, the whole sparse matrix is transformed into a self-containing long row or column vector, whichever best suites our needs. It is self-containing, because the needed address pointers in order to have access to nth block of the stored data are already embedded inside the vector. Those near and far indices that are already discussed in Section 5.2, also some other indices used in our program, are typical examples of this kind of sparse matrices. We use this method for the far-field interactions later.

A careful benchmarking reveals that the second method is faster than the first one, not only because of accessing a block of data in one shot, but rather due to the fact that one needs to compute the address pointer using a dedicated formula and an index prepared beforehand, and then use that pointer to access somewhere inside the data vector. For both methods, two important key points should be emphasized.

- First method cannot be replaced by the second one due to its versatility, and hence, we need to deploy both of them.
- The secondary matrix used as an index in the first method is also a kind of sparse matrix. Therefore, in order to save more memory, we adopted the use of the second method to store the required sparse index of the first method.

Utilizing the above mentioned methods to store either the near-field sparse matrix or the corresponding near indices, one has to consider two important facts to compute the matrix based upon existing near list. First, the testing/basis functions only exist inside the TBs. Second, within the near list of a given testing box, there may be a combination of TBs together with some other OCBs. Each one of the OCBs may also contain some other OCBs in a recursive manner and this may extend down to the very last level. Combining all together, Algorithm 3 briefly describes the steps needed to perform this task.

Regarding these pseudo-codes, two important concepts should be clarified. First, although it is possible to use a recursive function in MATLAB to extract and add the needed TBs within all OCBs, a simple loop ($2 \mapsto 8$) is preferred due to the fact that in MATLAB, calling a function is slower than updating a list. Second, all extracted box numbers are negative, hence, we need a minus sign before using them as a basis box ($\mapsto 10$).

Finally, it is important to emphasize that in multiscale meshes, due to possible small distances between the basis and testing functions, the radiated and received

Algorithm 3: Near-Field Sparse Matrix

```
Input: IL.NearList
Output: Near-Field Sparse Matrix
foreach TestingBox ∈ All of the TBs do
    Mixed.List := List of near and pseudo-near boxes regarding TestingBox
    using IL.NearList
    Final.NearList := Negative elements of Mixed.List
    Overcrowded.List := Positive elements of Mixed.List
    foreach TmpBox ∈ Overcrowded.List do
        Content.List := List of all TBs inside TmpBox
        Final.NearList := Final.NearList ∪ Content.List
    end
    TestingBox△s := List of △s inside TestingBox
    foreach BasisBox ∈ -Final.NearList do
        BasisBox△s := List of △s inside BasisBox
        foreach testing△ ∈ TestingBox△s do
            Compute integration points over testing△
            foreach basis△ ∈ BasisBox△s do
                Compute integration points over basis△
                forall the valid testing functions do
                    forall the valid basis functions do
                        Invoke singularity extraction routines to compute
                        formulation dependent integrals (e.g. MFIE)
                        Update Near-Field Sparse Matrix
                    end
                end
            end
        end
    end
end
```

fields are not smooth to be handled by a single point integration. Hence, the order of numerical integration must be increased for better accuracy.

5.5 Far-field Interactions

Far-field calculation routine is the most challenging part of the IL-MLFMA. It is challenging, because apart from the implementation difficulties due to variable box sizes inside a level and/or multiple levels, two different far-field algorithms, one associated with the corrected/scaled MLFMA (as discussed in Chapter 4) for smaller boxes to avoid the low-frequency breakdown, and the other associated with the traditional MLFMA (as summarized in chapter 3), are deployed simultaneously, in order to successfully calculate the far-field interactions across the levels. Frequency, population threshold, multiscale factor, number of levels or equivalently smallest box size are among the key factors that govern the border in which one has to switch between these algorithms. Both of them consist of three separate parts, i.e, aggregation, translation, and disaggregation, as briefly discussed in Section 3.2.2. Translation parts of both algorithms can be merged as a single routine making the program more modular. The aggregation and disaggregation parts are computationally different due to dissimilar intrinsic natures of the computations to be carried out.

The relative positions of the existing far-field algorithms concerning the box sizes are illustrated in Fig. 5.10. It is seen that, as we move from larger boxes to smaller ones (or equivalently from the higher levels to the lower levels), we have to pass through the border of "boxsize $\leq \lambda/4$". This fictitious border discriminates the "Traditional" MLFMA from the "Scaled/Corrected" MLFMA. A combination is also a possibility, where some of the boxes are larger than $\lambda/4$ and the rest of them are smaller than $\lambda/4$. For these cases, the traditional MLFMA routines either fail or give us inaccurate or completely wrong results. Interestingly, scaled/corrected routines can handle these problems rather easily by using a unity scale factor or $s = 1$, with a slightly higher complexity ($\mathcal{O}(N \log^2 N)$) as addressed in Section 4.4.2.4.

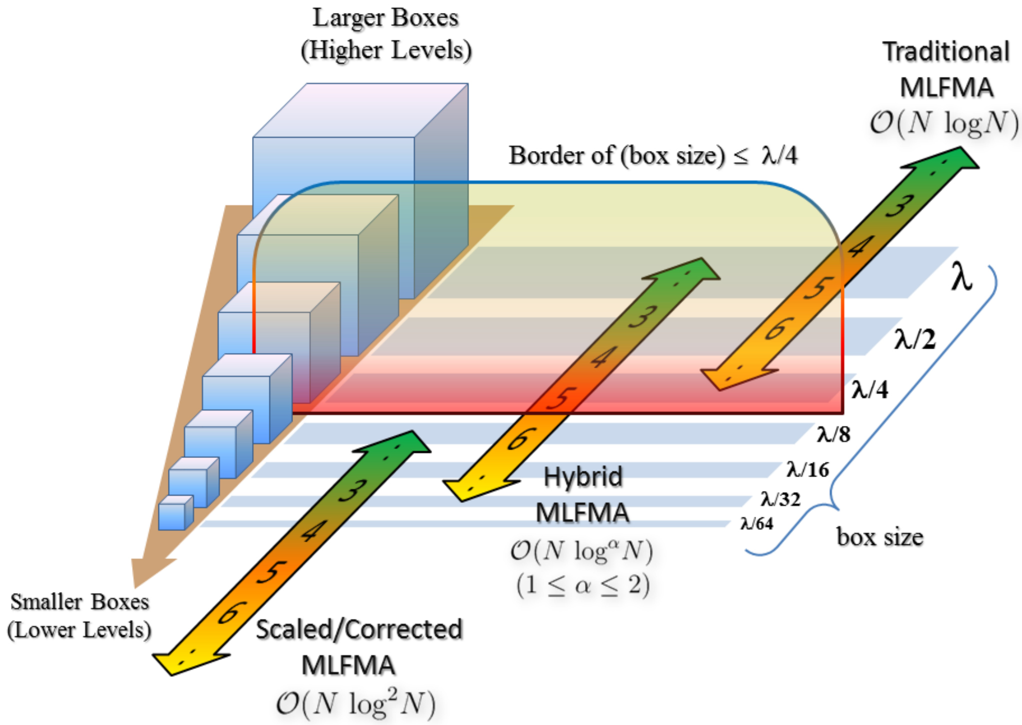


Figure 5.10: Two different types of MLFMA routines and their relative positions with respect to various box sizes. The border between their operational area is determined with box size $\leq \lambda/4$ as shown by a transparent wall. Complexities are given under each method.

At this point it is worth to emphasize the followings.

- In the literature, different standards are used to define this border, such as $\lambda/10$ or even lower. However, as an undeniable fact, there exists such a border and practically limits the operational area of the traditional MLFMA. This is not true for the Corrected/Scaled MLFMA, despite the fact that we will present some cases where this method also fails.

- Both of those methods, regardless of accuracy, need to deploy fixed-size uniform boxes in each and every level. This simply means that we cannot directly use them to handle variable sized boxes, because of quite different near and far definitions we should use.

- For Hybrid-MLFMA, i.e., a combination of traditional MLFMA for the higher levels and Corrected/Scaled MLFMA for the lower levels, the overall complexity must be $\mathcal{O}(N \log^{\alpha} N)$, where $(1 \leq \alpha \leq 2)$ ($\alpha = 2$ for the low-frequency regime [34]), totally dependent on the frequency and the leaf-level box sizes.

After a brief explanation about the translation routine, we proceed to develop new aggregation and disaggregation algorithms to finish this section.

5.5.1 A Review

As we know, the corrected N-MFIE formulation is based on the normally tested $\mathcal{K}$ operator. Details of this formulation are discussed in Section 4.5, but they are repeated here for convenience. We have

$$\boldsymbol{K}^{N}_{\text{Cor}}[m,n] = \left(\frac{ik}{4\pi}\right)^2 \int d^2\hat{\boldsymbol{k}}\, \boldsymbol{R}^{\mathcal{K},\text{N}}_{m,\text{Cor}}(\boldsymbol{k},\boldsymbol{r}_m)\, \tilde{\alpha}_s(k,\hat{\boldsymbol{k}},\boldsymbol{w},s) \cdot \boldsymbol{S}^{\mathcal{K},\text{N}}_{n,\text{Cor}}(\boldsymbol{k},\boldsymbol{r}_n), \quad (5.1)$$

where, $\tilde{\alpha}_s$, $\boldsymbol{S}$, and $\boldsymbol{R}$ are the translation, radiation, and receiving functions, respectively, defined as

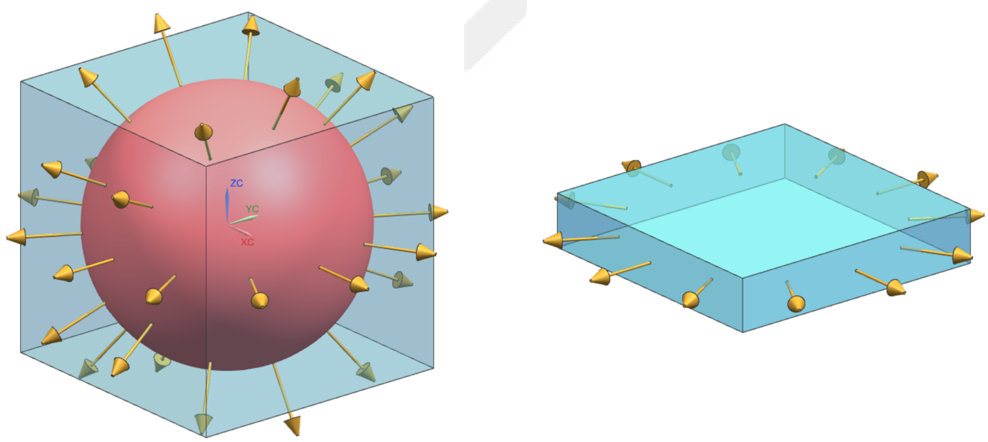
$$\boldsymbol{S}^{\mathcal{K},\text{N}}_{n,\text{Cor}}(\boldsymbol{k},\boldsymbol{r}_n) = \int d\boldsymbol{r'} e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}_{agg}}{s}} \left(1 + q\,|\boldsymbol{v}_{agg}|^2\right) \boldsymbol{b}_n(\boldsymbol{r'}), \quad (5.2)$$

$$\tilde{\alpha}_s(k,\hat{\boldsymbol{k}},\boldsymbol{w},s) = \sum_{t=0}^{\infty} (i)^t(2t+1)\left(s^t\right) h^{(1)}_t(kw) P_t(\hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{w}}), \quad (5.3)$$

$$\boldsymbol{R}^{\mathcal{K},\text{N}}_{m,\text{Cor}}(\boldsymbol{k},\boldsymbol{r}_m) = \int d\boldsymbol{r}\, e^{ik\hat{\boldsymbol{k}}\cdot\frac{\boldsymbol{v}_{dis}}{s}} \Big[\frac{\hat{\boldsymbol{k}}}{s}\left(1 + q\,|\boldsymbol{v}_{dis}|^2\right) - i\frac{2q}{k}\,\boldsymbol{v}_{dis}\Big] \times \Big[\hat{\boldsymbol{n}} \times \boldsymbol{t}_m(\boldsymbol{r})\Big]. \quad (5.4)$$

To carry out (5.1), integration over a unit sphere is needed. As we know, $\int d^2\hat{\boldsymbol{k}} = \iint sin(\theta) d\theta d\phi$, hence, the integrand should be calculated and stored for each and every separate direction, and then added up to the total sum. Despite the fact that we are going to deploy both of the above mentioned far-field algorithms, each with different number of $\hat{\boldsymbol{k}}-$space samples for the higher and lower levels, both of them obey the same rules to determine the optimum number of samples. Briefly, as discussed in Section 3.1.3, it is sufficient to use $(\tau + 1)$ and $(2\tau + 2)$ samples in θ and ϕ directions, respectively, where both of them are merely functions of the truncation number τ. Note that the optimal value for τ (based on 3.11) is only accurate and useful for the conventional MLFMA. In other words, for corrected/scaled MLFMA we have to use some other tabulated optimal values that are collected from a wide range of numerical simulations.

Considering the fact that we are going to extensively use the concept of $\hat{\boldsymbol{k}}-$space in upcoming sections, it is worth to have a 3D visualization regarding this space as graphically shown in Fig. 5.11(a). In this figure, the transparent blue (light gray) box represents a sample basis box containing some limited numbers of basis functions without considering its relative size with respect to a unit sphere, which is drawn in red (dark gray). Note that the number of separate $\hat{\boldsymbol{k}}$



(a) (b)

Figure 5.11: A 3D illustration of $\hat{\boldsymbol{k}}-$space around a typical basis box. In part (a) if $\tau = 5$, then we have $\theta \times \phi = (\tau + 1) \times 2(\tau + 1) = 6 \times 12$ separate directions. (a) A typical basis box with a hypothetical unit sphere containing $\hat{\boldsymbol{k}}$ directions. (b) A flat 2D equivalent version of the 3D $\hat{\boldsymbol{k}}-$space.

directions in the figure has been chosen to be consistent with the optimal sampling scheme described above. Due to the fact that it is not possible to include all features of a 3D structure inside a 2D picture, we consider their 2D equivalents in subsequent sections of this dissertation. For example, Fig. 5.11(b) is considered as a 2D representative of its 3D version. The last thing to bear in mind is that the directions shown here are dedicated for basis boxes (or functions) as they radiate

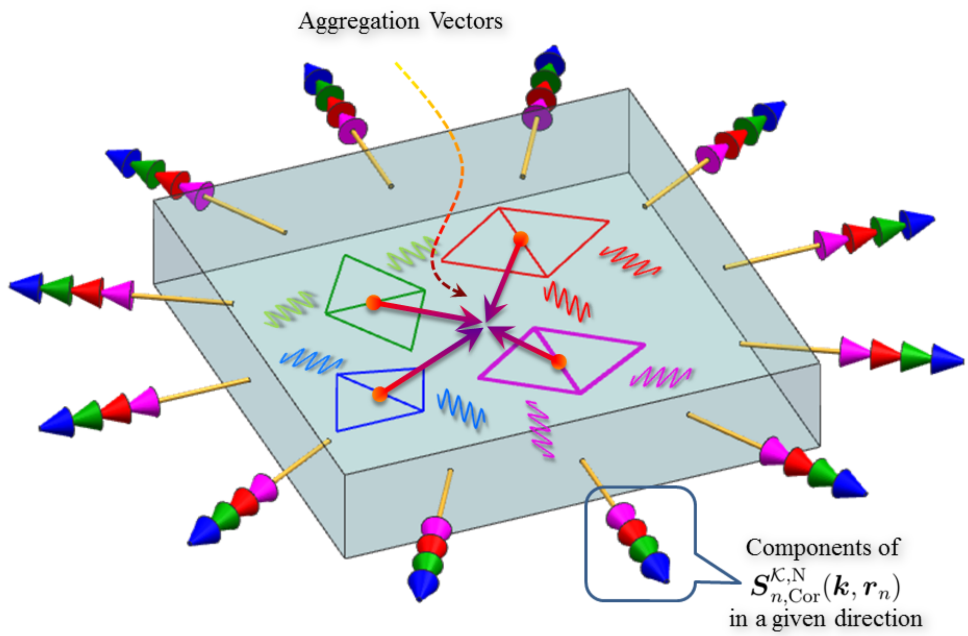


Figure 5.12: Aggregation process inside basis boxes.

outward. Obviously when we are going to deal with testing boxes (functions), we use opposite directions for the very same arrows to imply the receiving state.

5.5.2 Aggregation

By incorporating the above mentioned 2D equivalent pictures into the concept of aggregation, we can elaborate (5.2) in a much more comprehensive manner. In this equation, $\int d^2\hat{\boldsymbol{k}}$ means that we have to calculate, store, and then sum up the radiation functions ($\boldsymbol{S}^{\mathcal{K},\mathrm{N}}_{n,\mathrm{Cor}}(\boldsymbol{k},\boldsymbol{r}_n)$) of all the basis functions inside the box of interest for each and every $\hat{\boldsymbol{k}}$ vector over the dedicated $\hat{\boldsymbol{k}}-$space. Figure 5.12 is a graphical demonstration of how different components in the dedicated directions are added up. In this figure, each color represents the radiated fields of the corresponding basis functions and each direction represents a plane-wave independent of the others.

Obviously, this has to be done for all TBs as well as OCBs, but not PBs, since the content of PBs will be automatically taken into account at some higher levels. To implement this part, we have to develop another useful function (script) capable of extracting a list of RWG functions inside a given TB or OCB. This function, namely "ContentRWGs", is somehow similar to "ContentBoxes" function discussed earlier.

Considering the two algorithms discussed before, we need two kinds of aggregation routines. Even though the fundamental concept is the same for both of them, the one which belongs to the conventional MLFMA needs some extra computations to be carried out due to its hierarchical structure. As stated before, different implementations for MLFMA use different definitions to discriminate between "large" and "small' boxes. Here we will use $\lambda/4$ as a sharp boundary to switch between the two algorithms.

5.5.2.1 Aggregation in the Low-Frequency Regime

During the construction of the IL tree structure, the box size in each level must be compared against different upper and lower limits to assign a proper scale factor. These optimized scale factors are based on experimentally tabulated values. If the scale factor reads less than one, it simply means that we are dealing with a level that needs the low-frequency treatments. In this dissertation, we use the corrected version of the scaled MLFMA based on the level-wise optimization discussed in Section 4.4.2.4. In the other words, it is enough to apply the proper algorithm across the designated levels independently with no need to perform any interpolation between the successive layers. The whole algorithm is given in Algorithm 4.

5.5.2.2 Aggregation in the High-Frequency Regime

When we are dealing with larger boxes, $s = 1$ and we can reuse the aggregated values of the lower levels for their parent (or higher) levels in a hierarchical fashion. This is the result of a simple addition theorem for the exponential function which

Algorithm 4: Aggregation in the Low-Frequency Regime

Input: TBs, OCBs
Output: Aggregation Sparse Matrix
foreach level $\in$ low frequency levels **do**
    Joint.List := List of TBs $\cup$ OCBs within the level
    Compute N_θ, N_ϕ as the numbers of needed samples
    $\theta_i := \theta$ locations over unit sphere
    $\phi_j := \phi$ locations over unit sphere
    foreach BasisBox $\in$ Joint.List **do**
        foreach basis function inside BasisBox **do**
            foreach $\theta_i, i = 1 : N_\theta$ **do**
                foreach $\phi_j, j = 1 : N_\phi$ **do**
                    Compute radiation functions
                    (regarding needed formulation)
                    Update Aggregation Sparse Matrix
                end
            end
        end
    end
end

is

$$\exp(\boldsymbol{v}_{agg}) \times \exp(\boldsymbol{v}_{Agg}) = \exp(\boldsymbol{v}_{agg} + \boldsymbol{v}_{Agg}), \tag{5.5}$$

where $\boldsymbol{v}_{agg}$ and $\boldsymbol{v}_{Agg}$ denote the aggregation vectors for two consecutive levels (i.e., a given level and its parent level, respectively). Interestingly, by reusing the computations of each level at the higher levels, the overall complexity reduces from $\mathcal{O}(N \log^2 N)$ to $\mathcal{O}(N \log N)$. This can be attributed to the fact that there is no need to carry out the same computations for approximately $\log N$ higher levels.

Note that this is not true for the low-frequency regime with different values of the scale factor since

$$\exp(\frac{\boldsymbol{v}_{agg}}{s_1}) \times \exp(\frac{\boldsymbol{v}_{Agg}}{s_2}) \neq \exp(\frac{\boldsymbol{v}_{agg} + \boldsymbol{v}_{Agg}}{s_3}), \tag{5.6}$$

except for $s_1 = s_2 = s_3$ (which is the case when $s = 1$).

To implement an accurate aggregation procedure, a computational issue has to be solved, which is not evident at a first glance. The issue originates from the fact that, for the higher levels with some larger boxes, one needs a higher bandwidth to compute the radiation functions. This simply means that using (3.11) with larger values of a as the box size ends up with a larger number of samples over the unit sphere. Consequently, to use the aggregated values of a lower level (with smaller number of samples) at the higher levels (with a larger number of samples), one needs to interpolate them. In our routines, we use Lagrange interpolation for this purpose. Fig. 5.13 illustrates this procedure. In this figure, coarse samples that belong to smaller blue boxes are interpolated from 12 directions (yellow arrows) into 18 directions (silver arrows), shown as four tiny brown arrows at each new direction. After summation over all basis functions inside the box, the needed fine samples are ready to be used for the aggregation process at the higher level.

The needed algorithm for this kind of aggregation for the higher levels is almost the same as Algorithm 4 except there is an important difference. This difference comes from the fact that the OCBs need to be processed using interpolations. Hence, Algorithm 4 can be used twice; one for the TBs without interpolation, and another one for the OCBs with proper interpolation from the lower levels. Obviously we can wisely merge outer loops of both algorithms to make it faster and more modular.

5.5.3 Translation

Regarding the translation operator, we can reuse the routine of the traditional MLFMA, except one small modification. Comparing the translation function of the conventional MLFMA (3.10) with that of the scaled MLFMA (4.17), the only difference is the scale factor s^t inside the summation. Based on the detailed discussions in Section 4.2, using $s = 1$ leads us to the conventional MLFMA, and consequently, we can use the translation function of the scaled MLFMA for both of them as well. Considering the new translation function given by (4.19) for the corrected version of MLFMA, even though it seems to be a bit different,

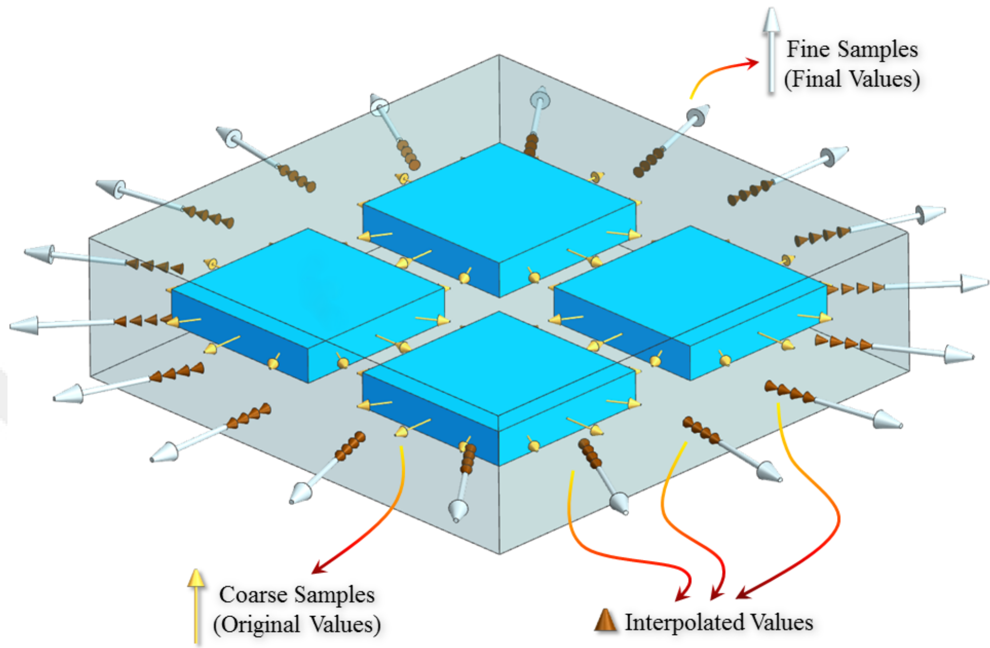


Figure 5.13: Aggregation process at the higher levels by means of interpolation.

we can reuse the same routine. This is exactly why we try to detach both of the correction factors $C_{agg}(k, s, \boldsymbol{v}_{agg})$ and $C_{dis}(k, s, \boldsymbol{v}_{dis})$ in (4.22) to use the same translation function.

As an algorithm, it is sufficient to process both of OCBs and TBs at each and every level to perform the required translations. Algorithm 5 demonstrates a versatile pseudo-code to perform this task. Note that, there is no need to change anything from the corrected/scaled version to the conventional version because of the fact that in both versions of MLFMA, all boxes in any level are exclusively either OCB or TB. Hence, all of them will be processed accordingly. It should be reminded that, based on the last item given at the definition of a TB (Table 5.1), the last level boxes of the IL-tree structure can be treated as TBs, regardless of their population. This helps us to prevent any interpolation for an OCB that may exist at the last level due to semi-automatic leveling or using the IL-tree structure to emulate the conventional MLFMA. A few remarks about this algorithm are as follows.

Algorithm 5: Translation

Input: given level

Output: Disaggregation Matrix

1 **foreach** level $\in$ designated levels **do**
2   Compute N_θ, N_ϕ as the numbers of needed samples
3   Joint.List := List of TBs $\cup$ OCBs within the level
4   **foreach** TestBox $\in$ Joint.List **do**
5   TestBox.center := Center coordinates of the TestBox
6   FarBox.List := List of far boxes concerning the TestBox
7   **foreach** BasisBox $\in$ FarBox.List **do**
8   BasisBox := |BasisBox|
9   BasisBox.center := Center coordinates of the BasisBox
10             $\boldsymbol{W}_{vec}$:= TestBox.center - BasisBox.center
11   Compute translation index($\boldsymbol{W}_{vec}$)
12   Compute translation function(translation index)
13   Compute aggregation index(BasisBox)
14   Compute disaggregation index(TestBox)
15   **foreach** $\theta_i, i = 1 : N_\theta$ **do**
16   **foreach** $\phi_j, j = 1 : N_\phi$ **do**
17   disaggregation := aggregation $\times$ translation
18   **end**
19   **end**
20   **end**
21   **end**
22 **end**

1. There are positive and negative box numbers inside the “far list” of a given test box, therefore, the use of abs($\cdot$) function is important in the 6th line of the algorithm.

2. Among the boxes within each level, there are many duplicated translation vectors that should be avoided. This is because of the fact that to compute the translation operator, only the relative position of the testing box with respect to the basis box matters. Therefore, one has to construct and save a lookup table (or index) to prevent any duplication ($9 \mapsto 10$). This is

another important routine, which is processed after the construction of the tree structure and before far-field calculations.

3. All three indices used here, i.e., aggregation, translation, and disaggregation are sparse matrices. Hence, they need their own address pointers (as discussed in Section 5.4).

5.5.4 Disaggregation

Disaggregation process is quite similar to its aggregation counter part, but in a reverse order. Clearly, we need two slightly different disaggregation routines for the low-frequency and high-frequency regimes.

5.5.4.1 Disaggregation in the Low-Frequency Regime

Fig. 5.14 illustrates the whole disaggregation process as well as the corresponding aggregation and translation operations. In this figure, pale pink and blue squares are placeholders for the near and far boxes, respectively. Among far boxes, three of them are explicitly shown to illustrate all of the required interactions in detail.

As an algorithm, it suffices to consider the fact that the levels are independent of each other and hence, both of the TBs and OCBs have the same role in this regime. It means that no matter which box is overcrowded or truncated, we perform the standard testing operation regarding the MFIE formulation.

5.5.4.2 Disaggregation in the High-Frequency Regime

Disaggregation process at the higher levels is also very similar to its counter part aggregation process except the need for anterpolation instead of interpolation. In this method, the normal testing procedure is accomplished for TBs. On the

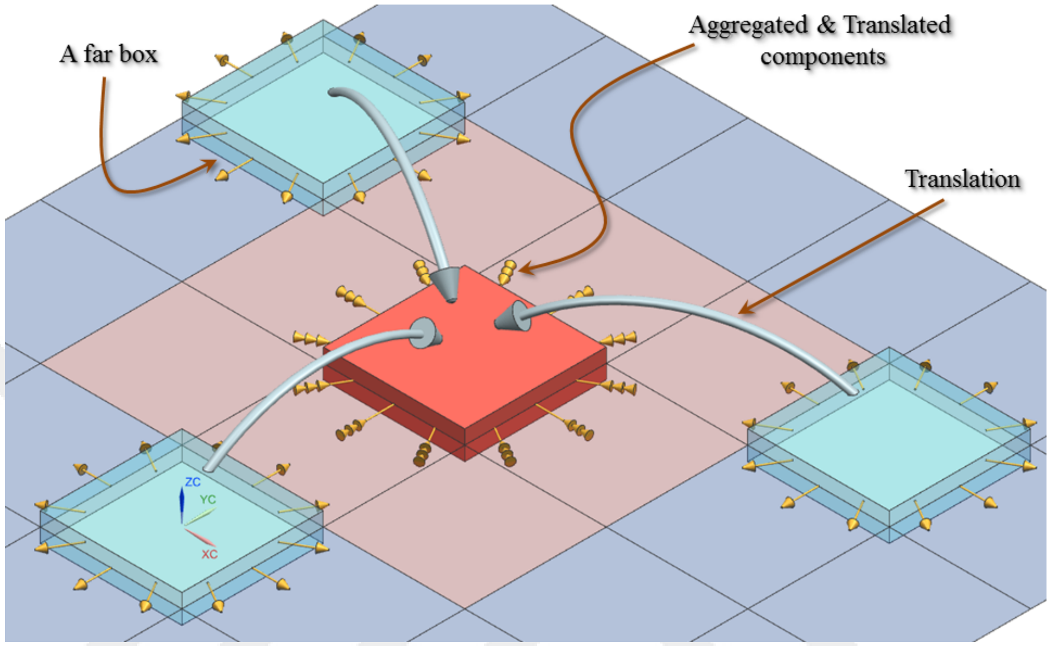


Figure 5.14: A complete illustration of the aggregation, translation, and disaggregation processes in the low-frequency regime at designated levels. Red box represents the testing box, where the light blue boxes are three far boxes among all others. Thick silver arrows show the translation vectors as well as operations regarding the far boxes.

other hand, for OCBs, we just anterpolate the translated values collected from surrounding far boxes and update the corresponding positions inside a dedicated matrix. These anterpolated components will be tested in the lower levels or may be at the last level as one of the truncated boxes in that level. The whole process of aggregation plus interpolation, translation, and anterpolation plus disaggregation are illustrated in Fig. 5.15. It should be emphasized one more time that regardless of the population, the last level boxes are treated as TBs to prevent any anterpolation over them.

5.5.4.3 Practical Notes

Since MATLAB is used to write the IL-MLFMA codes, it is worth mentioning some practical notes about its implementation.

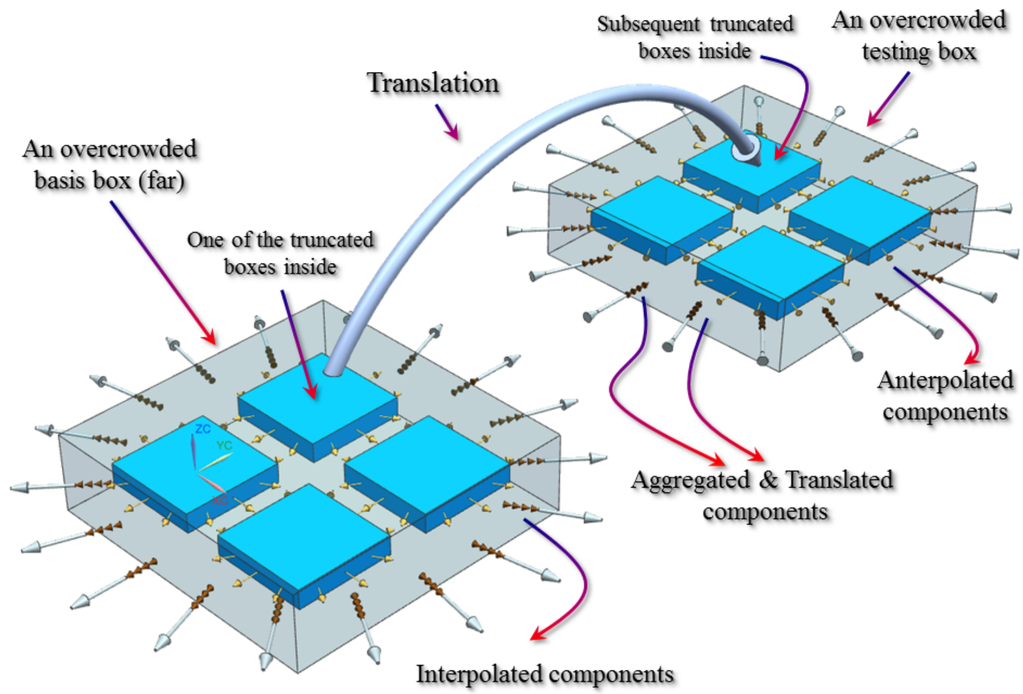


Figure 5.15: A complete illustration of the aggregation, translation, and disaggregation processes in the high-frequency regime at designated levels. It is considered that the interaction is between two OCBs in a typical level. Both transparent large boxes contain four TBs, where yellow arrows show radiation and receiving functions based on their direction (outward or inward). Thick silver arrows show the translation vectors as well as operations regarding the far boxes.

1. The program has been written in a modular fashion which means that, it is straightforward to convert the whole program from PEC to dielectric formulations.

2. We use MEX compiled codes to accelerate many of the routines containing concatenated for loops. Most of the time, this scenario effectively speeds up the execution of complicated routines, but in some occasions it slows down. To overcome this issue, the only remedy is to vectorize the needed routine(s) as much as possible.

3. Due to extensive use of large arrays inside the program, MEX codes are not able to handle arrays larger than $2^{31} = 2,147,483,647$ elements. This limit is currently an issue for both 32 and even 64 bit systems. Note that normal

routines do not suffer from this limitation. Thus, they can use all available memory but they will be very slow.

4. MATLAB built in m-files and routines are very versatile and general. Considering only the ease of programming, this is very important because of flexibility, but considering the run time, they are very slow. Consequently, we changed most of the needed functions, like dot product, cross product, binary to decimal and decimal to binary to achieve maximum speed.

5. Sharing a small or even a medium sized array or matrix among different programs (e.g., the main program to its subroutine(s)) is fast enough to ignore the latency, but when it becomes larger (around GB), the delay is considerable. This issue manifests itself within iterative routines in which large MVMs are in progress. To circumvent this issue, it is better to pass the large arrays as an argument and prevent global variables.

6. Writing and especially reading data in MATLAB are dramatically slow. Therefore, using hard disk as a temporary place to store data is not efficient. In other words, we have to store all needed data in memory. Note that this is not true for other popular programming languages like Fortran or C.

5.6 Numerical Results

To make a fair comparison between all three versions of MLFMA, we merged them together. This is mainly because of four reasons.

1. As mentioned in the fourth note of the previous subsection, our new codes are faster than previously written codes for the conventional MLFMA and even newly developed scaled version of it. Therefore, to evaluate the performance of the new method, we have to equalize other parts of the codes that are common among them. As an example, the most time-consuming part of the near-field calculation is the singularity extraction routines. Inside these

routines, dot product and cross product functions are widely used, which make them very slow. Hence, they should be replaced with faster ones.

2. In some of the routines that belong to the conventional and scaled MLFMA, there was no option to change the order of the numerical integration. We modified them to include this option. This is quite important as we will see for the upcoming results.

3. Obviously, timing information is one of the key factors to compare the results. Hence, we had to unite them in a structured manner to have a fair comparison.

4. We have only the Mie-series solution as an analytical reference for RMS error computations. Therefore, an appropriate Mie routine has been integrated within the program regardless of which MLFMA version is used.

We start the numerical examples by using a PEC sphere of radius $R = 0.25$ m (Fig. 5.16) at $f = 500$ MHz with a non-uniform flat triangular mesh consisting of 7458 RWG functions (4972 triangles), where the edge size varies from 60 mm at equator down to 3 mm at north pole (with a multiscale factor of 20). It should be noted that one of the key benefits of IL-MLFMA is the use of "population threshold". It provides an extra degree of freedom to optimize the efficiency or accuracy or may be both. By changing the population threshold using the automatic partitioning mode, the program decides how many levels it takes to fulfill this requirement.

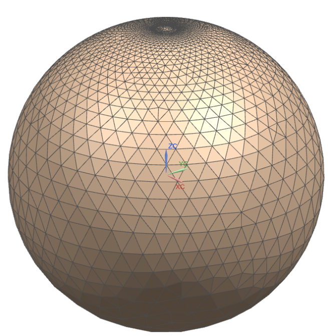


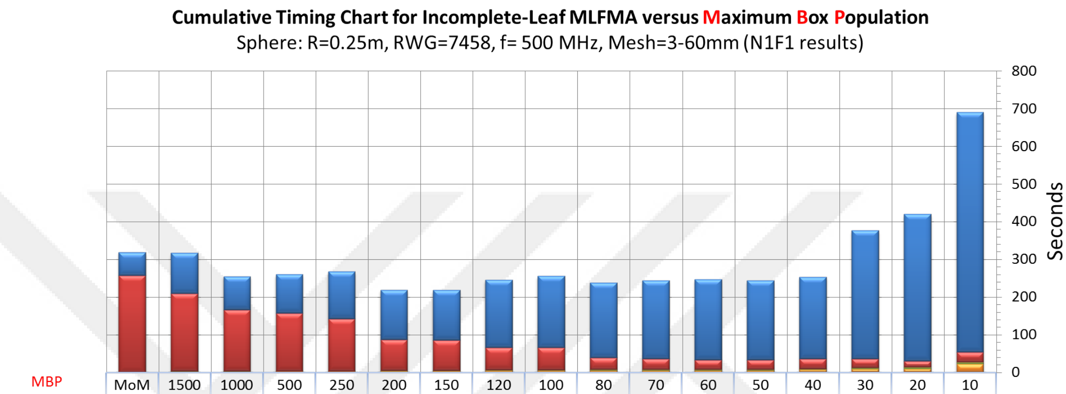
Figure 5.16: Sphere with a non-uniform mesh used in our simulations. Mesh size changes from 3 mm to 60 mm, with a multiscale factor of 20 and consisting of 4972 triangles.

On the other hand, practical observations based on a wide range of simulations reveal that the number of numerical integration points over basis and testing triangles (≡functions) used inside the near-field routines has a considerable impact on the overall accuracy of the results. This effect is somehow negligible for the far-field computations except the fact that it dramatically increases the simulation time. This is mainly because of intensive field variations regarding near and pseudo-near boxes. As already discussed before, one has to bear in mind that near-field calculations are not merely for near distances, so the size of those basis functions located inside the larger boxes are also effective in the overall distribution of the radiated fields and consequently in numerical computations of corresponding integrals.

5.6.1 Timing Results for Incomplete-Leaf MLFMA

By changing the order of numerical integration from 1 to 2, four different sets of results can be achieved. To name them properly, we use N or F labels beside a number (1 or 2) to designate near-field and far-field calculations with desired integration orders. For example, N2F1 means that the second order integration and first order integration are used for near-field and far-field calculations, respectively. Based on the aforementioned numerical reasons, only N1F1 and N2F1 simulations are included in timing results as a function of maximum box population (MBP, i.e., RWGs/box). Based on the results, illustrated in Fig. 5.17(a) and (b), we have the following conclusions:

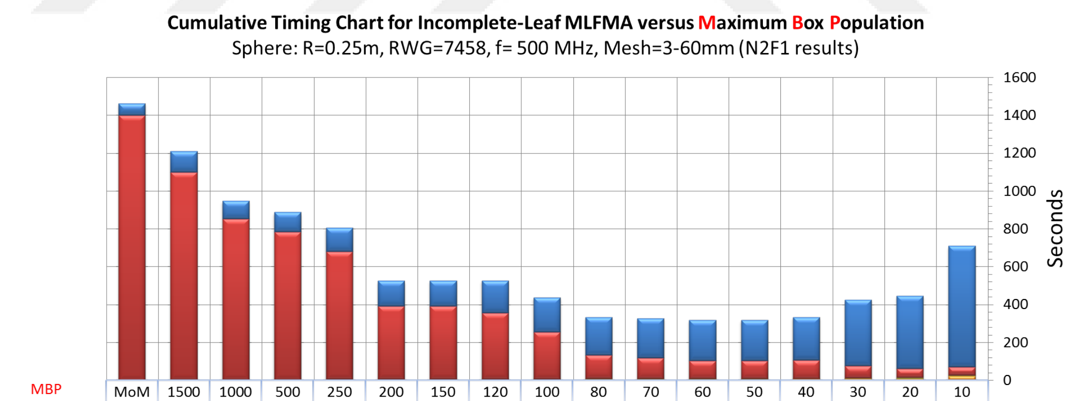
1. Construction of the tree structure and also computing the translation vectors (functions) are hardly visible in comparison to the solution time and near-field calculations.
2. Both timings clearly reveal that near-field calculations increase with population threshold in a significant manner (red part of each column).
3. Solution times decrease as the near-field times increase. Therefore, there are some optimum points in both sets.



MBP	MoM	1500	1000	500	250	200	150	120	100	80	70	60	50	40	30	20	10
Solve	61	107	89	102	124	132	132	179	190	198	207	213	210	216	340	389	636
Nearfield	257	207	163	154	139	82	81	59	59	32	29	26	26	27	24	17	27
Translation	0	0.6	0.8	0.9	1	1.8	1.77	2	2	2.7	2.7	2.7	2.7	2.7	3.5	3.6	4.3
Tree	0.6	2	1.9	2.9	2.9	2.9	2.9	4.7	4.6	4.8	4.9	5	5	7	9	9.7	22.5

Seconds

(a)



MBP	MoM	1500	1000	500	250	200	150	120	100	80	70	60	50	40	30	20	10
Solve	62	110	94	104	125	134	135	169	182	201	209	214	213	227	348	385	638
Nearfield	1400	1099	850	780	676	386	386	350	249	125	110	97	97	97	63	48	43
Translation	0	0.4	0.8	0.9	1.1	1.8	1.8	2.1	2.1	2.7	2.7	2.7	2.7	2.7	3.5	3.7	4.5
Tree	0.6	1.2	1.9	2.8	2.9	2.9	2.9	4.5	4.6	4.8	4.9	4.9	4.9	7	8.9	9.7	23

Seconds

(b)

Figure 5.17: Timing diagrams for important routines used in IL-MLFMA for two sets of integration orders regarding near-field and far-field calculations. Simulations are carried out for a sphere of $R = 0.25$ m at $f = 500$ MHz, depicted in Fig. 5.16. Multiscale factor for the mesh is about 20. (a) Considering the first-order integration for both near-field and far-field calculations. (b) Considering the second-order integration for the near-field and the first-order integration for the far-field calculations.

4. Although we can see two optimal timings for MBP around 150 $\mapsto$ 200 in the first set (N1F1), they are not real optimums due to some computational overheads concerning the near-field and far-field routines.

5. Careful probing of both sets (including many others) suggests that there are optimum MBP values between 50 and 250.

6. The number of required levels to keep box populations under the given MBP is not shown here, because it is not an important parameter for IL-MLFMA. However, from the computational point of view, and in order to compare its efficiency and accuracy with other MLFMA versions, the number of required levels varies from 8 to 2 ($\equiv$MoM) for MBP from 10 to 2000, respectively.

5.6.2 Accuracy Results for Incomplete-Leaf MLFMA

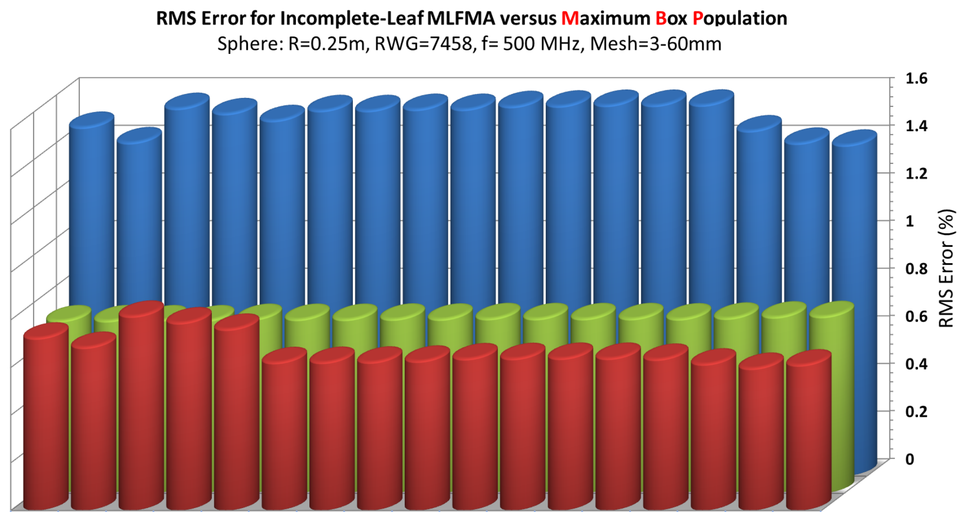
Using Mie-series as an analytic solution for a scattering problem of an ideal PEC sphere, we can calculate the RMS error as

$$\text{RMS Error} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \left|\boldsymbol{E}_{\theta_i}^{\text{Sim}} - \boldsymbol{E}_{\theta_i}^{\text{MIE}}\right|^2}, \tag{5.7}$$

where $\boldsymbol{E}_{\theta_i}^{\text{Sim}}$ and $\boldsymbol{E}_{\theta_i}^{\text{MIE}}$ are the dominant θ components of the far-zone electric field, calculated with MLFMA and Mie solutions, respectively. Fig. 5.18 shows the corresponding RMS errors for different values of MBP using the same sphere. N2F2 results are also included for further comparison. Some important aspects of RMS error results are as follows:

1. The RMS error is almost independent of the box population. This is very important, since it gives us the opportunity to optimize the simulation time as the second important parameter.

2. Considering only the optimum values of MBP obtained from the timing results, i.e., 50 $\mapsto$ 80, and by increasing the integration order from one to



MBP	MoM	1500	1000	500	250	200	150	120	100	80	70	60	50	40	30	20	10
RMS-Error (N2F1)	0.724	0.685	0.816	0.788	0.761	0.621	0.622	0.623	0.627	0.637	0.638	0.639	0.639	0.635	0.615	0.597	0.611
RMS-Error (N2F2)	0.724	0.724	0.724	0.724	0.73	0.73	0.73	0.731	0.731	0.733	0.733	0.733	0.733	0.732	0.735	0.74	0.742
RMS-Error (N1F1)	1.464	1.399	1.543	1.521	1.492	1.534	1.534	1.539	1.539	1.551	1.553	1.556	1.556	1.557	1.449	1.397	1.389
Max Level	2	3	4	5	5	5	5	6	6	6	6	6	6	7	7	7	8

Figure 5.18: RMS errors as a function of MBP for three different scenarios, i.e., N1F1, N2F1, and N2F2 for the same sphere shown in Fig. 5.16.

two, the RMS error effectively decreases by around 52%.

3. It should be noted that by decreasing the box population threshold, the chance for basis or testing functions to stick out of their corresponding boxes increases. Hence, we expect to see a bit higher RMS error for some lower values of MBP.

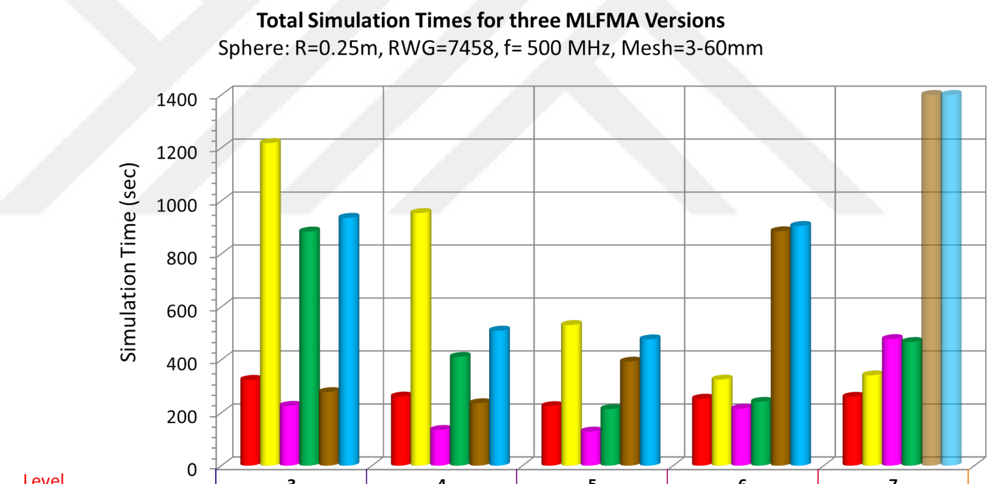
4. It seems that increasing the integration order, somehow degrades the accuracy of far-field calculations. This maybe due to the fact that the location of integration points may fall outside the corresponding boxes leading to some inaccuracies regarding the use of addition theorem. This is not true for near-field calculations because of two reasons: (a) We always use singularity extraction routines for the basis functions which means that the evaluated integrals are exact. (b) There is no other approximation (like addition theorem) other than the numerical evaluation of the needed integrals.

5. The last line of data shown under the graph is the maximum level used to partition OCBs. As we see later, none of the other MLFMA versions can

reach to such lower levels without dramatic loss in accuracy.

5.6.3 Accuracy and Timing Results: All Versions

Up to this point, we have demonstrated that by using a proper value for MBP, we can simultaneously optimize the accuracy and also the total run time. To see the performance of other two methods over multiscale meshes, we carry out similar simulations to compare with the new method. These simulations are based on the number of levels and also the order of the integration.



Level	3	4	5	6	7
Total Time (IL N1F1)	324	261	225	253	260
Total Time (IL N2F1)	1218	954	531	325	341
Total Time (Normal N1F1)	226	135	129	215	478
Total Time (Normal N2F1)	884	411	214	241	467
Total Time (Scaled N1F1)	278	236	393	885	
Total Time (Scaled N2F1)	936	510	477	905	

Figure 5.19: Total simulation times for three versions of MLFMA based on level-wise comparisons. Note that the last two transparent columns do not exist due to the fact that corresponding routines do not converge at all.

Using the same sphere as before, Fig. 5.19 illustrates the total simulation time for three versions of MLFMA. Note that IL-MLFMA does not work based on the number of levels (in automatic mode). Consequently, we choose the optimum values for the simulation results among the same group of levels. Something important to bear in mind is that, because none of the other MLFMA versions are able to converge at deep levels, no data is available for levels larger than 8 and even part of level 7. This is shown by cross-hatched cells inside the figure.

Also the last two columns are painted in transparent to imply no data. Careful investigation of the given data reveals the followings.

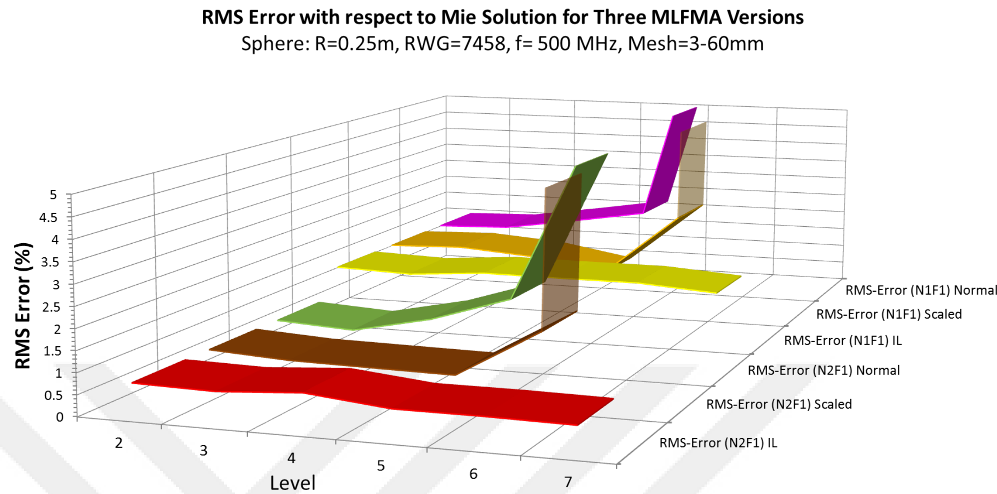
1. Considering N1F1 results for the IL, normal, and scaled versions (red, purple, and brown), one can see that the timings are comparable to each other before level 4; but starting from level 6, IL shows a faster simulation time. This is more evident at level 7.
2. Considering N2F1 results (yellow, green, and blue), which are more accurate, the same trend exists starting from level 5 on.

It is very important to note that even though the conventional MLFMA shows the best simulation time over N1F1 scenario around level 4, 5, and 6 (corresponding to 135, 129, and 215 seconds), as we see in accuracy charts, its RMS error dramatically increases by 18%, 40%, and 490% with respect to the IL-MLFMA results. To have a better view regarding the accuracy, Fig. 5.20 shows the performances of all three versions with respect to the level. It is important to remind that the scaled and conventional MLFMA versions either do not converge at level 7 or they converge with a very high amount of error. The vertical transparent patches at the end of the graphs imply this issue.

Regarding the accuracy, the conventional MLFMA with N1F1 scenario drops behind the others very quickly. For N2F1 results, even though it has an acceptable accuracy for levels 2 and 3, corresponding simulation time is two to three times longer than that of the IL version.

Regarding the scaled version, the accuracy level is close to the IL version and even sometimes better than that for both N1F1 and N2F1 scenarios before level 5. After level 5, the RMS error increases rapidly by an amount of 56% to 203%, which means that the computed RMS values are not accurate enough or even not trustable. One logical explanation for such a behavior originates from the protrusion of the basis functions from their assigned boxes at the leaf level.

Considering both figures carefully and regardless of aforementioned argument



Level	2	3	4	5	6	7
RMS-Error (N2F1) IL	0.724	0.685	0.816	0.621	0.622	0.597
RMS-Error (N2F1) Scaled	0.724	0.613	0.586	0.562	1.721	
RMS-Error (N2F1) Normal	0.724	0.611	1.024	1.658	5.943	17.11
RMS-Error (N1F1) IL	1.464	1.399	1.543	1.534	1.534	1.397
RMS-Error (N1F1) Scaled	1.464	1.524	1.378	1.039	2.665	
RMS-Error (N1F1) Normal	1.464	1.523	1.825	2.149	9.165	23.731

Figure 5.20: RMS error with respect to the Mie solution to investigate the accuracy of all three versions as a function of level. Cross-hatched cells represent the lack of data for that level due to convergence problems.

about validity of computed RMS values, it seems that the only potential competitor is the scaled version with N1F1 scenario at level 4, where the simulation time and corresponding accuracy is near to IL N1F1 at level 5. Other than that, the IL-MLFMA represents better timing along with robust and stable accuracy performance.

5.6.4 Efficiency and Accuracy for a Higher Multiscale Factor

At this stage, we introduce and use a modified version of (5.7), i.e., relative RMS error to assess the accuracy of the forthcoming results obtained from simulations. This is crucial and the main reason is that, for electrically smaller spheres, the absolute values of scattered electric fields get smaller, misleading to smaller RMS errors using (5.7). By normalizing the previously defined RMS error with the

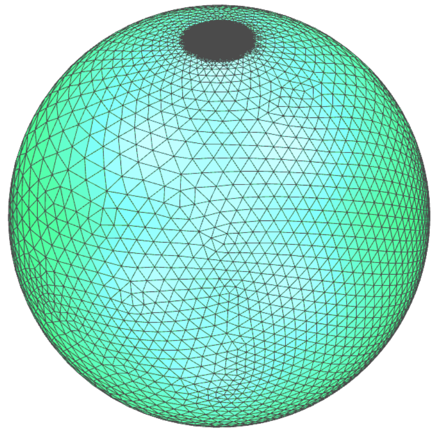


Figure 5.21: A PEC sphere of radius $R = 5$ cm possessing a dense discretization around the north pole with a multiscale factor of 110.

RMS value of the Mie-series solution, a more realistic error gauge can be obtained. This new error gauge is the same as (4.38) and it is repeated here for convenience

$$\text{Relative RMS Error} = \sqrt{\frac{\sum_{i=1}^{N} \left|\boldsymbol{E}_{\theta_i}^{\text{Sim}} - \boldsymbol{E}_{\theta_i}^{\text{Mie}}\right|^2}{\sum_{i=1}^{N} \left|\boldsymbol{E}_{\theta_i}^{\text{Mie}}\right|^2}}. \tag{5.8}$$

Note that in the previous section, we use a sphere of radius $R = 25$ cm at 500 MHz to illustrate and compare the results. Now, we proceed by more complicated simulations with wider range of frequencies and different spheres to test and demonstrate another aspect of this method, which is expected to be a broadband method.

To investigate the performance of IL-MLFMA for higher multiscale factors, another PEC sphere of radius $R = 5$ cm ($\lambda/2$) is used at $f = 3.0$ GHz as depicted in Fig. 5.21. The edge size for the triangles varies from 0.135 mm ($\lambda/740$) at the north pole upto 15 mm ($\lambda/7$) at the south pole resulting a nonuniform mesh that has a multiscale factor of 110 and consisting of 11343 RWG functions. The MFIE formulation in conjunction with N2F1 scenario is used for this simulation.

Table 5.2 includes the run-times and the relative RMS errors of IL-MLFMA for this geometry with respect to MBP. Regarding the run-time, we may arrive at almost the same conclusions as we had in Section 5.6.1.

1. Construction of the IL tree, though it is more complicated, takes at most 1% of the total run-time.

2. The processing time required to fill the near-field matrix decreases as the value of MBP decreases. On the other hand, due to the growing tree structure, the solution time increases. Therefore, there is an optimal value of MBP for the best efficiency in terms of the total time. This optimal value is between 50 and 250 for different problems.

3. It is observed that IL-MLFMA always converges regardless of the value of MBP but this is not true for other two versions of MLFMA. This is mainly because of small box sizes at leaf levels and protrusion of RWG functions from relatively small boxes.

Regarding the accuracy and based on Table 5.2, we may arrive at similar conclusions as we discussed in Section 5.6.2 as follows.

1. The accuracy is almost independent of the number of levels. This is important because it provides us the opportunity to only optimize the total run-time rather than the accuracy as a function of MBP.

2. The relative RMS error observed in the IL-MLFMA results is comparable with that of MoM.

Next, the same PEC sphere, depicted in Fig. 5.21, is used to compare the efficiency and accuracy of IL-MLFMA with (a) the conventional MLFMA, and with (b) the scaled MLFMA. Fig. 5.22 illustrates the total run-time (left vertical axis; and blue (black) lines are used) and the relative RMS error (right vertical axis; and orange (gray) lines are used) versus the number of levels for the three versions of MLFMA. Considering the accuracy, all three versions are comparable to each other upto level 4. However, due to a high multiscale factor, the results obtained from the conventional and scaled MLFMAs start to become inaccurate after level 4, and both versions fail to converge for large number of levels (no data is available for levels higher than 6 for both version, which is shown by dashed

Table 5.2: Timing and Error of IL-MLFMA Applied to a PEC Sphere of $R = 5$ cm at $f = 3.0$ GHz Consisting of 11,343 RWG Functions with Multiscale Factor of 110 with Respect to MBP.

MBP	2700	2500	2350	2200	1000	500	300	200	**100**	**90**	**60**	**50**	20
Maximum Level	2	3	4	5	6	7	7	8	8	8	9	9	10
Tree (sec)	1.1	1.81	2.8	4	5.4	4.3	6.8	8.4	9	8.4	10.4	10.5	15.5
Near-field Time (sec)	2961	2977	2896	2738	1972	1880	968	672	405	375	312	275	82
Solution Time (sec)	344	354	427	393	427	495	498	567	672	689	756	812	1153
Iteration	64	64	64	64	64	64	64	64	64	64	64	64	65
Total Run-Time (sec)	3316	3344	3336	3146	2415	2391	1484	1259	**1099**	**1084**	**1091**	**1110**	1263
Rel. RMS Error (%)	2.757	2.755	2.750	2.731	2.728	2.731	2.699	2.617	**2.617**	**2.617**	**2.614**	**2.613**	2.543

lines in the figure). On the other hand, the efficiency of IL-MLFMA is much better than the both versions for the optimal MBPs that correspond to 8th or 9th level for IL-MLFMA, where the accuracy remains almost stable. This is more evident regarding larger problems involving larger numbers of unknowns, where more levels are required.

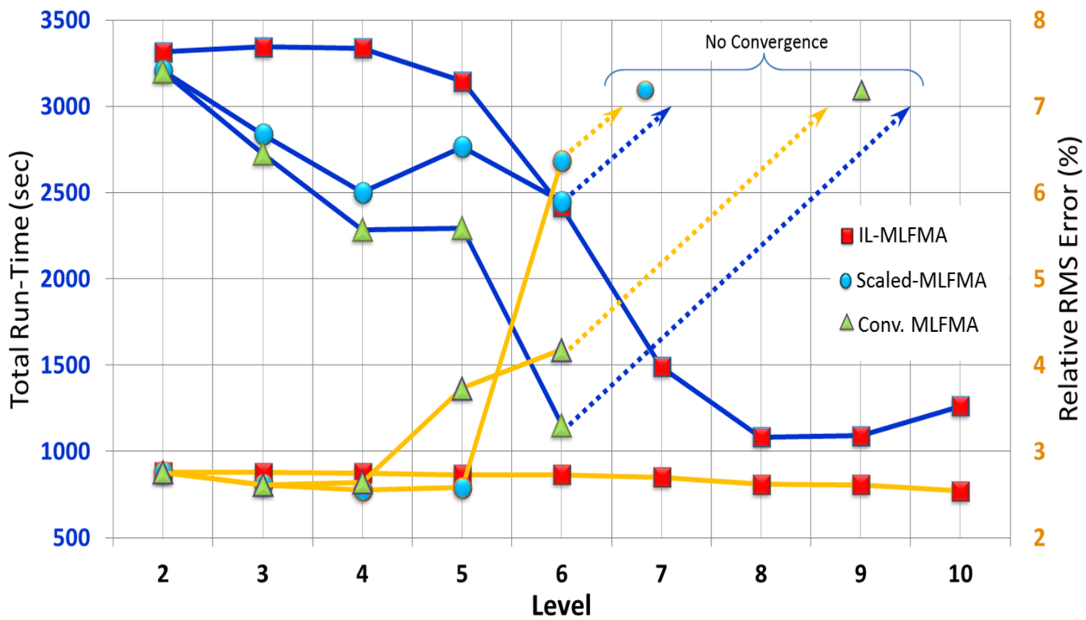


Figure 5.22: Total run-times and relative RMS errors versus the number of levels for three versions of MLFMA. The geometry in Fig. 5.21 is used at $f = 3.0$ GHz. Square, circle, and triangle markers are used to separate three MLFMA versions. Black (blue) lines and gray (orange) lines along with corresponding vertical axis at left and right show the total run-times and the relative RMS errors, respectively.

5.6.5 Efficiency and Accuracy for Larger Problems

To evaluate and demonstrate the superiority of IL-MLFMA, we investigate three larger problems. Table 5.3 summarizes the solutions of scattering problems involving four discretizations of the PEC sphere shown in Fig. 5.21 ($R = 5$ cm at 300 MHz) formulated with MFIE. The first model is the same as before, involving 11,343 RWG functions with a multiscale factor of 110. The other models are obtained by mesh refinement leading to 45,372, 181,488, and 725,952 RWG functions, respectively, retaining the same multiscale factor. Note that the conventional MLFMA results are not included either due to very large errors or due to convergence problems. According to these results, (a) IL-MLFMA is two to six times more efficient than the scaled-MLFMA, (b) accuracy is also more than six times better than the scaled version for a large problem.

Table 5.3: Performance Comparison Between IL-MLFMA and Scaled-MLFMA Applied to Four Densely Meshed Spheres of $R = 5$ cm at $f = 300$ MHz with Multiscale Factor of 110

Number of Unknowns	11,343		45,372		181,488		725,952	
Method	IL	Scaled [1]	IL	Scaled[2]	IL	Scaled[3]	IL	Scaled [4]
MBP	100	-	100	-	100	-	100	-
Level	8	3	9	6	10	8	11	-
Total Time (sec)	1331	2674	7018	22131	27171	184550	136441	-
Near-Field Time (sec)	399	2315	1898	16679	4030	26128	15568	-
Solution Time (sec)	917	346	5023	5394	22624	158039	113578	-
Iteration	61	60	64	64	66	196	67	-
Rel. RMS Error (%)	2.45	2.60	1.49	1.69	1.37	9.04	1.37	-

[1] The best solution regarding efficiency and accuracy [2] The best result among two available solutions
[3] The only available solution [4] No solution due to convergence problem

5.6.6 A More Complicated Sphere Problem

Finally, we consider a PEC sphere with a more complicated discretization including multiple overmeshed areas, as illustrated in the inset of Fig. 5.23. In this figure, three among six highly-overmeshed areas, which are aligned with the positive and negative directions of the x, y, and z axes over the sphere, can be seen. A refined version of this mesh with a multiscale factor of 100 is used at $f = 300$ MHz, containing 557,184 RWG functions. Fig. 5.23 depicts the far-zone electric field as a function of the bistatic angle, along with the Mie-series solution, where 0° and 180° correspond to the forward-scattering and backscattering directions, respectively. It is seen that there is a very good agreement between the Mie-series results and the values obtained with IL-MLFMA. Small amount of discrepancy is originated from the approximate method in the low-frequency regime. Note that, for the same reason, the relative RMS error of 1.37% given in Table 5.3 does not change for very fine meshes, i.e., when the discretization error is minimized and the low-frequency inaccuracies dominate the total error.

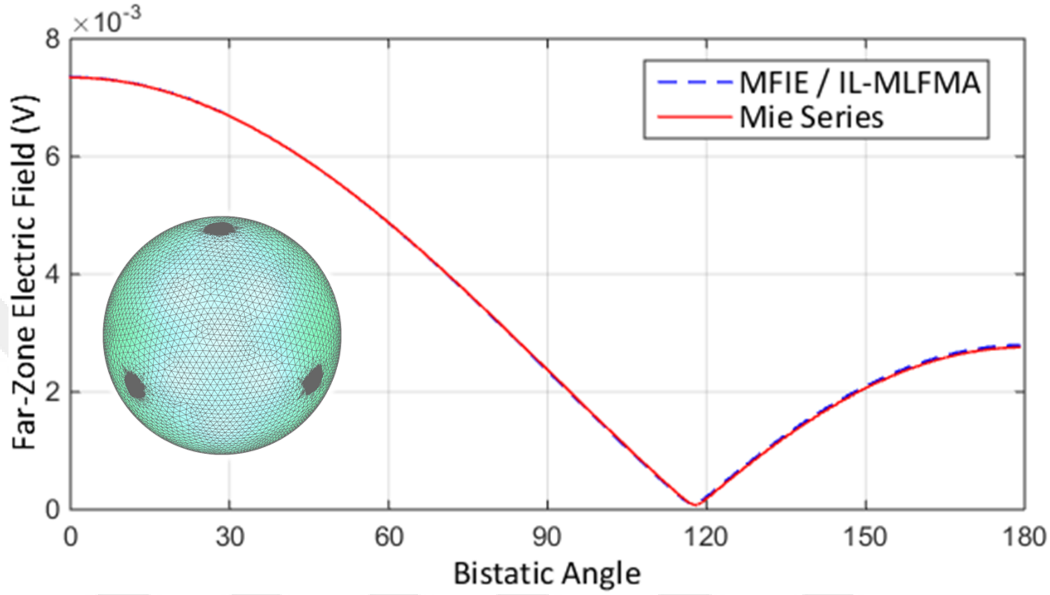


Figure 5.23: Far-zone electric field scattered from a PEC sphere of radius $R = 5$ cm with a more complicated discretization as shown in the inset, illuminated by a unit plane wave at 300 MHz. The sphere contains six dense mesh regions aligned with the positive and negative directions of x, y, and z axes, where only three of them are visible in the figure. The multiscale factor is about 100, leading to 557,184 RWG functions.

5.6.7 An Extreme Case: Sphere of $D = 2R = 4\lambda$

So far, we have simulated and compared three versions of MLFMA for electrically small objects, at most up to 0.5λ. To demonstrate how IL-MLFMA performs for much larger objects, we use another sphere of size $D = 2R = 1$ m $(= 4\lambda)$, similar to Fig. 5.16. A higher multiscale factor of 375 was chosen with minimum and maximum mesh size of 0.08 mm and 30 mm, respectively, consisting of 722,529 unknowns. It is excited by a plane-wave at $f = 1.2$ GHz $(\lambda = 25$ cm$)$. A population threshold of 100 leads to 12 levels of FMM, where 5 deep levels are in the low-frequency regime, and the rest in the high-frequency regime. Fig. 5.24 depicts the far-zone electric field as a function of the bistatic angle, besides the Mie-series solution. The relative RMS error is 4.24% with a simulation time of about 26.5 hours. It is seen that there is a good agreement between Mie-series results and the values obtained with the IL-MLFMA, except a small discrepancy at the forward scattering portion of the curve. This is mainly due to two reasons: (a) A big part of the meshed surface consists of elements with edge sizes larger than $\lambda/10 = 20$ mm, which degrade the simulation accuracy a bit. (b) Almost half of the deep FMM levels reside in the low-frequency regime, where an approximate method is utilized to handle it.

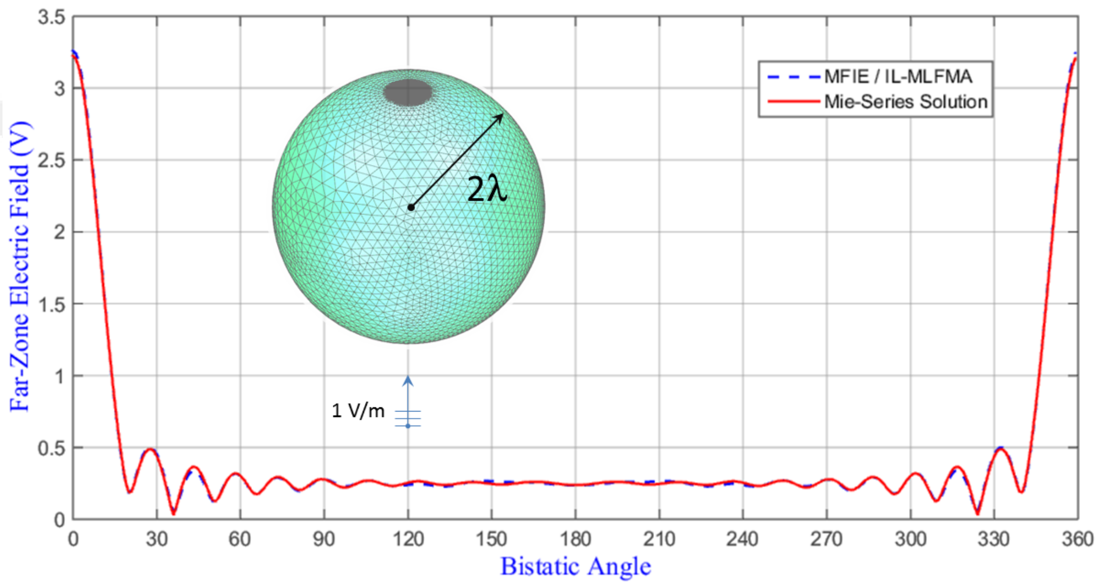
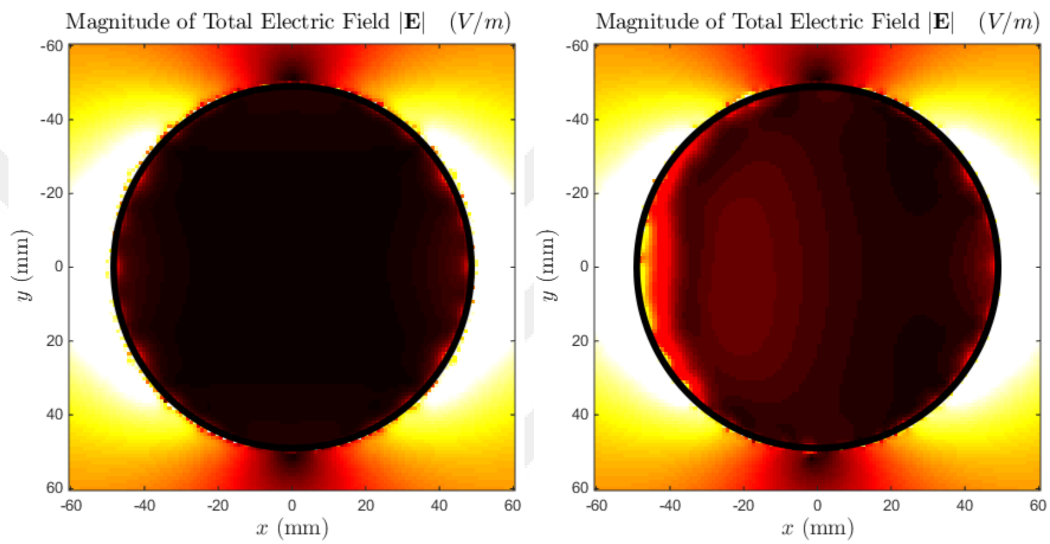


Figure 5.24: Solution of a scattering problem using IL-MLFMA, involving a large PEC sphere of radius $R = 0.5$ m (2λ) illuminated by a unit plane wave at $f = 1.2$ GHz. The discretized surface consist of 722,529 basis functions possessing a multiscale factor of 375. The population threshold was chosen to be 100 leading to 12 levels of FMM.

5.6.8 A Near-Field Test

To find out the performance of IL-MLFMA in near-zone fields, we use the third sphere example given in Table 5.3 with a radius of $R = 50$ mm consisting of 181,488 RWG functions. Both of the figures in Fig. 5.25 illustrate the magnitudes of the total electric field on the xy plane, inside the PEC sphere as well as an extra 10 mm around it, assuming the center of sphere at origin. The sphere is illuminated by a unit plane wave at 300 MHz, propagating in the z direction with the electric field polarized in the x direction. Fig. 5.25(a) shows the IL-MLFMA solution with a relative RMS error of 1.37%, while Fig. 5.25(b) shows the only available scaled-MLFMA solution with a relative RMS error of 9.04%. High amount of error can be distinguished in the latter specially inside the sphere, where a zero electric field is expected. It should be noted that the high values of the field on the boundary are due to singularity.



(a) (b)

Figure 5.25: Near-field total electric field values on the xy plane for a PEC sphere of radius $R = 50$ mm illuminated by a unit plane wave at 300 MHz, propagating in the z direction with the electric field polarized in the x direction. The sphere is chosen to be the third example given in Table 5.3 regarding 181,488 RWG functions. (a) Solution using IL-MLFMA with 1.37% error. (b) Solution using Scaled-MLFMA with 9.04% error.

5.6.9 Realistic Examples

In the previous numerical examples, we use spheres with different discretizations to demonstrate the efficiency and accuracy of the proposed IL-MLFMA. This is mainly due to the fact that for a scattering problem of a PEC sphere, we have an analytic Mie-series solution to compare the results with. But when we deal with some real-life problems, there is no such an analytic solution and hence, the accuracy assessment procedure fails or at least, it is not solid enough. To circumvent this important issue, it seems that we can rely upon three approximate solutions as follows.

1. If a full MoM solution can be computed, even though its cost could be quite high, then it is the best feasible solution to compare the other results, and to have an estimation about the relative RMS error. Note that MoM solution is just a two-level MLFMA solution.

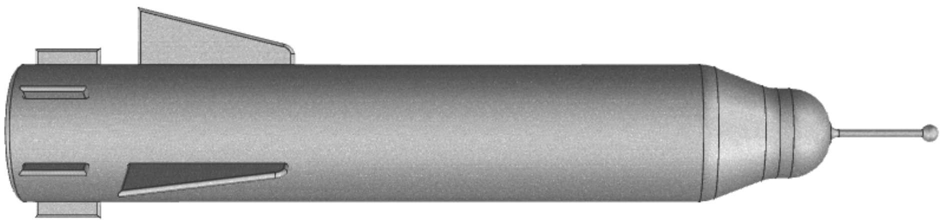
2. If a MoM solution is not available, the next most trusted solution would be the FMM solution, which is a three-level MLFMA solution. In FMM, the memory consumption is considerably lower than the MoM, but still the simulation time may be very high since 24.4% of the computations are based on MoM (1000 out of $64 \times 64 = 4096$ interactions among $4^3 = 64$ boxes). The rest (75%) of the computations are based on FMM, which in turn relies on the addition theorem via well separated large boxes with acceptable accuracy.

3. Apart from the above, there is no obvious choice than tracing the convergence of the solutions at some higher levels.

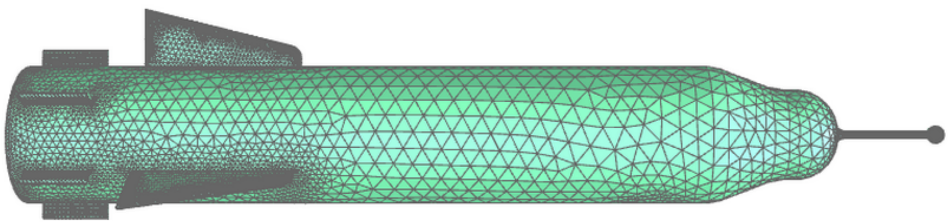
Here, we present two other numerical examples, in which the first two options mentioned above are used as reference solutions (instead of the Mie-series) to compute the relative RMS error via (5.8).

5.6.9.1 Missile

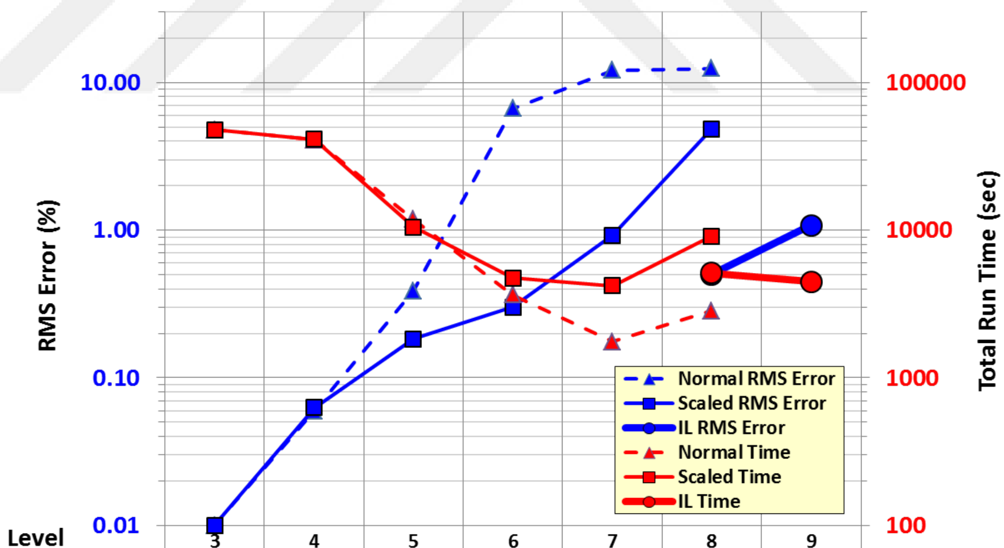
As the first example, we consider a hypothetical metallic missile (assumed to be a PEC object) of length 1.3 m ($\approx 2\lambda$ at $f = 500$ MHz) as illustrated in Fig. 5.26(a). It is illuminated by a unit plane wave that propagates in the z direction with the electric field polarized in the x direction. Fig. 5.26(b) partially depicts its discretized surface with 31,966 triangles or 47,949 RWG functions. The multiscale factor is about 54. By considering the FMM solution as a reference, we compute the far-zone scattered electric fields, and calculate the relative RMS error. The RMS errors as well as the total run times are given in Fig. 5.26(c) for comparisons. Note that the axes are in logarithmic scale. We can arrive at the following conclusions.



(a)



(b)



Level	3	4	5	6	7	8	9
Normal RMS Error	0.01	0.06	0.39	6.65	12.12	12.40	
Scaled RMS Error	0.01	0.06	0.18	0.30	0.92	4.86	
IL RMS Error						0.50	1.08
Normal Time	47779	40973	11813	3658	1763	2838	
Scaled Time	47850	41100	10449	4726	4213	9085	
IL Time						5134	4477

(c)

Figure 5.26: (a) A 1.3 m ($\approx 2.15\lambda$) hypothetical PEC missile illuminated by a unit plane wave at 500 MHz, propagating in the z direction with the electric field polarized in the x direction. (b) The surface of the missile is discretized with 31,966 triangles or 47,949 RWG functions. The multiscale factor is 54. (c) The relative RMS error and the total run time for the missile problem calculated via three versions of MLFMA. The red and blue lines are matched with their respective axes.

1. The results for the third and fourth levels are almost the same for both conventional and scaled versions. This is because of the fact that the scaled version is the same as the conventional version for those objects in which, the box sizes are larger than or equal to $\lambda/4$ after recursively clustering the whole object. In this case and at level four, the box size is $2.15\lambda/2^{(4-1)} \geq \lambda/4$.

2. At level five (where the low-frequency treatment starts for the scaled version), despite the fact that the error levels are a bit less than the IL version, the timing of both versions are much higher than the IL method.

3. After level six, the RMS error of the conventional MLFMA is not acceptable, though it shows a better timing. The same happens for the scaled version starting from level seven. Hence, it seems that only the sixth level of the scaled version is the only competitor for our proposed IL version.

5.6.9.2 Water Mine

It is worth to remind that the IL-MLFMA is most suitable for large problems with high multiscale factors, where there is no other choice than using some higher levels of MLFMA to solve the problem due to computational issues. As a final numerical example, we solve a medium-sized PEC scattering problem with a multiscale factor of 550. The object is a hypothetical water mine of size 0.7 m ($\approx \lambda/4$) as shown in Fig. 5.27(a). Its discretized version consisting of 103,527 RWG functions is shown in Fig. 5.27(b), where the mesh size varies from 110 mm on its surface down to 0.2 mm on the side actuators.

In this case, a more powerful computer with a larger RAM has been used to carry out the needed computations up to level 5. Fig. 5.28 illustrates the simulation results for the three versions of MLFMA from level 3 (FMM) up to level 10. Note that the MoM result is missing due to the lack of memory (it needs 190 GB), but the FMM result has been computed as a reference (using 72 GB). Based on these results, the following conclusions can be made.

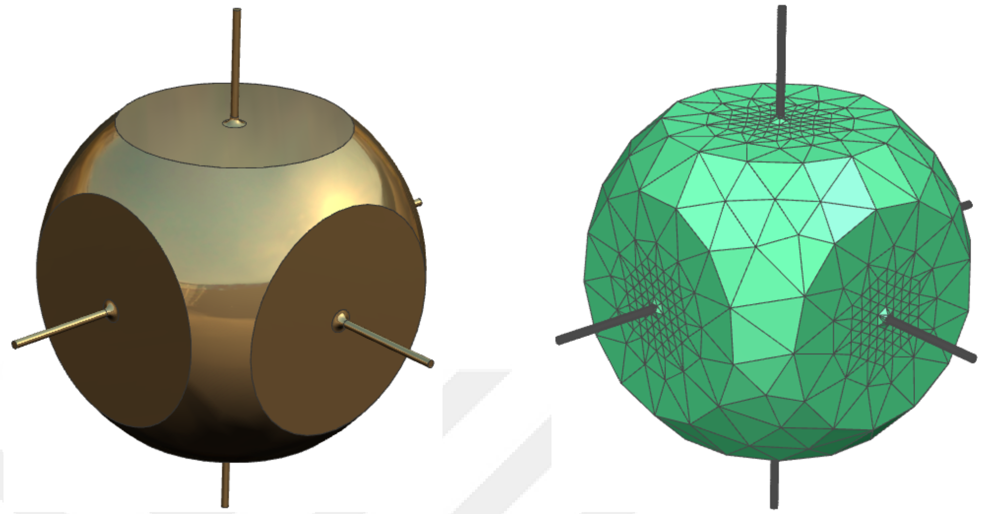


Figure 5.27: (a) A 0.7 m ($\approx \lambda/4$) hypothetical PEC water mine illuminated by a unit plane wave at 100 MHz, propagating in the z direction with the electric field polarized in the x direction. (b) The surface of the mine is discretized with 69,105 triangles or 103,527 RWG functions. The mesh size varies from 110 mm down to 0.2 mm, giving rise to a multiscale factor of 550.

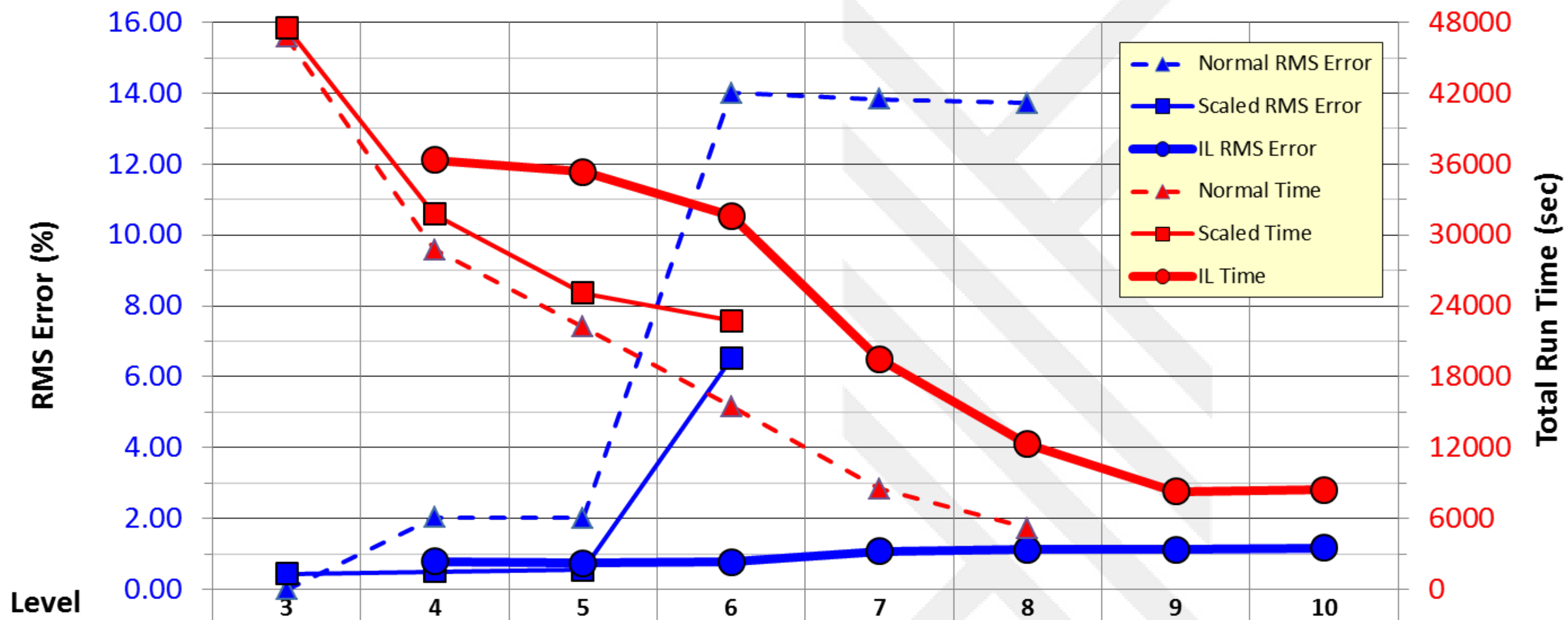
1. At level four, the accuracy and efficiency of all versions are comparable, but the timing of IL version is a bit higher since its tree structure and the total number of near-field interactions are different from the other two versions.

2. Level five is the only useful level for both of the conventional and scaled versions since starting from level six, the error dramatically increases.

3. The IL version shows almost a fixed accuracy around 1% for all levels. Interestingly, the efficiency of the IL version manifests itself at higher levels (9 or 10), where it is 4 times faster than the conventional version and 2.8 times faster than the scaled version.

5.7 Complexity Assessment for IL-MLFMA

The proposed broadband solver shows $\mathcal{O}(N \log^{\alpha} N)$ computational complexity, where α ($1 \leq \alpha \leq 2$) depends on how deep we are getting into low-frequency regime after recursively clustering the object based on the given population threshold.



Level	3	4	5	6	7	8	9	10
Normal RMS Error	0.00	2.04	2.02	14.00	13.83	13.72		
Scaled RMS Error	0.44	0.50	0.56	6.51				
IL RMS Error		0.78	0.75	0.77	1.07	1.13	1.13	1.17
Normal Time	46734	28779	22206	15473	8505	5192		
Scaled Time	47562	31764	25101	22767				
IL Time		36332	35365	31631	19482	12274	8310	8438

Figure 5.28: A comparison between three versions of MLFMA for the water mine problem. The red and blue lines are matched with their respective axes.

The number of levels follows an $\mathcal{O}(\log N)$ complexity, where the base of the logarithm mainly depends on the population statistics. By reusing the information given in Table 5.3, all three important processing times, i.e., the near-field time, the solution time, and the total run-time of the proposed method, are plotted in Fig. 5.29. For comparison purposes, both $\mathcal{O}(N \log_2 N)$ and $\mathcal{O}(N)$ lines are also plotted (in dashed black and gray lines, respectively), where the former passes through the first point of the total run-time line and the latter passes through the first point of the near-field line. As it can be seen from the figure, the efficiency of the proposed method manifests itself for larger and more complicated geometries with a higher number of unknowns. This is more evident comparing the number of near-field interactions (the blue line) and the $\mathcal{O}(N)$ line. Even though the solution and near-field computation times show a bit higher and lower complexities with respect to their dedicated reference dashed-lines, the total run-time closely follows the claimed linearithmic complexity.

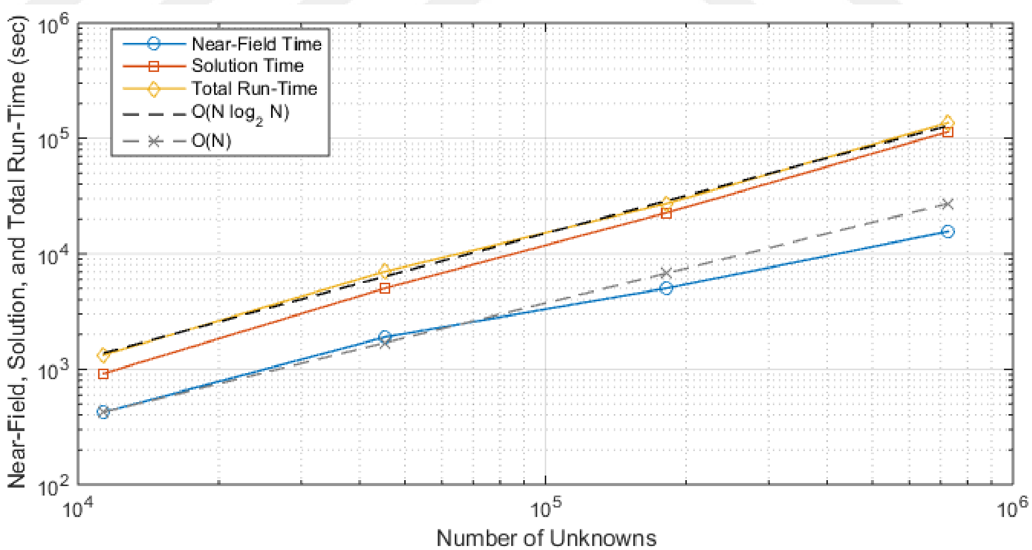


Figure 5.29: Three important processing times, i.e., the near-field time, the solution time, and the total run-time for four scattering problems given in Table 5.3.

Chapter 6

Incomplete-Leaf MLFMA for Volume Integral Equations

6.1 Introduction

There is a wide variety of frequency-domain, and also time-domain, integral-equation based methods that can be used for electromagnetic problems, but frequency-domain SIE and VIE solvers are proved to be among the best solvers for multi-scale problems. This is mainly because of two important reasons: (a) They only find unknown currents on the surface of the object (for SIE) or the fields inside the object (for VIE) by analytically enforcing the boundary conditions, and they never discretize the background medium. (b) These methods do not invoke artificial, and most of the time, approximate mesh termination techniques. Also they are capable of modeling and solving inhomogeneous materials.

SIE solvers are very useful at microwave frequencies, where most of the metalic parts can be modeled as an ideal PEC. Such an approximation reduces computational costs significantly due to the fact that they can be formulated in terms of only the tangential electric currents on the surface. When the PEC approximation is not valid or when there are dielectric materials or some composite objects, SIE solvers can still be used effectively for piecewise homogeneous structures by formulating integral equations in terms of tangential electric and magnetic fields on interface surfaces [67]. As the inhomogeneity increases, SIE solvers are no longer a good choice in comparison to VIE solvers, since the area of the interface surfaces converges to the object volume itself. Obviously for inhomogeneous objects, volume integral equations are inevitable.

Similar to SIE, the VIE solvers use MoM in a quite similar fashion, i.e, the volume fields inside the object are discretized by N sub-domain basis functions with unknown coefficients and then, the resulting integral equations are tested by N testing functions and converted into a linear system of equations. At this stage one has to consider three important issues regarding the MoM [68]. These issues are: (a) The choice of basis/testing functions. (b) The frequency regime in which the problem is intended to be solved. (c) The total computational cost. We briefly elaborate these issues as follows: Considering the first two issues, recall that for SIEs in the high-frequency regime, divergence-conforming functions (e.g., the RWG functions) can be used successfully, but in the low-frequency regime, IEs generally break down when the edge lengths shrink down to a small fraction of the wavelength. This is mainly due to the finite precision arithmetic [69], and the imbalance between vector and scalar potential terms at the frequency of interest. Wide range of solutions have been proposed by different papers to resolve this issue, as discussed in the first chapter. Interestingly, VIEs are not affected by low-frequency breakdown as discussed in [70], and hence, there is no need to separate low- and high-frequency regimes in VIE formulations. On the other hand, regarding the last issue, the traditional MOM suffers from a high computational cost. As we know, it requires $\mathcal{O}(N^2)$ floating point operations (as well as memory space) to compute and store a dense linear system of equations and $\mathcal{O}(N^2)$ operations per iteration to solve this system using an iterative solver. But this computational cost can be reduced by invoking fast methods, e.g., FMM, low-rank approximation, FFT-based [71] and multigrid-based algorithms [68]. Most of these methods have been successfully demonstrated to reduce the memory and computational costs of MoM for almost uniform discretizations. But, when the object possesses fine features, and needs to be discretized with multi-sized meshes (to reduce the number of unknowns) or when it resides in a layered-medium [68], most of these fast algorithms break down or at least lose efficiency, and must be modified [23],[72],[73] in order to reduce the computational complexity.

In this dissertation and up to this chapter, we focus on the scattering problems for homogeneous dielectric objects, although the developed codes and methods may be directly used for the analysis of inhomogeneous dielectric objects as well [74],[75],[76]. For these inhomogeneous objects, the solution to the scattering problem is usually formulated with different versions of the electric-field volume integral equation (EFVIE). EFVIE can be formulated either for the electric flux, "D-formulation", or for the electric field , "E-formulation" [77],[78]. The major difference in the D- and E-formulations is the choice of the basis functions used to represent the unknowns. In the D-formulation, the unknown $\boldsymbol{D}$ has continuous normal component on the element boundaries, whereas in the E-formulation, the unknown $\boldsymbol{E}$ has continuous tangential component on the boundaries. Consequently, to satisfy continuity properties, the D-formulation should be discretized with divergence-conforming basis functions and the E-formulation with curl-conforming ones.

Both formulations of EFVIE have a double derivative operator, i.e., $\nabla\nabla$. For numerically accurate solutions, the singularity may be reduced by moving one of the derivatives away from the singular Green's function onto the testing functions. This means that, a proper choice for a testing function is quite important to obtain an efficient numerical method. By discretizing the D-formulation with Galerkin's method using divergence-conforming basis and testing functions, the gradient can be moved onto the testing function and a boundary integral appears on the outer surface of the object only. However, in the case of the E-formulation, discretized with Galerkin's method using curl-conforming basis and testing functions, we end up with boundary integrals on the surfaces of all elements of the mesh. To avoid the additional boundary integrals one can rewrite the EFVIE in an alternative form containing a double curl operator [79]. This formulation requires the curl of the testing functions rather than the divergence, and hence, with curl-conforming testing functions the boundary integrals appear only on the outer surface of the object.

The volume integral equations may be formulated for the magnetic field, giving the magnetic-field volume integral equation (MFVIE). Botha [77] investigated the solution of MFVIE and EFVIE (D-formulation), and revealed that they give almost the same accuracy considering electrically small and low contrast objects. Scattering by small and low contrast magnetic objects with an alternative electric formulation dual to MFVIE has been proposed by Sun and Chew in [80]. The method can also be generalized for those scatterers where both the permittivity and permeability differ from the background medium [81], [82]. All of these new formulations contain two unknowns ($\boldsymbol{D}$ and $\boldsymbol{B}$ or $\boldsymbol{E}$ and $\boldsymbol{H}$) and two equations. However, it is possible to derive formulations that contain only a single unknown [77]. The VIE method has also been generalized for bi-isotropic [83] and anisotropic scatterers [84]. It is also possible to combine VIEs with SIEs to analyze more complicated inhomogeneous composite objects. Recently, a new formulation was proposed for VIEs in terms of the volume equivalent currents by employing piecewise constant basis functions, which do not impose continuity of either normal flux or the tangential fields [85].

In this dissertation we use the well-known D-formulations to solve the time-harmonic electromagnetic scattering by low to medium contrast penetrable objects of arbitrary shape. The formulation is discretized with the Galerkin's method using divergence-conforming surface elements known as Schaubert-Wilton-Glisson (SWG) functions [75]. To accelerate the MVM operations, we use MLFMA with IL tree structures. Our numerical results demonstrate that the accuracy and efficiency of EFVIE can be significantly improved by incorporating the newly developed IL concept.

6.2 Volume Integral Equations

6.2.1 The Volume Equivalence Principle

We commence by considering the Maxwell equations in phasor form as

$$\nabla \times \boldsymbol{E}(\boldsymbol{r}) = iw\boldsymbol{B}(\boldsymbol{r}) - \boldsymbol{M}_{\mathrm{src}}(\boldsymbol{r}) \tag{6.1}$$
$$\nabla \times \boldsymbol{H}(\boldsymbol{r}) = -iw\boldsymbol{D}(\boldsymbol{r}) + \boldsymbol{J}_{\mathrm{src}}(\boldsymbol{r}) \tag{6.2}$$
$$\boldsymbol{D}(\boldsymbol{r}) = \epsilon(\boldsymbol{r})\boldsymbol{E}(\boldsymbol{r}) \tag{6.3}$$
$$\boldsymbol{B}(\boldsymbol{r}) = \mu(\boldsymbol{r})\boldsymbol{H}(\boldsymbol{r}), \tag{6.4}$$

where the impressed electric and magnetic volume current sources $\boldsymbol{J}_{\mathrm{src}}$ and $\boldsymbol{M}_{\mathrm{src}}$ provide the excitation. The permittivity and permeability may be functions of position, but the background medium is considered to be the homogeneous free space with constants ϵ_0 and μ_0.

The set of equations (6.1)–(6.4) can be reformulated by invoking the volume equivalence principle, which states that field-dependent sources may be introduced into Maxwell's equations in such a way that an equivalent set of equations may be obtained with the same field solution, but now existing in free space [58]. By eliminating $\boldsymbol{D}$ and $\boldsymbol{B}$ from (6.1)–(6.2), and rewriting the resulting equations as

$$\nabla \times \boldsymbol{E}(\boldsymbol{r}) = iw\mu_0\boldsymbol{H}(\boldsymbol{r}) - \boldsymbol{M}_{\mathrm{src}}(\boldsymbol{r}) + iw\mu_0(\mu_r(\boldsymbol{r}) - 1)\boldsymbol{H}(\boldsymbol{r}) \tag{6.5}$$
$$\nabla \times \boldsymbol{H}(\boldsymbol{r}) = -iw\epsilon_0\boldsymbol{E}(\boldsymbol{r}) + \boldsymbol{J}_{\mathrm{src}}(\boldsymbol{r}) - iw\epsilon_0(\epsilon_r(\boldsymbol{r}) - 1)\boldsymbol{E}(\boldsymbol{r}), \tag{6.6}$$

one can define the equivalent volume (polarization) current sources as

$$\boldsymbol{J}_{\mathrm{eq}}(\boldsymbol{r}) = -iw\epsilon_0(\epsilon_r(\boldsymbol{r}) - 1)\boldsymbol{E}(\boldsymbol{r}) \tag{6.7}$$
$$\boldsymbol{M}_{\mathrm{eq}}(\boldsymbol{r}) = -iw\mu_0(\mu_r(\boldsymbol{r}) - 1)\boldsymbol{H}(\boldsymbol{r}), \tag{6.8}$$

to obtain a new set of free space equations with the same solution as (6.1)–(6.4):

$$\nabla \times \boldsymbol{E}(\boldsymbol{r}) = iw\boldsymbol{B}(\boldsymbol{r}) - \boldsymbol{M}_{\text{src}}(\boldsymbol{r}) - \boldsymbol{M}_{\text{eq}}(\boldsymbol{r}) \tag{6.9}$$

$$\nabla \times \boldsymbol{H}(\boldsymbol{r}) = -iw\boldsymbol{D}(\boldsymbol{r}) + \boldsymbol{J}_{\text{src}}(\boldsymbol{r}) + \boldsymbol{J}_{\text{eq}}(\boldsymbol{r}). \tag{6.10}$$

Finally, decomposing the total electric and magnetic fields into incident fields ($\boldsymbol{E}^{\text{inc}}$, $\boldsymbol{H}^{\text{inc}}$) and scattered fields ($\boldsymbol{E}^{\text{scat}}$, $\boldsymbol{H}^{\text{scat}}$), and considering the fact that the effects of the impressed sources $\boldsymbol{J}_{\text{src}}$ and $\boldsymbol{M}_{\text{src}}$ can be consolidated into the incident field terms, we may arrive at the following equations regarding solely the scattered fields

$$\nabla \times \boldsymbol{E}^{\text{scat}}(\boldsymbol{r}) = iw\mu_0\boldsymbol{H}^{\text{scat}}(\boldsymbol{r}) - \boldsymbol{M}_{\text{eq}}(\boldsymbol{r}) \tag{6.11}$$

$$\nabla \times \boldsymbol{H}^{\text{scat}}(\boldsymbol{r}) = -iw\epsilon_0\boldsymbol{E}^{\text{scat}}(\boldsymbol{r}) + \boldsymbol{J}_{\text{eq}}(\boldsymbol{r}). \tag{6.12}$$

6.2.2 VIE Formulations

Solutions of equations (6.11)–(6.12) using equivalent current sources in a homogeneous background medium are straightforward and they are given by (2.27)–(2.28). Using k_0 and η_0 for the background medium and the definitions of $\mathcal{T}$ and $\mathcal{K}$ operators, the total electric and magnetic fields, i.e., incident fields plus the scattered fields, can be written as

$$\begin{bmatrix} \boldsymbol{E} \\ \boldsymbol{H} \end{bmatrix} = \begin{bmatrix} \boldsymbol{E}^{\text{inc}} \\ \boldsymbol{H}^{\text{inc}} \end{bmatrix} + \begin{bmatrix} ik_0\eta_0 \left(\bar{\bar{\boldsymbol{I}}} + \frac{1}{k_0^2}\nabla\nabla\cdot\right)\boldsymbol{S} & -\nabla \times \boldsymbol{S} \\ \nabla \times \boldsymbol{S} & \frac{ik_0}{\eta_0}\left(\bar{\bar{\boldsymbol{I}}} + \frac{1}{k_0^2}\nabla\nabla\cdot\right)\boldsymbol{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{J}_{\text{eq}} \\ \boldsymbol{M}_{\text{eq}} \end{bmatrix}, \tag{6.13}$$

known as electric- and magnetic-field VIEs where, $\bar{\bar{\boldsymbol{I}}}$ is the identity dyadic operator and $\boldsymbol{S}$ is the volume integral operator defined as

$$\boldsymbol{S}\{\boldsymbol{F}\}(\boldsymbol{r}) = \int_{\text{V}} d\boldsymbol{r}' g_0(\boldsymbol{r}, \boldsymbol{r}')\boldsymbol{F}(\boldsymbol{r}'). \tag{6.14}$$

We briefly consider three different types of formulations, the DB-formulation using the flux densities as unknowns, the EH-formulation using the fields as unknowns, and the single volume integral equations, EE- and HH-formulations, that contain either the electric field and its curl, or the magnetic field and its curl.

6.2.2.1 DB-Formulation

The volume integral equation formulation regarding the fluxes, i.e., $\boldsymbol{D}$ and $\boldsymbol{B}$, can be further simplified by replacing the equivalent currents in (6.13) with the flux densities as

$$\boldsymbol{J}_{\text{eq}}(\boldsymbol{r}) = -iw\kappa_E(\boldsymbol{r})\boldsymbol{D}(\boldsymbol{r}) \tag{6.15}$$

$$\boldsymbol{M}_{\text{eq}}(\boldsymbol{r}) = -iw\kappa_H(\boldsymbol{r})\boldsymbol{B}(\boldsymbol{r}), \tag{6.16}$$

leading us to the DB-formulation given by

$$\begin{bmatrix} \frac{1}{\epsilon_r}\boldsymbol{D} \\ \frac{1}{\mu_r}\boldsymbol{B} \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}^{\text{inc}} \\ \boldsymbol{B}^{\text{inc}} \end{bmatrix} + \begin{bmatrix} k_0^2\left(\bar{\bar{\boldsymbol{I}}} + \frac{1}{k_0^2}\nabla\nabla\cdot\right)\boldsymbol{S} & iw\epsilon_0\nabla\times\boldsymbol{S} \\ -iw\mu_0\nabla\times\boldsymbol{S} & k_0^2\left(\bar{\bar{\boldsymbol{I}}} + \frac{1}{k_0^2}\nabla\nabla\cdot\right)\boldsymbol{S} \end{bmatrix} \begin{bmatrix} \kappa_E\boldsymbol{D} \\ \kappa_H\boldsymbol{B} \end{bmatrix}. \tag{6.17}$$

In (6.17), κ_E and κ_H defined as

$$\kappa_E(\boldsymbol{r}) = 1 - \frac{1}{\epsilon_r(\boldsymbol{r})} \tag{6.18}$$

$$\kappa_H(\boldsymbol{r}) = 1 - \frac{1}{\mu_r(\boldsymbol{r})}, \tag{6.19}$$

are the permittivity and permeability contrast functions, respectively. Since the DB-formulation in (6.17) uses the fluxes as unknowns, it should be solved with divergence-conforming basis functions maintaining the normal continuity on the boundaries.

6.2.2.2 EH-Formulation

The VIE formulation may also be written for the fields $\boldsymbol{E}$ and $\boldsymbol{H}$. Using

$$\boldsymbol{J}_{\text{eq}}(\boldsymbol{r}) = -iw\epsilon_0\tau(\boldsymbol{r})\boldsymbol{E}(\boldsymbol{r}) \tag{6.20}$$

$$\boldsymbol{M}_{\text{eq}}(\boldsymbol{r}) = -iw\mu_0\tau_H(\boldsymbol{r})\boldsymbol{H}(\boldsymbol{r}), \tag{6.21}$$

one may arrive at the EH-formulation as

$$\begin{bmatrix}\boldsymbol{E}\\ \boldsymbol{H}\end{bmatrix} = \begin{bmatrix}\boldsymbol{E}^{\text{inc}}\\ \boldsymbol{H}^{\text{inc}}\end{bmatrix} + \begin{bmatrix} k_0^2\left(\bar{\bar{\boldsymbol{I}}} + \frac{1}{k_0^2}\nabla\nabla\cdot\right)\boldsymbol{S} & iw\mu_0\nabla\times\boldsymbol{S} \\ -iw\epsilon_0\nabla\times\boldsymbol{S} & k_0^2\left(\bar{\bar{\boldsymbol{I}}} + \frac{1}{k_0^2}\nabla\nabla\cdot\right)\boldsymbol{S}\end{bmatrix}\begin{bmatrix}\tau_E\boldsymbol{E}\\ \tau_H\boldsymbol{H}\end{bmatrix}. \tag{6.22}$$

Similarly,

$$\tau_E(\boldsymbol{r}) = \epsilon_r(\boldsymbol{r}) - 1 \tag{6.23}$$

$$\tau_H(\boldsymbol{r}) = \mu_r(\boldsymbol{r}) - 1, \tag{6.24}$$

are two other permittivity and permeability contrast functions, respectively. Considering the continuity of the tangential components of the fields on the boundaries, it is more reasonable to expand $\boldsymbol{E}$ and $\boldsymbol{H}$ fields with the curl-conforming basis functions. Since (6.22) involves the divergence of basis functions rather than the curl, it is not computationally a good choice for curl-conforming functions. To overcome this problem, one can use the identity

$$\nabla\times\nabla\times\boldsymbol{F} = \left(\nabla\nabla\cdot - \nabla^2\right)\boldsymbol{F} \tag{6.25}$$

and the fact that the volume integral operator defined in (6.14) satisfies the Helmholtz equation [78] to obtain

$$\left(\nabla\nabla\cdot + k_0^2\right)\boldsymbol{S}\left\{\boldsymbol{F}\right\} = \nabla\times\nabla\times\,\boldsymbol{S}\left\{\boldsymbol{F}\right\} - \boldsymbol{F}. \tag{6.26}$$

Substitution of (6.26) into (6.22) results in a more suitable form of this formulation as

$$\begin{bmatrix} \epsilon_r \boldsymbol{E} \\ \mu_r \boldsymbol{H} \end{bmatrix} = \begin{bmatrix} \boldsymbol{E}^{\text{inc}} \\ \boldsymbol{H}^{\text{inc}} \end{bmatrix} + \begin{bmatrix} \nabla \times \nabla \times \boldsymbol{S} & iw\mu_0 \nabla \times \boldsymbol{S} \\ -iw\epsilon_0 \nabla \times \boldsymbol{S} & \nabla \times \nabla \times \boldsymbol{S} \end{bmatrix} \begin{bmatrix} \tau_E \boldsymbol{E} \\ \tau_H \boldsymbol{H} \end{bmatrix}. \tag{6.27}$$

6.2.2.3 EE- and HH- Formulations

By using the Maxwell's curl equations outside the source region, we can manipulate the last formulation to separate the unknowns as

$$\epsilon_r \boldsymbol{E} = \boldsymbol{E}^{\text{inc}} + \begin{bmatrix} \nabla \times \nabla \times \boldsymbol{S} & \nabla \times \boldsymbol{S} \end{bmatrix} \begin{bmatrix} \tau_E \boldsymbol{E} \\ \kappa_H \nabla \times \boldsymbol{E} \end{bmatrix}, \tag{6.28}$$

known as the EE-formulation and

$$\mu_r \boldsymbol{H} = \boldsymbol{H}^{\text{inc}} + \begin{bmatrix} \nabla \times \boldsymbol{S} & \nabla \times \nabla \times \boldsymbol{S} \end{bmatrix} \begin{bmatrix} \kappa_E \nabla \times \boldsymbol{H} \\ \tau_H \boldsymbol{H} \end{bmatrix}, \tag{6.29}$$

known as the HH-formulation. Note that both formulations need to be discretized with curl-conforming functions due to use of $\boldsymbol{E}$ and $\boldsymbol{H}$ fields.

6.2.2.4 Formulations for Inhomogeneous Dielectric Objects

For an inhomogeneous dielectric object, $\mu_r(\boldsymbol{r}) = 1$, and hence, $\tau_H = 0$ and $\kappa_H = 0$. In this case, the first formulation in (6.17) separates into two equations as

$$\frac{1}{\epsilon_r} \boldsymbol{D} = \boldsymbol{D}^{\text{inc}} + k_0^2 \left(\bar{\bar{\boldsymbol{I}}} + \frac{1}{k_0^2} \nabla \nabla \cdot \right) \boldsymbol{S} \left\{ \kappa_E \boldsymbol{D} \right\}, \tag{6.30}$$

$$\boldsymbol{B} = \boldsymbol{B}^{\text{inc}} - iw\mu_0 \nabla \times \boldsymbol{S} \left\{ \kappa_E \boldsymbol{D} \right\}. \tag{6.31}$$

Similarly, (6.27) separates into two smaller equations given by

$$\epsilon_r \boldsymbol{E} = \boldsymbol{E}^{\text{inc}} + \nabla \times \nabla \times \, \boldsymbol{S}\left\{\tau_E \boldsymbol{E}\right\} \tag{6.32}$$

$$\boldsymbol{H} = \boldsymbol{H}^{\text{inc}} - iw\epsilon_0 \nabla \times \boldsymbol{S}\left\{\tau_E \boldsymbol{E}\right\}, \tag{6.33}$$

and (6.28) and (6.29) reduce to (6.32) and

$$\boldsymbol{H} = \boldsymbol{H}^{\text{inc}} + \nabla \times \boldsymbol{S}\left\{\kappa_E \nabla \times \boldsymbol{H}\right\}, \tag{6.34}$$

respectively. The formulations in (6.31) and (6.33) are not useful due to the existence of two unknowns. Note that expressing $\boldsymbol{B}$ in terms of $\boldsymbol{D}$ in (6.31) requires taking the curl of a divergence-conforming function. Also note that casting $\boldsymbol{E}$ in terms of $\boldsymbol{H}$ in (6.33) leads us to (6.34).

Finally for an inhomogeneous dielectric object, we end up with three computational-friendly formulations (6.30), (6.32), and (6.34). These formulations are called electric flux volume integral equation (EDVIE or EFVIE-D) or the D-formulation, electric-field volume integral equation (EFVIE or EFVIE-E) or the E-formulation, and the magnetic-field volume integral equation (MFVIE or MFVIE-H) or the H-formulation, respectively. More details and comparison between all formulations can be found in [86].

Clearly, all of them fall into the category of the second kind Fredholm integral equations. This means that all three formulations are in the form of $(\mathcal{L}+\mathcal{I})\left\{\boldsymbol{F}\right\} = \boldsymbol{R}$ where $\mathcal{L}$ is a linear integral operator, $\mathcal{I}$ is the identity operator, $\boldsymbol{F}$ is an unknown function to be found, and $\boldsymbol{R}$ is a known right-hand side function. It is worth mentioning that the MFIE formulation we use for SIE in the previous chapter was also of this kind. This is beneficial due to the fact that these integral equations lead us to a better conditioned matrix equation than the first kind [1].

Hereafter we proceed by using the formulation given in (6.30) along with divergence-conforming SWG functions to implement IL version of EFVIE-MLFMA.

6.3 SWG Functions

Let V denote the volume of an inhomogeneous dielectric object with a position dependent dielectric constant of $\epsilon(\boldsymbol{r})$. V is subdivided into a number of tetrahedral elements such that an inhomogeneous dielectric region is approximated by a proper number of tetrahedrons, in each of which the dielectric properties are approximated as constant. Each of these tetrahedrons acts as a homogeneous dielectric region, which is bounded by a surface that is approximated by simple triangular faces. After properly modeling the volume of the scatterer object by tetrahedral elements, the faces of the tetrahedrons are of primary importance for the development of the basis functions. Fig. 6.1 shows a pair of tetrahedra, D_n^+

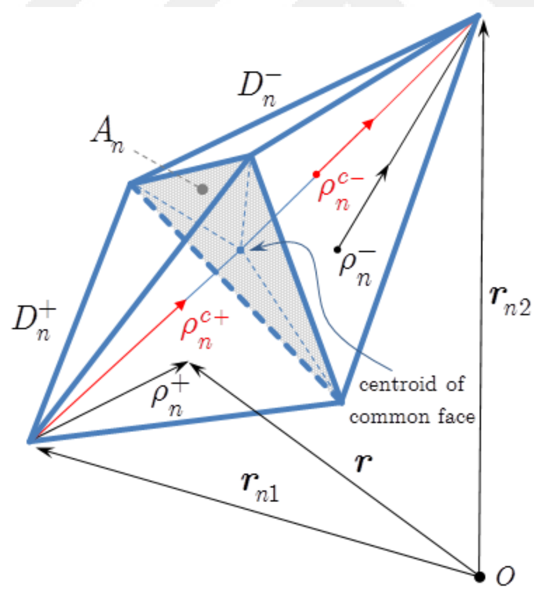


Figure 6.1: A pair of tetrahedra as a full SWG function and geometrical parameters associated with nth face. $\rho_n^{c\pm}$ are centroid of coresponding tetrahedrons.

and D_n^- associated with the nth face of the subdivided region V modeling the scatterer object. Any point inside D_n^+ may be addressed either by the position vector $\boldsymbol{r}$ defined with respect to the origin, or by the position vector ρ_n^+, defined with respect to the free vertex of D_n^+. Similar definition may apply for the points in D_n^- and the position vector ρ_n^-. Note that the former is directed from the free vertex of D_n^+ toward the common face, but the latter is directed from the common face toward the free vertex of D_n^-. The plus or minus designation of a tetrahedron is determined by the choice of a positive flux direction, which is assumed to be from D_n^+ to D_n^-. Each SWG function is defined as a pair of tetrahedra sharing a common face. The nth SWG basis function with a spatial support of $D_n = D_{n1} + D_{n2}$ can be written as

$$\boldsymbol{b}_n^{SWG}(\boldsymbol{r}) = \begin{cases} \dfrac{A_n}{3V_{n1}}(\boldsymbol{r} - \boldsymbol{r}_{n1}), & \boldsymbol{r} \in D_{n1} \\ \dfrac{A_n}{3V_{n2}}(\boldsymbol{r}_{n2} - \boldsymbol{r}), & \boldsymbol{r} \in D_{n2} \\ 0, & \boldsymbol{r} \notin D_n, \end{cases} \tag{6.35}$$

where A_n represents the area of the common face, whereas V_{n1} and V_{n2} are volumes of the first and second tetrahedra, respectively. In (6.35), r_{n1} and r_{n2} are the coordinates of the two vertices opposite to the common face. For numerical solutions of (6.30), the flux density $\boldsymbol{D}$ (or later on $iw\boldsymbol{D}$) can be expanded by a set of SWG basis functions, i.e.,

$$\boldsymbol{D}(\boldsymbol{r}) = \sum_{n=1}^{N} a_n \boldsymbol{b}_n(\boldsymbol{r}). \tag{6.36}$$

Using SWGs as the basis functions for $\boldsymbol{D}(\boldsymbol{r})$ has the following properties that make them useful for representing [75]:

1. Inside each tetrahedron, $\boldsymbol{D}(\boldsymbol{r})$ is the vectorial sum of four linearly independent basis functions, one associated with each face, which can represent any constant vector in any direction.

2. $\boldsymbol{b}_n(\boldsymbol{r})$ is tangential to three side faces and has only a normal component on the common face of the conjoined pair D_n^+ and D_n^-.

3. The component of $\boldsymbol{b}_n(\boldsymbol{r})$ normal to the nth face is continuous across the face and it is constant, because the normal component of $\rho_n^{\pm}$ along the common face is just the height of $D_n^{\pm}$, which is equal to $3V_n^{\pm}/A_n$ with the common face n being the base. This is essential in order to normalize $\boldsymbol{b}_n(\boldsymbol{r})$ in (6.35) such that its flux density normal to face n is unity ensuring the continuity of the $\boldsymbol{D}$ normal to the face.

4. The charge density represented by the nth SWG function can be derived as

$$\nabla \cdot \boldsymbol{b}_n^{SWG}(\boldsymbol{r}) = \begin{cases} \dfrac{A_n}{V_{n1}}, & \boldsymbol{r} \in D_{n1} \\ -\dfrac{A_n}{V_{n2}}, & \boldsymbol{r} \in D_{n2} \\ 0, & \boldsymbol{r} \notin D_n. \end{cases} \tag{6.37}$$

   Note that the charge density is constant within each tetrahedron.

5. The expansion coefficient a_n in (6.36) represents the normal component of $\boldsymbol{D}$ over the nth face.

6. The moment of $\boldsymbol{b}_n(\boldsymbol{r})$ over $D_n^{\pm}$ is

$$\int_{D_n^{\pm}} d\boldsymbol{r'}\boldsymbol{b}_n(\boldsymbol{r'}) = \frac{A_n}{3}\rho_n^{c\pm}, \tag{6.38}$$

   where $\rho_n^{c\pm}$ are the vectors between the free vertices and the centroid of each tetrahedron directed toward and away from the common face (Fig. 6.1), respectively.

Note that if the nth face is on the boundary of V, then only one of the tetrahedrons, D_n^+ or D_n^- is interior to V. In this case only a half SWG function is defined regarding the interior tetrahedron.

6.4 Discretization

In order to carry out a code-friendly formulation, the double derivative operator, i.e., $\nabla\nabla$ in (6.30) should be handled properly. The detailed procedure is as follow: First, as a shorter notation, we replace κ_E with κ for all up-coming formulations. Then by substituting $\kappa(\boldsymbol{r})\boldsymbol{D}(\boldsymbol{r})$ into (6.14), which is the definition of $\boldsymbol{S}$ operator, we may rewrite (6.30) or the D-formulation as

$$\frac{\boldsymbol{D}(\boldsymbol{r})}{\epsilon_0\epsilon_r(\boldsymbol{r})} + ik_0\eta_0 \int_D d\boldsymbol{r'} iw\kappa(\boldsymbol{r'})\boldsymbol{D}(\boldsymbol{r'})g_0(\boldsymbol{r},\boldsymbol{r'})$$
$$- \frac{i\eta_0}{k_0}\nabla \int_D d\boldsymbol{r'} iw\kappa(\boldsymbol{r'})\boldsymbol{D}(\boldsymbol{r'})\cdot\nabla' g_0(\boldsymbol{r},\boldsymbol{r'}) = \boldsymbol{E}^{\mathrm{inc}}(\boldsymbol{r}). \tag{6.39}$$

Expanding $iw\boldsymbol{D}$ by a set of basis functions, i.e., $iw\boldsymbol{D}(\boldsymbol{r}) = \sum_{n=1}^{N} a_n\boldsymbol{b}_n(\boldsymbol{r})$, and testing the resultant equation by a set of testing functions $\boldsymbol{t}_n(\boldsymbol{r})$ for $m = 1, 2, \cdots, N$, leads us to an $N \times N$ dense matrix equation in the form of

$$\bar{\boldsymbol{Z}}^{\text{EFVIE-D}} \cdot \boldsymbol{a} = \boldsymbol{w}^{\text{EFVIE-D}}, \tag{6.40}$$

where the matrix elements are defined as [35]

$$\begin{aligned}\bar{\boldsymbol{Z}}^{\text{EFVIE-D}}[m,n] = & \frac{1}{iw\epsilon_0}\int_{D_m} d\boldsymbol{r}\,(1-\kappa(\boldsymbol{r}))\,\boldsymbol{t}_m(\boldsymbol{r})\cdot\boldsymbol{b}_n(\boldsymbol{r}) \\ & + iw\mu_0\int_{D_m} d\boldsymbol{r}\,\boldsymbol{t}_m(\boldsymbol{r})\cdot\int_{D_n} d\boldsymbol{r'}\,\kappa(\boldsymbol{r'})\boldsymbol{b}_n(\boldsymbol{r'})g_0(\boldsymbol{r},\boldsymbol{r'}) \\ & + \frac{1}{iw\epsilon_0}\int_{D_m} d\boldsymbol{r}\,\boldsymbol{t}_m(\boldsymbol{r})\cdot\nabla\int_{D_n} d\boldsymbol{r'}\kappa(\boldsymbol{r'})\boldsymbol{b}_n(\boldsymbol{r'})\cdot\nabla' g_0(\boldsymbol{r},\boldsymbol{r'}),\end{aligned} \tag{6.41}$$

and the right-hand-side (RHS) vector is given by

$$\boldsymbol{w}^{\text{EFVIE-D}}[m] = \int_{D_m} d\boldsymbol{r}\,\boldsymbol{t}_m(\boldsymbol{r})\cdot\boldsymbol{E}^{\mathrm{inc}}(\boldsymbol{r}), \tag{6.42}$$

respectively. Now by considering the interaction of two half SWG functions, i.e., $\boldsymbol{t}_{ma}$ and $\boldsymbol{b}_{nb}$ associated with the ath tetrahedron of the mth face and bth tetrahedron of the nth face, assuming a constant permittivity within each tetrahedron, and using (6.35), (6.41) can be written as [35]

$$\bar{\boldsymbol{Z}}^{\text{EFVIE-D}}[m,n,a,b] = \frac{\mathcal{C}_{ma,nb}\,\delta_{ma,nb}(1-\kappa_{nb})}{iw\epsilon_0}\int_{D_{ma}} d\boldsymbol{r}\ (\boldsymbol{r}-\boldsymbol{r}_{ma})\cdot(\boldsymbol{r}-\boldsymbol{r}_{nb})$$
$$+\, iw\mu_0\kappa_{nb}\,\mathcal{C}_{ma,nb}\int_{D_{ma}} d\boldsymbol{r}\,(\boldsymbol{r}-\boldsymbol{r}_{ma})\cdot\int_{D_{nb}} d\boldsymbol{r'}\,(\boldsymbol{r'}-\boldsymbol{r}_{nb})g_0(\boldsymbol{r},\boldsymbol{r'})$$
$$+\,\frac{\kappa_{nb}\,\mathcal{C}_{ma,nb}}{iw\epsilon_0}\int_{D_{ma}} d\boldsymbol{r}\,(\boldsymbol{r}-\boldsymbol{r}_{ma})\cdot\nabla\int_{D_{nb}} d\boldsymbol{r'}(\boldsymbol{r'}-\boldsymbol{r}_{nb})\cdot\nabla' g_0(\boldsymbol{r},\boldsymbol{r'}), \tag{6.43}$$

where $\delta_{i,j}$ is the Kronecker's delta function, which is 1 for $i=j$ and 0 otherwise, and

$$\mathcal{C}_{ma,nb} = \frac{A_m A_n}{9V_{ma}V_{nb}}\gamma_{ma}\gamma_{nb}, \tag{6.44}$$

where $\gamma_{ma}, \gamma_{nb} = \pm 1$, depending on the direction of the basis and testing functions in each tetrahedron. To reduce the singularity in the last term of (6.43), we can use the identity $\boldsymbol{F}\cdot\nabla' g = \nabla'\cdot(\boldsymbol{F}g) - g\nabla'\cdot\boldsymbol{F}$, and write it as [35]

$$\int_{D_{nb}} d\boldsymbol{r'}(\boldsymbol{r'}-\boldsymbol{r}_{nb})\cdot\nabla' g_0(\boldsymbol{r},\boldsymbol{r'}) = \int_{D_{nb}} d\boldsymbol{r'}\nabla'\cdot((\boldsymbol{r'}-\boldsymbol{r}_{nb})g_0(\boldsymbol{r},\boldsymbol{r'}))$$
$$-\int_{D_{nb}} d\boldsymbol{r'} g_0(\boldsymbol{r},\boldsymbol{r'})\nabla'\cdot(\boldsymbol{r'}-\boldsymbol{r}_{nb}). \tag{6.45}$$

Using the divergence theorem for the first integral, in the right-hand-side of (6.45), the expression simplifies as [35]

$$\int_{D_{nb}} d\boldsymbol{r'}(\boldsymbol{r'}-\boldsymbol{r}_{nb})\cdot\nabla' g_0(\boldsymbol{r},\boldsymbol{r'}) = \int_{\partial D_{nb}} \hat{\boldsymbol{n}}'\,ds'\cdot(\boldsymbol{r'}-\boldsymbol{r}_{nb})\,g_0(\boldsymbol{r},\boldsymbol{r'})$$
$$-\,3\int_{D_{nb}} d\boldsymbol{r'} g_0(\boldsymbol{r},\boldsymbol{r'}), \tag{6.46}$$

where ∂D_{nb} represents the surface of the four faces of the tetrahedron D_{nb}, consisting of three side faces and a front face (with respect to the free vertex $\boldsymbol{r}_{nb}$), and $\hat{\boldsymbol{n}}'$ is the unit vector normal to these faces. Recalling the second and the third properties of the SWG functions, only the front (common) face contributes.

Hence, one can arrive at [35]

$$\int_{D_{nb}} d\boldsymbol{r'}(\boldsymbol{r'} - \boldsymbol{r}_{nb}) \cdot \nabla' g_0(\boldsymbol{r}, \boldsymbol{r'}) = \frac{3V_{nb}}{A_n} \int_{S_n} d\boldsymbol{r'}\, g_0(\boldsymbol{r}, \boldsymbol{r'}) - 3 \int_{D_{nb}} d\boldsymbol{r'} g_0(\boldsymbol{r}, \boldsymbol{r'}), \tag{6.47}$$

where S_n is the surface of the front (common) face. Now, we can insert (6.47) into the last term of (6.43) and use the divergence theorem one more time. For convenience, we repeat the formula here and proceed with more details.

$$\int_{D_{ma}} d\boldsymbol{r}\, \boldsymbol{F} \cdot \nabla\psi \equiv \int_{D_{ma}} d\boldsymbol{r}\, \overbrace{(\boldsymbol{r} - \boldsymbol{r}_{ma})}^{\boldsymbol{F}} \cdot \nabla \overbrace{\left\{ \frac{3V_{nb}}{A_n} \int_{S_n} d\boldsymbol{r'}\, g_0(\boldsymbol{r}, \boldsymbol{r'}) - 3 \int_{D_{nb}} d\boldsymbol{r'} g_0(\boldsymbol{r}, \boldsymbol{r'}) \right\}}^{\psi}$$

$$\begin{aligned} \int_{D_{ma}} d\boldsymbol{r}\, \boldsymbol{F} \cdot \nabla\psi &= \int_{D_{ma}} d\boldsymbol{r}\, \nabla \cdot (\boldsymbol{F}\, \psi) - \int_{D_{ma}} d\boldsymbol{r}\, \psi\, \nabla \cdot \boldsymbol{F} \\ &= \int_{S_m} \hat{\boldsymbol{n}}\, ds\, \cdot (\boldsymbol{F}\, \psi) - 3 \int_{D_{ma}} d\boldsymbol{r}\, \psi = \frac{3V_{ma}}{A_m} \int_{S_m} d\boldsymbol{r}\, \psi - 3 \int_{D_{ma}} d\boldsymbol{r}\, \psi. \end{aligned} \tag{6.48}$$

Note that the properties of SWG functions are used one more time to arrive at the above simplified formula. It is clear that both equations (6.47) and (6.48) are similar except that the former is carried out for basis functions and the latter for testing functions. On the other hand, the second ∇ operator has been moved onto the testing function to reduce the singularity. Finally, replacing ψ by its equivalent term defined above, substituting it into (6.43), and factorizing the whole equation, one can derive the final formulation for the matrix elements as follows [35]

$$
\begin{aligned}
\bar{\boldsymbol{Z}}^{\text{EFVIE-D}}[m,n,a,b] = &\frac{\mathcal{C}_{ma,nb}\,\delta_{ma,nb}}{iw\epsilon_0\epsilon_{r,nb}}\int_{D_{ma}} d\boldsymbol{r}\ (\boldsymbol{r}-\boldsymbol{r}_{ma})\cdot(\boldsymbol{r}-\boldsymbol{r}_{nb}) \\
&+ iw\mu_0\kappa_{nb}\,\mathcal{C}_{ma,nb}\int_{D_{ma}} d\boldsymbol{r}\,(\boldsymbol{r}-\boldsymbol{r}_{ma})\cdot\int_{D_{nb}} d\boldsymbol{r'}\,(\boldsymbol{r'}-\boldsymbol{r}_{nb})g_0(\boldsymbol{r},\boldsymbol{r'}) \\
&+ \frac{9\,\mathcal{C}_{ma,nb}\,\kappa_{nb}}{iw\epsilon_0}\int_{D_{ma}} d\boldsymbol{r}\int_{D_{nb}} d\boldsymbol{r'}\,g_0(\boldsymbol{r},\boldsymbol{r'}) \\
&- \frac{A_m\gamma_{nb}\gamma_{ma}\kappa_{nb}}{iw\epsilon_0\,V_{ma}}\int_{D_{ma}} d\boldsymbol{r}\int_{S_n} d\boldsymbol{r'}\,g_0(\boldsymbol{r},\boldsymbol{r'}) \\
&- \frac{A_n\gamma_{nb}\gamma_{ma}\kappa_{nb}}{iw\epsilon_0\,V_{nb}}\int_{S_m} d\boldsymbol{r}\int_{D_{nb}} d\boldsymbol{r'}\,g_0(\boldsymbol{r},\boldsymbol{r'}) \\
&+ \frac{\gamma_{nb}\gamma_{ma}\kappa_{nb}}{iw\epsilon_0}\int_{S_m} d\boldsymbol{r}\int_{S_n} d\boldsymbol{r'}\,g_0(\boldsymbol{r},\boldsymbol{r'}).
\end{aligned} \tag{6.49}
$$

Similarly, the RHS vector is evaluated as [35]

$$
\boldsymbol{w}^{\text{EFVIE-D}}[m,a] = \frac{\gamma_{ma}A_m}{3V_{ma}}\int_{D_{ma}} d\boldsymbol{r}\,(\boldsymbol{r}-\boldsymbol{r}_{ma})\cdot\boldsymbol{E}^{\text{inc}}(\boldsymbol{r}). \tag{6.50}
$$

It is clear that accurate evaluation of surface and volume integrals in (6.49) and (6.50) are very important from the computational point of view. This is more critical due to the singularity of the Green's function. Therefore, we have to use proper singularity extraction routines for inner surface and volume integrals regarding the last five terms of (6.49). Among different methods detailed in [87], [49], and [50], we will use [49], which is quite similar to what we have utilized for SIE routines.

Considering outer integrals, one may use Gaussian quadrature rules with different orders or even fully analytical methods as introduced in [59]. Here, we use the second order quadrature for near-field calculations, although single order is also possible and quite usual. The main reason to use the second order quadrature is due to the fact that in IL tree structure we encounter wide range of tetrahedra and triangles with orders of magnitude difference in their sizes. Therefore, it seems much wiser to use higher order quadratures to ensure better accuracy. On the other hand, another important reason, which validates the use of the second order numerical integration, originates from the mathematical observation we already discussed in Section 6.2.2.4. Recall that all VIE formulations are categorized as the the second kind Fredholm integral equations, or more precisely in the form of $(\mathcal{L}+\mathcal{I})\{\boldsymbol{F}\} = \boldsymbol{R}$. Here, the identity operator $\mathcal{I}$ manifests itself as the first term of (6.49) known as the "self-interaction" term. Clearly, it is a second order volumetric integral with respect to the vector position $\boldsymbol{r}$, and and hence, we need at least a second order Gaussian quadrature to calculate it accurately. Its accurate evaluation determines the values of diagonal elements regarding the MoM matrix.

It is worth to mention that the penalty of using higher order quadratures for better accuracy is a much higher computation time. For example, switching from the first order quadrature to the second order one, quadruples the computation time due to the use of four points inside a tetrahedron instead of a single point at the centroid of a tetrahedron (and equivalently triples for triangular computations). Taking into account such a computational burden, and also considering more smooth variation of far fields and the corresponding interactions, we may use single-point quadrature for far-field computations. Numerical simulations also confirm the use of first order quadrature for far-field calculations.

6.5 FMM for Far-field Interactions

To derive a suitable form for far-field interactions and extract the receiving, sending, and translation functions to be used in FMM codes, one can omit the self-interaction term from (6.43), and write it as [35]

$$\begin{aligned}\bar{\boldsymbol{Z}}^{\text{EFVIE-D}}_{\text{far-field}}[m,n,a,b] = {} & iw\mu_0\kappa_{nb}\,\mathcal{C}_{ma,nb}\int_{D_{ma}} d\boldsymbol{r}\,(\boldsymbol{r}-\boldsymbol{r}_{ma})\cdot\int_{D_{nb}} d\boldsymbol{r'}\,(\boldsymbol{r'}-\boldsymbol{r}_{nb})g_0(\boldsymbol{r},\boldsymbol{r'}) \\ & + \frac{\kappa_{nb}\,\mathcal{C}_{ma,nb}}{iw\epsilon_0}\int_{D_{ma}} d\boldsymbol{r}\,(\boldsymbol{r}-\boldsymbol{r}_{ma})\cdot\nabla\int_{D_{nb}} d\boldsymbol{r'}(\boldsymbol{r'}-\boldsymbol{r}_{nb})\cdot\nabla' g_0(\boldsymbol{r},\boldsymbol{r'})\end{aligned} \tag{6.51}$$

By rescaling (4.2) as the diagonalized version of the Green's function (note that $g = \tilde{g}\frac{ik}{4\pi}$), one can simply obtain

$$g_0(\boldsymbol{r}, \boldsymbol{r'}) = \frac{e^{ik_0|\boldsymbol{r}-\boldsymbol{r'}|}}{4\pi|\boldsymbol{r}-\boldsymbol{r'}|} = \frac{e^{ik_0|\boldsymbol{w}+\boldsymbol{v}_{\text{agg}}+\boldsymbol{v}_{\text{dis}}|}}{4\pi|\boldsymbol{w}+\boldsymbol{v}_{\text{agg}}+\boldsymbol{v}_{\text{dis}}|}$$
$$= \frac{ik_0}{(4\pi)^2}\int d^2\hat{\boldsymbol{k}}\ \exp(ik_0\hat{\boldsymbol{k}}\cdot(\boldsymbol{v}_{\text{agg}}+\boldsymbol{v}_{\text{dis}}))\,\alpha(k_0,\hat{\boldsymbol{k}},\boldsymbol{w}), \tag{6.52}$$

where $\boldsymbol{w} = \boldsymbol{r}_C - \boldsymbol{r}_{C'}$ is a constant translation vector, and $\boldsymbol{v}_{\text{agg}} = \boldsymbol{r'} - \boldsymbol{r}_{C'}$ and $\boldsymbol{v}_{\text{dis}} = \boldsymbol{r} - \boldsymbol{r}_C$ are aggregation and disaggregation vectors with $\boldsymbol{r}_{C'}$ and $\boldsymbol{r}_C$ vectors referring to aggregation and disaggregation box centers, respectively. Now by considering two ∇ operators operating on the Green's function at the right hand side of (6.51), we can write [35]

$$\bar{\boldsymbol{Z}}^{\text{EFVIE-D}}_{\text{far-field}}[m,n,a,b] = -\frac{w\mu_0k_0}{(4\pi)^2}\int d^2\hat{\boldsymbol{k}}\,\boldsymbol{R}_{ma}(k_0,\hat{\boldsymbol{k}},\boldsymbol{r}_C)\cdot\alpha(k_0,\hat{\boldsymbol{k}},\boldsymbol{r}_C-\boldsymbol{r}_{C'})\boldsymbol{S}_{nb}(k_0,\hat{\boldsymbol{k}},\boldsymbol{r}_{C'}) \tag{6.53}$$

with

$$\alpha(k_0,\hat{\boldsymbol{k}},\boldsymbol{w}) = \sum_{t=0}^{\tau}(i)^t(2t+1)h_t^{(1)}(kr)P_t(\hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{w}}) \tag{6.54}$$

$$\boldsymbol{R}_{ma}(k_0,\hat{\boldsymbol{k}},\boldsymbol{r}_C) = \frac{\gamma_{ma}A_m}{3V_{ma}}(\bar{\bar{\boldsymbol{I}}}-\hat{\boldsymbol{k}}\hat{\boldsymbol{k}})\cdot\int_{D_{ma}}d\boldsymbol{r}\ \exp\left[ik_0\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}-\boldsymbol{r}_C)\right](\boldsymbol{r}-\boldsymbol{r}_{ma}) \tag{6.55}$$

$$\boldsymbol{S}_{nb}(k_0,\hat{\boldsymbol{k}},\boldsymbol{r}_{C'}) = \frac{\gamma_{nb}A_n\,\kappa_{nb}}{3V_{nb}}(\bar{\bar{\boldsymbol{I}}}-\hat{\boldsymbol{k}}\hat{\boldsymbol{k}})\cdot\int_{D_{nb}}d\boldsymbol{r'}\ \exp\left[-ik_0\hat{\boldsymbol{k}}\cdot(\boldsymbol{r'}-\boldsymbol{r}_{C'})\right](\boldsymbol{r'}-\boldsymbol{r}_{nb}). \tag{6.56}$$

From the numerical point of view, and using a first order quadrature, one may compute the integrals given in (6.55) and (6.56) by evaluating the integrands at the centroid ($\boldsymbol{r}_{\text{cent}}$) of the corresponding tetrahedron. Note that this approximation, i.e., using $\int_V d\boldsymbol{r}\boldsymbol{f}(\boldsymbol{r}) \cong V\boldsymbol{f}(\boldsymbol{r}_{\text{cent}})$ with $\boldsymbol{f}$ representing a general vector function becomes exact if $\boldsymbol{f}$ is a linear function of $\boldsymbol{r}$. Clearly, the integrands of the last two integrals regarding the receiving and sending patterns are not strictly linear. However, for far-field interactions, the exponential part shows small variations with respect to the integration variable allowing us to assume a quasi-linear behavior and use the first order quadrature.

6.6 Computation of Far-fields

After solving the problem, expansion coefficients a_n in $iw\boldsymbol{D}(\boldsymbol{r}) = \sum_{n=1}^{N} a_n \boldsymbol{b}_n(\boldsymbol{r})$ are known and can be used to compute all far-field values. Invoking the far-field radiation formulas given by (2.37) and utilizing previously defined equivalent volume current given by (6.15), we can simply write

$$\begin{aligned}\boldsymbol{E}^{\text{scat}}_{\text{far-field}} &= \frac{iw\mu_0}{4\pi}(\bar{\bar{\boldsymbol{I}}} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}}) \cdot \int_D d\boldsymbol{r}'\, \boldsymbol{J}(\boldsymbol{r}')\, \exp(-ik_0\hat{\boldsymbol{k}} \cdot \boldsymbol{r}') \\ &= -\frac{iw\mu_0}{4\pi}(\bar{\bar{\boldsymbol{I}}} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}}) \cdot \int_D d\boldsymbol{r}'\, iw\kappa(\boldsymbol{r}')\boldsymbol{D}(\boldsymbol{r}')\, \exp(-ik_0\hat{\boldsymbol{k}} \cdot \boldsymbol{r}'),\end{aligned} \tag{6.57}$$

where after discretization we end up with

$$\boldsymbol{E}^{\text{scat}}_{\text{far-field}} = -\frac{iw\mu_0}{4\pi}\,(\bar{\bar{\boldsymbol{I}}} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}}) \cdot \int_D d\boldsymbol{r}'\, \kappa(\boldsymbol{r}') \sum_{n=1}^{N} a_n \boldsymbol{b}_n(\boldsymbol{r}')\, \exp(-ik_0\hat{\boldsymbol{k}} \cdot \boldsymbol{r}').$$

6.7 Implementation of IL-VIE

We will deploy the D-formulation in order to implement the IL version of EFVIE. Also we use MATLAB programming environment to simulate and compare the results on a 64-bit Windows platform. This is chosen because it is fully compatible with divergence-conforming basis functions, i.e., SWG functions.

A full SWG function may be considered as three full RWG functions connected to each other from the side edges with two shared vertices at two ends, and hence, some similar methods with low complexity can be used to read the file containing node coordinates and quadruple node numbers per each tetrahedron and detect the needed edges and faces. The file containing such nodal information to model the discretized structure are produced by commercial 3D-meshing software like

Siemens NX, COMSOL, Ansys, Autocad and many more.

6.7.1 Tree Structure

Interestingly, the tree structures we used before in SIEs are very similar to what we need in VIEs. In both of them we start from a large box, encompassing the whole object, and bisect the edges in three directions to get smaller boxes, each of which containing a part of the original box. This operation starts from level 1 (the root box), and recursively continues down to a given level, commonly known as the leaf level. Obviously the empty boxes at each level are eliminated. Similar operations have to be performed to find and create a list of near and far boxes, which are going to be used later on in near-field and far-field computations.

Assuming fine enough meshes (at the leaf level) to discretize the surface/volume of the object, we can highlight an important difference between the two trees, i.e., the tree used in SIEs and the tree used in VIEs. Most of the near boxes in SIEs are located in the same plane of the box of interest (except at the sharp corners). This means that, each box has a near interaction with 8 nearby boxes (in average). But in VIEs, each box involves with all 26 surrounding boxes (except those boxes close to the boundaries). This simply means that for VIEs, the list of near and far boxes are much more dense than what we had for SIEs. Also the protrusion of the large SWG functions are worse than the protrusion of the RWG functions, and hence, we expect to see a higher amount of error regarding the multiscale meshes.

Having access to detailed population information and needed indices regarding the near and far boxes in each and every level, we may construct the IL tree structures quite similar to what we discussed in Chapter 3.

6.7.2 MVM Routines

As already mentioned in the introduction, the low-frequency breakdown is not a concern in VIEs, and MoM may be used in conjunction with SWG function at an arbitrary low frequency. The main reason is that VIEs are second kind Fredholm integral equations, and the resultant impedance matrix is well-conditioned [70]. Based on this important fact, there is no need to use two distinct MVM routines to separate the high- and low-frequency regimes. Recall that regarding this issue in SIEs, we used an approximate diagonalization method to properly expand the Green's function and deploy it in far-field interactions. By eliminating this need, we may achieve two goals: (a) We may have the same $\mathcal{O}(N \log N)$ complexity instead of $\mathcal{O}(N \log^2 N)$ for MVM routines. (b) Because we are not using an approximate method, we expect to see better accuracy for VIEs than what we have for SIEs. Besides, we may use the same interpolation and anterpolation routines among different levels.

6.8 Numerical Results

Generally speaking, a multiscale volume discretization may be used in one or a combination of these cases:

1. When the object is homogeneous but possesses fine details.
2. When the object is piece-wise homogeneous or when there are multiple homogeneous objects each of which with different electrical properties. Apparently, smaller mesh sizes should be considered for those materials with higher permittivity and/or permeability. A multilayer coated dielectric filter is a good example for this case.
3. When the object is inhomogeneous, as most of the real-life problems are in this category.

For the first two cases above, one can use canonical sphere problems (either single or multilayer spheres) along with the Mie-series solution as an analytic solution in order to compare the results and calculate the error. But for the last case, there is no other choice than accepting the MoM solution or even the FMM solution as a reference. For the upcoming numerical results, we use a similar definition to calculate the relative RMS error given by

$$\text{Relative RMS Error} = \sqrt{\frac{\sum_{i=1}^{N} \left|\boldsymbol{E}_{\theta_i}^{\text{Sim}} - \boldsymbol{E}_{\theta_i}^{\text{Ref}}\right|^2}{\sum_{i=1}^{N} \left|\boldsymbol{E}_{\theta_i}^{\text{Ref}}\right|^2}}, \tag{6.58}$$

where $\boldsymbol{E}_{\theta_i}^{\text{Sim}}$ and $\boldsymbol{E}_{\theta_i}^{\text{Ref}}$ are the dominant θ components of the far-zone electric field, computed from the simulation and the reference solutions (Mie, MoM, or at least FMM), respectively, to assess the accuracy of the proposed IL-MLFMA.

It should be reminded that in the conventional MLFMA, the proper choice of the number of levels has an important role in the efficiency and also the accuracy of the final solution. However, in IL-MLFMA, the number of levels is not a concern, since it is automatically determined by the program based on a given (maximum) box population (MBP). Consequently, there may be a couple of solutions based on different MBPs leading to the very same number of levels. This is particularly noteworthy about IL-MLFMA, which gives us an opportunity to optimize the overall efficiency and find the best value (or range of values) for MBP. Due to above reasoning, in the upcoming subsections, we will present the efficiency and accuracy results based on the number of levels, since it is the only common parameter between the conventional MLFMA and IL-MLFMA.

6.8.1 A Homogeneous Sphere (I)

Fig. 6.2 shows a homogeneous dielectric sphere of radius $R = 50$ mm and $\epsilon_r = 12$, discretized with a highly non-uniform volumetric mesh, consisting of 19,515 tetrahedrons (40,273 SWG functions) with a wide range of edge sizes varying from 0.1 mm at the north pole up to 16 mm at the south pole, i.e., a multiscale factor of 160. It is illuminated by a unit plane wave at 300 MHz propagating in

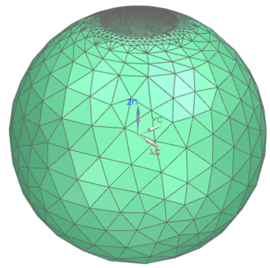


Figure 6.2: A homogeneous dielectric sphere of radius $R = 50$ mm, $\epsilon_r = 12$, discretized with a highly non-uniform volumetric mesh, consisting of 19,515 tetrahedrons (40,273 SWG functions) with a wide range of edge sizes varying from 0.1 mm at the north pole up to 16 mm at the south pole. The multiscale factor is about of 160.

the z direction with the electric field polarized in the x direction. Fig. 6.3 shows the total run-times and the relative RMS errors of IL-MLFMA for this geometry as a function of maximum box population, which is changed from 20 to 9000. Note that we have two vertical axis with red and blue numbers corresponding to red and blue data points. The red one at the right concerns the simulation time, and the blue one at the left concerns the RMS error.

Interestingly, we can see almost a similar behavior that we have seen in SIEs. This means that, even though the population threshold changes more than two orders of magnitude from 20 to 9000 (shown by MoM in the figure), the relative RMS error changes by less than 0.8%. On the other hand, the total simulation time is minimized for some population thresholds, around 30 up to 100. We solved the same scattering problem using a modified version of the conventional MLFMA (to be discussed later) to have a comparison among both versions. Fig. 6.4 illustrates the results. Note that, IL-MLFMA is based on the population threshold and the number of levels is determined automatically by the program. Hence, we may have the same level for multiple populations. We have chosen the best results in Fig. 6.4.

In Fig. 6.4, if we consider the best timing that belongs to level 5 of the conventional MLFMA, the RMS error is unacceptably high. Therefore, we consider the next best timing at level 4 with a relative RMS error of 1.8% and simulation time of 21,000 seconds. Regarding IL-MLFMA, we obtain 2.2% RMS error and a simulation time of less than 3,000 seconds.

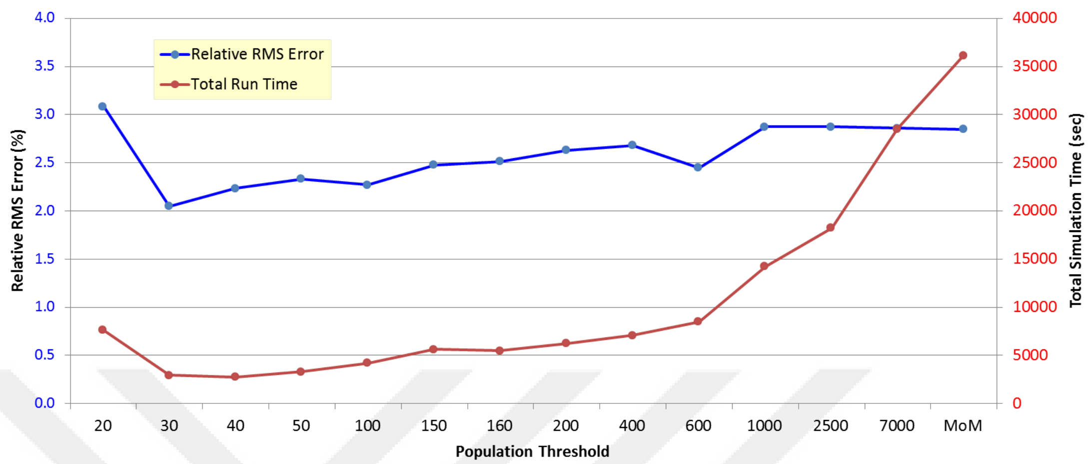


Figure 6.3: Total run-times and the relative RMS errors of IL-MLFMA for the geometry shown in Fig. 6.2 as a function of box population threshold.

6.8.2 A Homogeneous Sphere (II)

As a larger problem, Fig. 6.5 shows a homogeneous dielectric sphere of radius $R = 60$ mm and a relative permittivity of $\epsilon_r = 4$, illuminated by a similar plane wave at 600 MHz ($R \approx \lambda/4$). The sphere is discretized with a non-uniform volume mesh consisting of 49,452 tetrahedrons (105,081 SWG functions), where the edge sizes vary from 0.1 mm ($\lambda/2500$) at the north pole up to 25 mm ($\lambda/10$) at the south pole giving rise to a multiscale factor of 250. By assigning four different values for MBP from 50 to 1500 (regarding IL-MLFMA) and four consecutive levels from 3 to 6 (regarding conventional MLFMA), Fig. 6.6 illustrates the total run-time (right vertical axis and the solid (red) lines) and the relative RMS error (left vertical axis and the doted (blue) lines) versus the common number of levels for both versions of MLFMA. Note that both axes are in logarithmic scale and the relative RMS error is calculated using the Mie-series solution for a scattering problem of a dielectric sphere, as a reference.

Considering the accuracy, even though the MoM solution is not included due to a large amount of needed memory (> 180 GB), both versions are comparable to each other upto level 4 (with error $< 2\%$). However, due to a high multiscale factor, the results obtained from the conventional MLFMA either start to become inaccurate after level 4 or it fails to converge beyond level 5. Interestingly, this

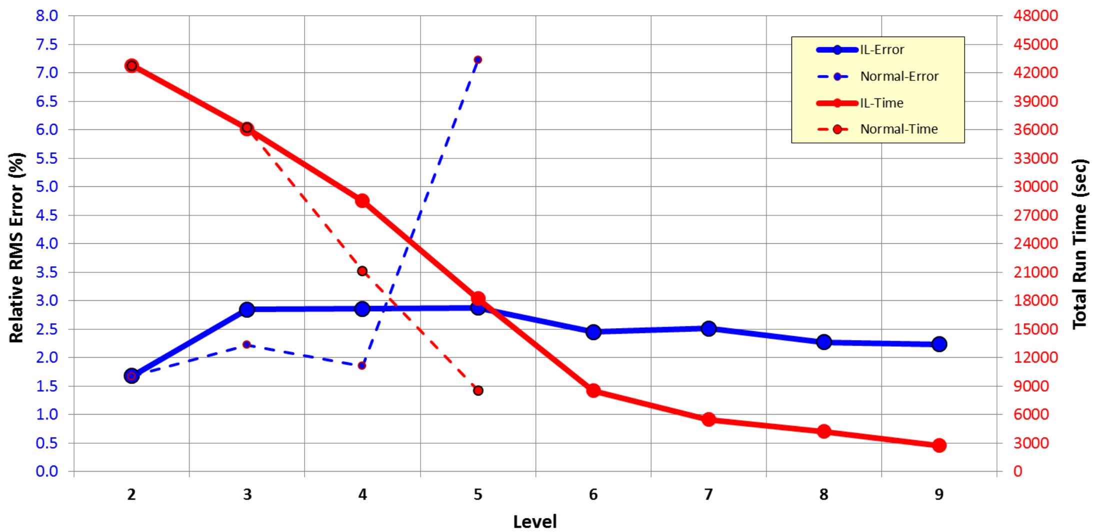


Figure 6.4: Total run-times and relative RMS errors versus the number of levels for both versions of MLFMA. The geometry in Fig. 6.2 is used at $f = 300$ MHz. Blue lines and red lines along with corresponding vertical axis at left and right show the total run-times and the relative RMS errors, respectively.

is not true for the IL version, since it always converges for a wide range of MBP spanning three orders of magnitude, i.e., from 25 up to 25000 (equivalent with level 10 down to the MoM level), although the values higher than 300 are not very efficient.

Regarding the timing results, the IL version demonstrates much better efficiency than the conventional version for some optimal values of MBP (i.e., 120 and 50) which are equivalent with the 8th or 9th level. Note that the accuracy

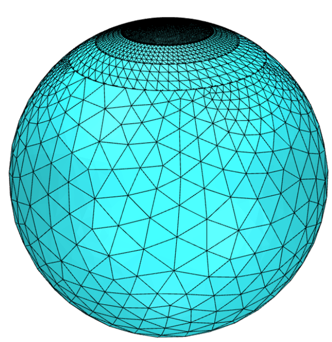


Figure 6.5: A homogeneous dielectric of radius $R = 60$ mm and relative permittivity of $\epsilon_r = 4$ possessing a dense discretization around the north pole with a multiscale factor of 250. The mesh size varies from 0.1 mm up to 25 mm, consisting of 49,452 tetrahedrons and 105,081 SWG functions.

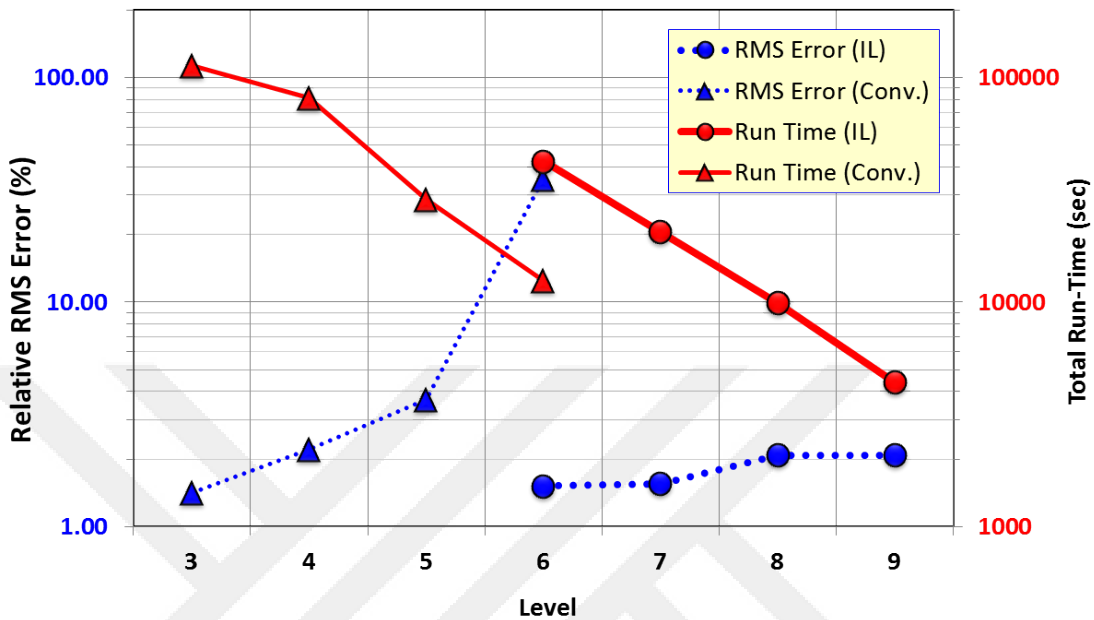


Figure 6.6: Total run-times and relative RMS errors versus the common number of levels for both versions of MLFMA, where the geometry shown in Fig. 6.5 is used at $f = 600$ MHz. Circle and triangle markers are used to separate the IL-MLFMA and the conventional MLFMA, respectively. Black (blue) doted lines and gray (red) solid lines along with the corresponding logarithmic vertical axis at left and right, show relative RMS errors and total run-times, respectively.

for the conventional MLFMA increases dramatically from 1.5% upto 35%. However, regarding IL-MLFMA, it remains almost stable with only 0.5% increase from level 6 (MBP=1500) upto level 9 (MBP=50), mainly due to the fact that for lower MBP one expects to see more protrusions. Consequently, by accepting the error margin as less than 2%, the best timing for the conventional MLFMA belongs to 4th level with 81000 seconds and for IL-MLFMA, the 9th level with 4400 seconds, which is 18 times better.

This issue regarding the conventional MLFMA (i.e., using fixed-box sizes) can be better seen for larger problems involving with either larger numbers of unknowns or larger sizes. When we are dealing with larger number of unknowns, more levels are required to efficiently solve the problem and hence, the box sizes shrink significantly, leaving behind a bunch of leaf-level boxes with severe protrusions. On the other hand, when dealing with geometrically larger objects, even at some deep levels, the leaf-level box sizes are overcrowded enough to encounter

$\mathcal{O}(N^2)$ complexity regarding near-filed computations.

6.8.3 A Double Layer Sphere

To demonstrate the performance of IL-MLFMA for a piecewise homogeneous dielectric object, we choose a double layer sphere as depicted in Fig. 6.7 with inner and outer radii of 20 mm ($\epsilon_r = 16$) and 200 mm ($\epsilon_r = 2$), respectively. The volume discretization is such that the mesh size varies between 1 mm and 75 mm, leading to a multiscale factor of 75 with 33,450 tetrahedrons (71,594 SWG functions). Due to different permittivities, the finer meshes mostly belong to the inner core of the sphere while the coarse meshes are embedded inside the cladding.

By exciting such a double layer sphere at $f = 800$ MHz, we solve a canonical scattering problem first by using the Mie-series solution, then using the conventional MLFMA for levels 4, 5, and 6 (no other solutions for higher levels due to convergence problems), and finally with IL-MLFMA for MBP=100. The far-zone electric field as a function of bistatic angle along with the Mie-series solution are shown in Fig. 6.7. It is seen that there is a very good agreement between Mie-series result and the values obtained from both versions except the 6th level of the conventional MLFMA. The small amount of discrepancy for all other solutions are originated from the fact that some of the large meshes inside the cladding (between 30 mm and 75 mm) are larger than the minimum acceptable mesh size at $f = 800$ MHz for $\epsilon_r = 2$ ($\frac{\lambda_0}{10\sqrt{\epsilon_r}} = 26.5$ mm). Note that such a discretization error is more dominant in lower levels than the error incurred by the protrusions and thus, we expect to see comparable accuracies for both solutions since the IL-tree structure cannot compensate for discretization errors. Table 6.1 includes both run-times and relative RMS errors for the aforementioned sphere problem. As expected, the accuracy for the 4th and 5th level of the conventional MLFMA is comparable with IL-MLFMA, but the 6th level delivers a higher amount of error. Note that there is no other solution for the conventional MLFMA after level 6 due to convergence problems. Considering the total run-times, IL-MLFMA presents

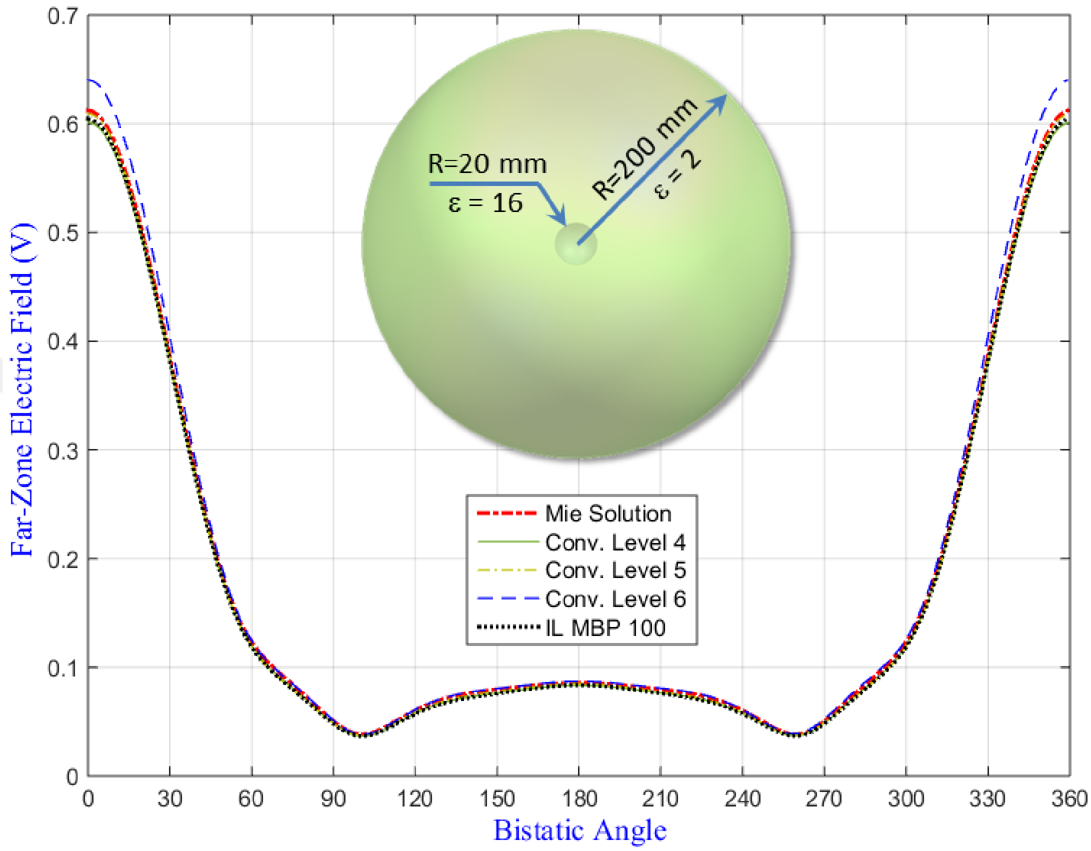


Figure 6.7: Far-zone electric field scattered from a double-layer dielectric sphere with the inner radius of $R = 20$ mm ($\epsilon_r = 16$) and the outer radius of $R = 200$ mm ($\epsilon_r = 2$). The sphere is illuminated by a unit plane wave at $f = 800$ Mhz. The results from the Mie-series solution and from both MLFMA versions are presented for comparison.

3.5 to 7.5 times better efficiency than the conventional MLFMA for an optimum value of MBP.

6.8.4 A More Complicated Object

A more complicated dielectric resonance structure consisting of three layers are shown in Fig. 6.8. The bottom layer is a 40×40 mm^2 dielectric slab of thickness 5 mm and $\epsilon_r = 2$, and it is loaded by two groups of perpendicular arrays at the top of each other, where each group contains 15 dielectric rods of $\epsilon_r = 8$ in parallel. The sizes along with the left- and right-side views are shown in the picture. The bottom layer is discretized with an average mesh size of 5 mm, while

Table 6.1: Performance Comparison Between The IL-MLFMA and Conventional MLFMA Applied to the Double Layer Sphere Shown in Fig. 6.7 at f = 800 MHz with a Multiscale Factor of 75.

	Total Run-Time	Rel. RMS Error
Conv. (Level 4)	95,131 s	3.26 %
Conv. (Level 5)	46,316 s	3.55 %
Conv. (Level 6)	31,900 s	4.88 %
IL (MBP 100)	12,690 s	3.66 %

the dielectric bars are discretized with an average mesh size of 0.5 mm, leading to 37,253 tetrahedrons and a multiscale factor of 22.

The whole structure is illuminated by the same plane wave as used for previous examples at $f = 2$ GHz. The far-zone electric fields are calculated for both MLFMA versions for different levels and also MBPs. Due to the fact that there is no analytic solution for such a scattering problem, it is reasonable to use the MoM solution as a reference to assess the relative RMS error. Fig. 6.9 illustrates the total run-time (right logarithmic vertical axis and the solid (red) lines) and the relative RMS error (left logarithmic vertical axis and the doted (blue) lines) versus the common number of levels for both versions of MLFMA. Based on the given results, one can arrive at the following results regarding the efficiency and more importantly, regarding the accuracy.

By considering the efficiency, the best timing for the conventional version belongs to level 5 with about 10500 seconds, where the error is 0.16%, however, for the equivalent 6th level of the IL version and almost the same accuracy, the total timing shows 4380 seconds, which is 2.4 times more efficient. Note that the 8th equivalent level of IL-MLFMA delivers even a better efficiency (about 3000 seconds) if under 0.8% error is acceptable. Regarding the accuracy, it is seen that the conventional version acts perfectly for levels up to 5, however, the error jumps 1.5 orders of magnitude starting from the level 6. Considering the IL version, the error smoothly increases due to lower MBPs (and actually higher protrusions).

It should be remarked that there is an important difference between this example and the others regarding the discretization, which is not clear at the first

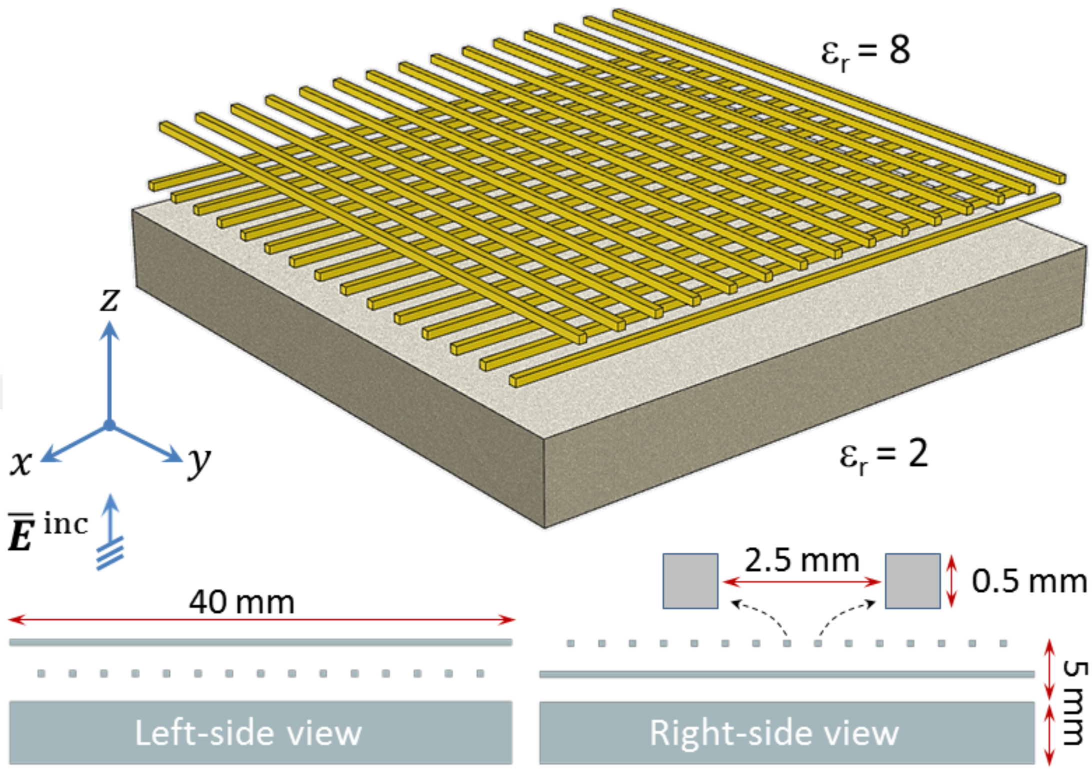


Figure 6.8: A dielectric resonance structure consisting of three layers. The bottom layer is a slab of $40 \times 40 \times 5$ mm^3 with $\epsilon_r = 2$, while two other top layers contain two sets of 15 parallel dielectric rods perpendicular to each other. All dimensions and side views are shown.

glance. In this example, the leaf-level box boundaries at levels 3 up to 5 (equivalent to box sizes from 10 mm down to 2.5 mm) are located in very special positions. This is mainly because the object has a simple geometrical structure consisting of flat and mostly parallel boundaries which are also parallel with the box boundaries. On the other hand, the discretized object (not shown here) mostly includes right-angled tetrahedrons within the parallel rods and some other very ordered larger meshes inside the bottom layer. Both of the above mentioned geometrical features produce a situation that a very few number of tetrahedrons are getting cut by the box boundaries, leading to minimum possible protrusions regarding the SWG functions. This is why the conventional MLFMA performs very good up to level 5. But starting from level 6 (i.e., a box size of 1.25 mm) almost all of the SWG functions are severely getting cut by the leaf-level box boundaries and hence, one expects to experience heavy protrusions by which, the conventional MLFMA is unable to compensate for it. Note that for the other examples, there

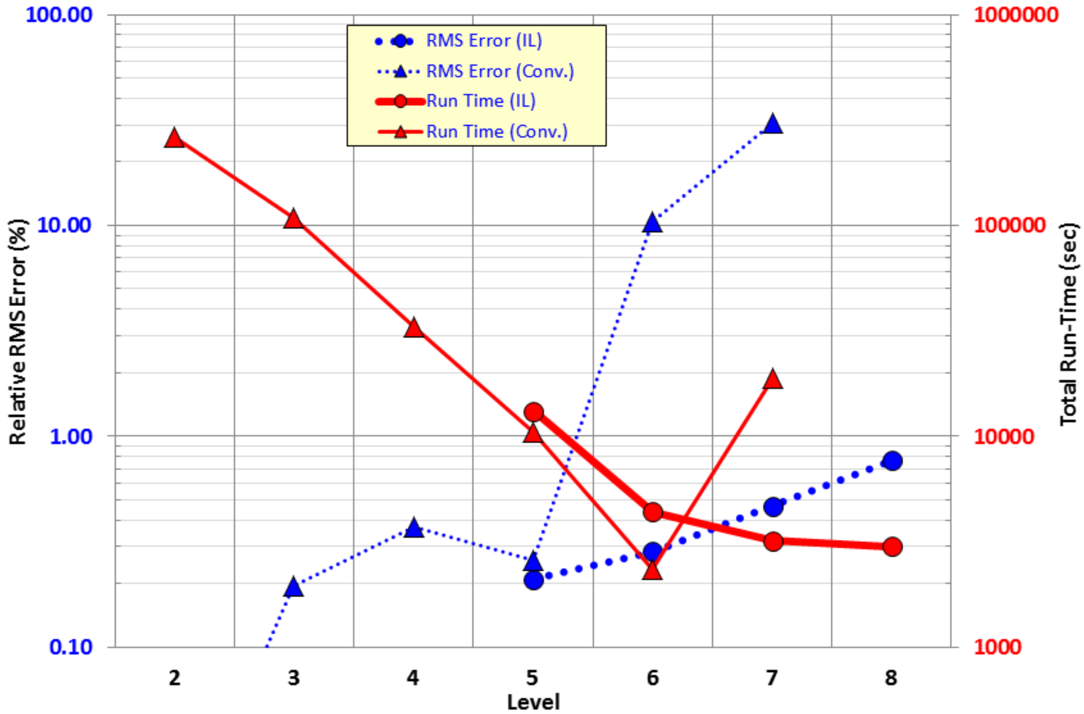


Figure 6.9: Total run-times and relative RMS errors versus the common number of levels for both versions of MLFMA, where the geometry shown in Fig. 6.8 is used at $f = 2$ GHz. Circle and triangle markers are used to separate the IL-MLFMA and the conventional MLFMA, respectively. Black (blue) doted lines and gray (red) solid lines along with the corresponding logarithmic vertical axis at left and right, show relative RMS errors and total run-times, respectively.

is no such a special condition and the large number of the SWG functions (or the tetrahedrons) are cut from almost the first levels so the error increases gradually (and not abruptly as we see in this example).

6.9 Memory Performance of IL-MLFMA

Up to this point, we investigated the superiority of IL-MLFMA from both efficiency and accuracy point of view for some medium and large scattering problems. Considering the fact that for IL-MLFMA, there is no theoretical limit for the number of levels to be used within the solution procedure, we can conclude that the memory consumption for IL-MLFMA will be much less than the conventional version. This is simply because IL-MLFMA is able to use the higher

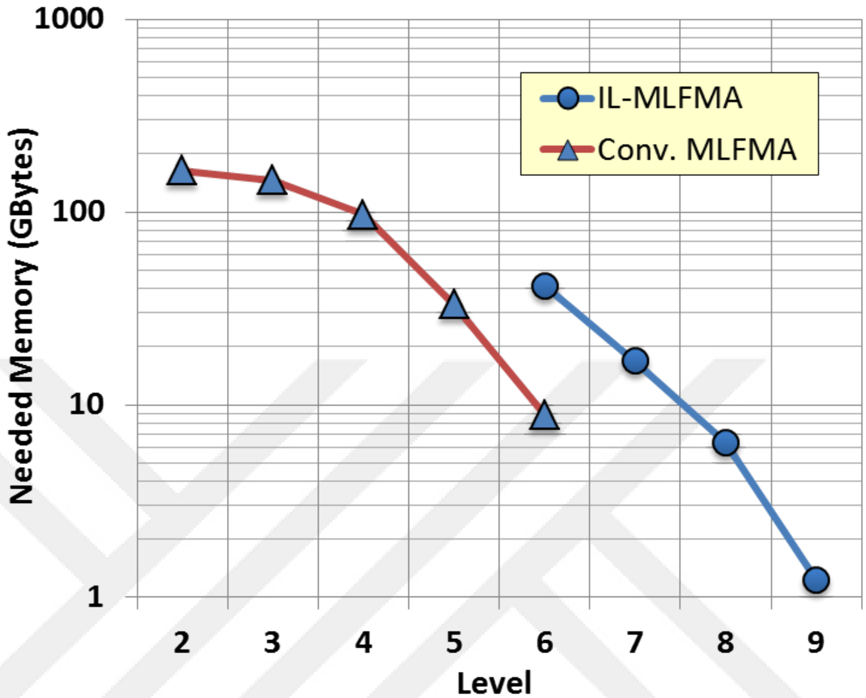


Figure 6.10: The amount of memory needed to store the near-field interactions and also compute the far-field interactions for a canonical scattering problem shown in Fig. 6.5 with the results given in Fig. 6.6.

(equivalent) levels very safely and without compromising the accuracy, which is not possible for the conventional MLFMA.

To compare the performance of the IL versus the conventional version regarding the memory consumption, it is enough to reconsider the second numerical example (a homogeneous sphere) shown in Fig. 6.5 since all other examples have some similar behaviors. In this case we just compare the required memory to hold the near-field and also far-field computations.

Recall from Fig. 6.6 that for an accuracy of better than 2%, only the 3rd and 4th levels of the conventional MLFMA may be used. These levels correspond to 150 GB and 90 GB of memory, respectively. Even for the 5th level (RMS error of 3.5%), 32 GB of memory is needed. However, by invoking IL-MLFMA, all four equivalent levels may be used with wide range of memory needs from 40 GB downto 1.2 GB, where the last one is an obvious choice from all three points of view, i.e., efficiency, accuracy, and also hardware resources.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

An efficient, accurate, and versatile broadband MLFMA, referred to as IL-MLFMA, is presented to solve multiscale electromagnetic problems for either PEC objects under SIEs or inhomogeneous dielectric objects under VIEs. The concept of IL-tree structures is introduced, where only the overcrowded boxes are divided into smaller ones for a given population threshold, leading to a nonuniform clustering. Thus, protrusions of RWG or SWG functions from the boxes are minimized that improves the accuracy, and the total number of interacting boxes is reduced that improves the efficiency. For SIEs, the error of the proposed method is almost independent of the required levels, while for VIEs, the error increases very slowly as we use deeper levels. Consequently, for the geometries that can be discretized with nonuniform meshes with a large multiscale factor, IL-MLFMA is always more efficient than the conventional one for the same accuracy, and it is more accurate if the efficiency is comparable, as demonstrated with some canonical and also real-life examples. Furthermore, IL-MLFMA recovers the conventional MLFMA for uniform meshes, if desired. As a result, all other computational methods that can be combined with the conventional MLFMA can also be used in conjunction with the proposed IL-MLFMA.

7.2 Future Works

This dissertation opens various potential research avenues from different points of view. Therefore, we can propose possible future works in two categories.

7.2.1 From Programming and Computational Point of View

In order to successfully implement the proposed IL-MLFMA for both SIEs and VIEs, it is necessary to write proper codes and carefully combine lots of routines and algorithms in a consistent manner. These codes begin from a very simple task of calculating the area of a triangle or volume of a tetrahedron and ascend to more complicated routines like nonuniform clustering of an arbitrary 3D object. In each of these routines, it is almost always possible to find and write more efficient codes. Significant improvements may be obtained by implementing the following suggestions.

1. Even though MATLAB is known as one of the best programming environments for engineering applications and rapid numerical prototyping, its performance for large scale problems is not even comparable with other engineering languages like Fortran or Python. For example, Fortran's low level input/output (IO) commands are at least 10 times faster than MATLAB. As another important limitation, MATLAB has serious problems with MEX codes regarding memory allocation, which is not the case for Python or Fortran. Therefore, if using MATLAB is preferred due to its wide range of flexible commands, it is highly recommended to use it in conjunction with Fortran, C, Python, Java or any other proper language.

2. Using "Profile" feature of MATLAB helps us to find the bottlenecks of the written codes from timing point of view. Using this feature, one can exactly find out that how many times a command is executed, a subroutine is called,

a variable or an array or a matrix is updated and many more. This helps us to optimize the codes for some time-consuming commands, vectorize the loops and some array manipulations, or use some ready libraries written in other languages (e.g., Fortran, C, Java) to do the job. Note that in this dissertation, we have already optimized and vectorized the codes as much as possible.

3. Sparse-oriented and cell-based MATLAB commands, though they are very useful, are comparatively slower than other normal commands. Also using these commands inside the MEX codes are not implemented yet. Instead, using linked-lists in C or Fortran are highly recommended to overcome such a shortage in MATLAB.

4. Commonly used MATLAB commands are designed to be as flexible as possible. This flexibility sacrifices the execution time, which manifests itself in large problems where a command or a group of commands are executed hundreds of million times. Consequently, as already mentioned in Section 5.5.4.3, it is better to optimize the codes as much as possible, or even rewrite some application specific scripts to minimize the execution times.

5. Although many of the time-consuming loops are converted into fast MEX codes, there are many others than can be partially replaced by proper MEX codes to increase the performance. For example, the loops required to process the meshes and cluster the boxes within the IL-tree structure are among these codes. For medium-sized EM problems, where the object size is no more than a few wavelengths, the construction time of the tree structure takes at most 10% of the total run-time. However, when dealing with slightly larger problems, 10 or even 12 percent of the total run-times leads to hours of extra simulation time.

6. The most time-consuming and CPU demanding piece of our codes belong to singularity extraction routines. Though it seems not to be trivial, singularity extraction routines can be merged due to the fact that the methodology used in our codes are triangle-based rather than RWG-based or SWG-based computations. This means that, it is possible to compute and save many

of the variables inside these routines and wisely use them multiple times to save time.

7. Similar to our newly developed VIE routines, which exhibit no memory limit regarding large or very large problems, it is also possible to modify the SIE routines to get rid of the memory limitations imposed by fast MEX routines. Note that MATLAB has an inherit 256 GB memory limit under 64-bit Windows platform.

7.2.2 From Algorithmic and Structural Point of View

1. SIEs suffer from an approximate method regarding the diagonalization of the Green's function. This is remediable by using other accurate diagonalization methods, though they might be more complicated than what is used in our codes (i.e., the scaled MLFMA).

2. Construction of near and far lists can be optimized within overcrowded boxes by temporarily saving the list of child boxes and reusing them wisely. This is extremely useful for larger problems.

3. The list of far boxes for a given basis box can be properly reused for other boxes which are near to the given box to save time. This is very useful for large problems.

4. Higher order basis functions maybe used with IL-MLFMA to reduce the number of unknowns.

5. Since volume discretization shows much sever protrusion inside the buffer boxes, it may be a good idea to use two-buffer-box scheme rather than the standard "one-buffer-box" scenario to reduce the computational error arising from the closeness of SWG functions.

6. It is worth to investigate a different scenario where most of those SWG functions that are cut by the faces of the boxes, can be calculated directly

by the MoM. This ensures us about the accuracy of the addition theorem used by the far boxes.

7. The program assumes a background medium at the opposite side of a half-SWG function, where there is a common face (as a triangle) with the surface of the object. Now, when two half-SWG functions with different electrical properties and more importantly, with comparatively different sizes, are in a touching situation, this assumption is not valid anymore and hence, a small amount of error occurs. This happens when the discretizing program is unable to properly discretize the object and create a smooth transition among the mesh sizes. Therefore, a pre-processing routine is required to check the proximity of half-SWG functions and modify the needed calculations.

Bibliography

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

[2] R. F. Harrington, *Field Computation by Moment Methods.* New York: MacMillan, 1968.

[3] X.-m. Pan, J. Wei, Z. Peng, and X. Sheng, "A fast algorithm for multiscale electromagnetic problems using interpolative decomposition and multilevel fast multipole algorithm," *Radio Sci.*, vol. 47, pp. 1–11, Feb. 2012.

[4] F. X. Canning, "Solution of impedance matrix localization form of moment method problems in five iterations," *Radio Sci.*, vol. 30, pp. 1371–1384, Sep. 1995.

[5] A. Boag and R. Mittra, "Complex multipole beam approach to electromagnetic scattering problems," *IEEE Trans. Antennas Propag.*, vol. 42, pp. 366–372, Mar. 1994.

[6] L. Tarricone and F. Malucelli, "Efficient linear system solution in moment methods using wavelet expansions," *IEEE Trans. Antennas Propag.*, vol. 48, pp. 1257–1259, Aug. 2000.

[7] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas Propag. Mag.*, vol. 35, no. 3, pp. 7–12, 1993.

[8] J. Song, C.-C. Lu, and W. C. Chew, “Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects,” *IEEE Trans. Antennas Propag.*, vol. 45, pp. 1488–1493, Oct. 1997.

[9] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, “AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems,” *Radio Sci.*, vol. 31, no. 5, pp. 1225–1251, 1996.

[10] J. R. Phillips and J. K. White, “A precorrected-FFT method for electrostatic analysis of complicated 3-D structures,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, pp. 1059–1072, Oct. 1997.

[11] E. Michielssen and A. Boag, “Multilevel evaluation of electromagnetic fields for the rapid solution of scattering problems,” *Microw. Opt. Technol. Lett.*, vol. 7, pp. 790–795, Dec. 1994.

[12] S. M. Seo and J. F. Lee, “A single-level low rank IE-QR algorithm for PEC scattering problems using EFIE formulation,” *IEEE Trans. Antennas Propag.*, vol. 52, no. 8, pp. 2141–2146, 2004.

[13] S. Kurz, O. Rain, and S. Rjasanow, “The adaptive cross-approximation technique for the 3-D boundary-element method,” *IEEE Trans. Magn.*, vol. 38, no. 2, pp. 421–424, 2002.

[14] L. Greengard and V. Rokhlin, “A fast algorithm for particle simulations,” *J. Comput. Phys.*, vol. 73, no. 2, pp. 325–348, 1987.

[15] V. Rokhlin, “Rapid solution of integral equations of classical potential theory,” *J. Comput. Phys.*, vol. 60, no. 2, pp. 187–207, 1985.

[16] V. Rokhlin, “Rapid solution of integral equations of scattering theory in two dimensions,” *J. Comput. Phys.*, vol. 86, no. 2, pp. 414–439, 1990.

[17] V. Rokhlin, “Diagonal forms of translation operators for the Helmholtz equation in three dimensions,” *Appl. Comput. Harmon. Anal.*, vol. 1, no. 1, pp. 82–93, 1993.

[18] S. Velamparambil, W. C. Chew, and J. Song, “10 million unknowns: is it that big?,” *IEEE Antennas Propag. Mag.*, vol. 45, no. 2, pp. 43–58, 2003.

[19] L. Gürel and Ö. Ergül, "Fast and accurate solutions of extremely large integral-equation problems discretised with tens of millions of unknowns," *Electronics Letters*, vol. 43, no. 9, pp. 499–500, 2007.

[20] E. Darve, "The fast multipole method (I): Error analysis and asymptotic complexity," *SIAM J. Numer. Anal.*, vol. 38, no. 1, pp. 98–128, 2000.

[21] M. Nilsson, "Stability of the high frequency fast multipole method for Helmholtz' equation in three dimensions," *BIT Numer. Math.*, vol. 44, no. 4, pp. 773–791, 2004.

[22] J.-S. Zhao and W. C. Chew, "Three-dimensional multilevel fast multipole algorithm from static to electrodynamic," *Microw. Opt. Technol. Lett.*, vol. 26, no. 1, pp. 43–48, 2000.

[23] M. Vikram, H. Huang, B. Shanker, and T. Van, "A novel wideband FMM for fast integral equation solution of multiscale problems in electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 57, pp. 2094–2104, Jul. 2009.

[24] L. Jiang and W. C. Chew, "Low-frequency fast inhomogeneous plane-wave algorithm (LF-FIPWA)," *Microw. Opt. Technol. Lett.*, vol. 40, no. 2, pp. 117–122, 2004.

[25] L. J. Jiang and W. C. Chew, "A mixed-form fast multipole algorithm," *IEEE Trans. Antenna Propag.*, vol. 53, no. 12, pp. 4145–4156, 2005.

[26] W. Chew, S. Koc, J. Song, C. Lu, and E. Michielssen, "A succinct way to diagonalize the translation matrix in three dimensions," in *Proc. AP-S Int. Symp.*, vol. 3, pp. 2072–2075, IEEE, 1997.

[27] L. Greengard, J. Huang, V. Rokhlin, and S. Wandzura, "Accelerating fast multipole methods for the Helmholtz equation at low frequencies," *IEEE Computat. Sci. Eng.*, vol. 5, no. 3, pp. 32–38, 1998.

[28] T. F. Eibert, "A diagonalized multilevel fast multipole method with spherical harmonics expansion of the k-space integrals," *IEEE Trans. Antennas Propag.*, vol. 53, no. 2, pp. 814–817, 2005.

[29] H. Cheng, W. Y. Crutchfield, Z. Gimbutas, L. F. Greengard, J. F. Ethridge, J. Huang, V. Rokhlin, N. Yarvin, and J. Zhao, “A wideband fast multipole method for the Helmholtz equation in three dimensions,” *J. Comput. Phys.*, vol. 216, no. 1, pp. 300–325, 2006.

[30] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert, “Randomized algorithms for the low-rank approximation of matrices,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 104, no. 51, pp. 20167–20172, 2007.

[31] J.-G. Wei, Z. Peng, and J.-F. Lee, “Multiscale electromagnetic computations using a hierarchical multilevel fast multipole algorithm,” *Radio Sci.*, vol. 49, no. 11, pp. 1022–1040, 2014.

[32] I. Bogaert, J. Peeters, and F. Olyslager, “A nondirective plane wave MLFMA stable at low frequencies,” *IEEE Trans. Antennas Propag.*, vol. 56, pp. 3752–3767, Dec. 2008.

[33] I. Bogaert and F. Olyslager, “A low frequency stable plane wave addition theorem,” *J. Comput. Phys.*, vol. 228, pp. 1000–1016, Mar. 2009.

[34] Ö. Ergül and B. Karaosmanoğlu, “Approximate stable diagonalization of the Green’s function for low frequencies,” *IEEE Antennas Wirel. Propag. Lett.*, vol. 13, no. 2, pp. 1054–1056, 2014.

[35] Ö. Ergül and L. Gürel, *The Multilevel Fast Multipole Algorithm MLFMA for Solving Large-Scale Computational Electromagnetics Problems.* Wiley-IEEE Press, 2014.

[36] I. Lashuk, A. Chandramowlishwaran, H. Langston, T.-A. Nguyen, R. Sampath, A. Shringarpure, R. Vuduc, L. Ying, D. Zorin, and G. Biros, “A massively parallel adaptive fast multipole method on heterogeneous architectures,” *Commun. ACM*, vol. 55, no. 5, pp. 101–109, 2012.

[37] T. J. Giese and D. M. York, “Extension of adaptive tree code and fast multipole methods to high angular momentum particle charge densities,” *J. Comput. Chem.*, vol. 29, no. 12, pp. 1895–1904, 2008.

[38] K. Sanbonmatsu and C.-S. Tung, "High performance computing in biology: multimillion atom simulations of nanoscale systems," *J. Struct. Biol.*, vol. 157, no. 3, pp. 470–480, 2007.

[39] L. Ying, G. Biros, D. Zorin, and H. Langston, "A new parallel kernel-independent fast multipole method," in *Proc. ACM/IEEE Conf. on Supercomp.*, pp. 14–14, IEEE Computer Society, Nov. 2003.

[40] W.-B. Kong, H.-X. Zhou, W.-D. Li, G. Hua, and W. Hong, "The MLFMA equipped with a hybrid tree structure for the multiscale EM scattering," *Int. J. Antennas Propag.*, vol. 2014, no. 281303, pp. 1–6, 2014.

[41] J. Song and W. C. Chew, "Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering," *Microw. Opt. Technol. Lett.*, vol. 10, no. 1, pp. 14–19, 1995.

[42] S. M. Rao and D. R. Wilton, "E-field, H-field, and combined field solution for arbitrarily shaped three-dimensional dielectric bodies," *Electromagnetics*, vol. 10, pp. 407–421, oct 1990.

[43] P. Ylä-Oijala, M. Taskinen, and S. Järvenpää, "Analysis of surface integral equations in electromagnetic scattering and radiation problems," *Engineering Analysis with Boundary Elements*, vol. 32, pp. 196–209, mar 2008.

[44] O. Ergül and L. Gürel, "Investigation of the inaccuracy of the MFIE discretized with the RWG basis functions," in *Proc. AP-S Int. Symp.*, vol. 3, pp. 3393–3396, IEEE, June 2004.

[45] W. Cai, T. Yu, H. Wang, and Y. Yu, "High-order mixed RWG basis functions for electromagnetic applications," *IEEE Trans. Microw. Theory Tech.*, vol. 49, pp. 1295–1303, Jul. 2001.

[46] S. Yan, J.-m. Jin, Z. Nie, Su Yan, Jian-Ming Jin, and Zaiping Nie, "On the testing of the identity operator and the accuracy improvement of the second-kind SIEs," *Proc. IEEE Antennas Propagat. Soc. Int. Symp. (APSURSI)*, pp. 3185–3188, Jul. 2011.

[47] Wikipedia, "Condition number." http://en.wikipedia.org/wiki/Condition number, 2015.

[48] S. Koc, J. Song, 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, Jan. 1999.

[49] D. Wilton, S. Rao, A. Glisson, D. Schaubert, O. Al-Bundak, and C. Butler, "Potential integrals for uniform and linear source distributions on polygonal and polyhedral domains," *IEEE Trans. Antennas Propag*, vol. 32, pp. 276–281, Mar. 1984.

[50] R. Graglia, "On the numerical integration of the linear shape functions times the 3-D Green's function or its gradient on a plane triangle," *IEEE Trans. Antennas Propag*, vol. 41, no. 10, pp. 1448–1455, 1993.

[51] P. Ylä-Oijala and M. Taskinen, "Calculation of CFIE impedance matrix elements with RWG and $\hat{n}\times$RWG functions," *IEEE Trans. Antennas Propag.*, vol. 51, pp. 1837–1846, Aug. 2003.

[52] D. A. Dunavant, "High degree efficient symmetrical Gaussian quadrature rules for the triangle," *Int. J. Numer. Meth. Eng.*, vol. 21, pp. 1129–1148, Jun. 1985.

[53] Y. Zhang, T. J. Cui, W. C. Chew, and J.-S. Zhao, "Magnetic field integral equation at very low frequencies," *IEEE Trans. Antennas Propag.*, vol. 51, pp. 1864–1871, Aug. 2003.

[54] S. Rao, D. Wilton, and A. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. 30, pp. 409–418, May 1982.

[55] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition.* Philadelphia, PA: SIAM, 1994.

[56] H. Shao, J. Hu, Z. Nie, and L. Jiang, "Simulation of multiscale structures using equivalence principle algorithm with grid-robust higher order vector basis," *J. Electromagn. Waves Appl.*, vol. 28, no. 11, pp. 1333–1346, 2014.

[57] F. Vipiana, P. Pirinoli, and G. Vecchi, "A multiresolution method of moments for triangular meshes," *IEEE Trans. Antennas Propag.*, vol. 53, no. 7, pp. 2247–2258, 2005.

[58] C. A. Balanis, *Advanced Engineering Electromagnetics*. New York: Wiley, 1989.

[59] M. Taskinen, "On the fully analytical integration of singular double integrals arising from the integral equation methods," in *in Computational Electromagnetics International Workshop (CEM)*, vol. 4, pp. 13–18, IEEE, Aug. 2011.

[60] N. Morita, N. Kumagai, and J. R. Mautz, *Integral Equation Methods for Electromagnetics*. Boston: Artech House, 1990.

[61] R. Graglia, D. Wilton, and A. Peterson, "Higher order interpolatory vector bases for computational electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 45, pp. 329–342, Mar. 1997.

[62] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, eds., *NIST Handbook of Mathematical Functions*. New York, NY: Cambridge University Press, 2010. Print companion to [64].

[63] J. A. Stratton, *Electromagnetic Theory*. John Wiley & Sons, 2007.

[64] "NIST Digital Library of Mathematical Functions." http://dlmf.nist.gov/, Release 1.0.10 of 2015-08-07. Online companion to [62].

[65] M. Bohme and D. Potts, "A fast algorithm for filtering and wavelet decomposition on the sphere," *Electron. Trans. Numer. Anal.*, 2003.

[66] O. Ergül, "Fast multipole method for the solution of electromagnetic scattering problems," Master's thesis, Bilkent University, Turkey, June 2003.

[67] M.-F. Wu, G. Kaur, and A. E. Yilmaz, “A multiple-grid adaptive integral method for multi-region problems,” *IEEE Trans. Antennas Propag*, vol. 58, no. 5, pp. 1601–1613, 2010.

[68] K. Yang, *FFT and multigrid accelerated integral equation solvers for multi-scale electromagnetic analysis in complex backgrounds.* PhD thesis, University of Texas at Austin, Aug. 2014.

[69] Ö. Ergül and B. Karaosmanoğlu, “Using multiple-precision arithmetic to prevent low-frequency breakdowns in the diagonalization of the green’s function,” *Progress in Electromagnetics Research Symp. (PIERS)*, pp. 2311–2315, 2014.

[70] I.-T. Chiang and W.-C. Chew, “New formulation and iterative solution for low-frequency volume integral equation,” *J. Electromagnet Wave*, vol. 19, no. 3, pp. 289–306, 2005.

[71] A. Polimeridis, J. Villena, L. Daniel, and J. White, “Stable FFT-JVIE solvers for fast analysis of highly inhomogeneous dielectric objects,” *J. Comput. Phys.*, vol. 269, pp. 280–296, 2014.

[72] X. Millard and Q. H. Liu, “A fast volume integral equation solver for electromagnetic scattering from large inhomogeneous objects in planarly layered media,” *IEEE Trans. Antennas Propag.*, vol. 51, no. 9, pp. 2393–2401, 2003.

[73] K. Yang and A. E. Yilmaz, “Integral equation solution of 3-d anisotropic lossy dielectrics in uniaxial layered media,” in *Antenn. Propag. Soc. Int. Symp.*, 2012.

[74] M. Li and W. Chew, “Applying divergence-free condition in solving the volume integral equation,” *Progress In Electromagnetics Research*, vol. 57, pp. 311–333, 2006.

[75] D. Schaubert, D. Wilton, and a. Glisson, “A tetrahedral modeling method for electromagnetic scattering by arbitrarily shaped inhomogeneous dielectric bodies,” *IEEE Trans. Antennas Propag.*, vol. 32, no. 1, pp. 77–85, 1984.

[76] J. P. Kottmann and O. J. F. Martin, “Accurate solution of the volume integral equation for high-permittivity scatterers,” *IEEE Trans. Antennas Propag.*, vol. 48, no. 11, pp. 1719–1726, 2000.

[77] M. M. Botha, “Solving the volume integral equations of electromagnetic scattering,” *J. Comput. Phys.*, vol. 218, no. 1, pp. 141–158, 2006.

[78] J. Markkanen, C. C. Lu, X. Cao, and P. Ylä-Oijala, “Analysis of volume integral equation formulations for scattering by high-contrast penetrable objects,” *IEEE Trans. Antennas Propag.*, vol. 60, no. 5, pp. 2367–2374, 2012.

[79] C.-C. Lu, P. Yla-Oijala, M. Taskinen, and J. Sarvas, “Comparison of two volume integral equation formulations for solving electromagnetic scattering by inhomogeneous dielectric objects,” *Antenn. Propag. Soc. Int. Symp.*, pp. 1–4, June 2009.

[80] L. Sun and W. Chew, “A novel formulation of the volume integral equation for general large electromagnetic scattering problems,” *Antenn. Propag. Soc. Int. Symp.*, vol. I, pp. 1–4, Jul. 2008.

[81] X. C. Nie, N. Yuan, L. W. Li, Y. B. Gan, and T. S. Yeo, “A fast combined field volume integral equation solution to EM scattering by 3-D dielectric objects of arbitrary permittivity and permeability,” *IEEE Trans. Antennas Propag.*, vol. 54, no. 3, pp. 961–969, 2006.

[82] M. I. Sancer, K. Sertel, J. L. Volakis, and P. Van Alstine, “On volume integral equations,” *IEEE Trans. Antennas Propag.*, vol. 54, no. 5, pp. 1488–1495, 2006.

[83] D. X. Wang, E. K. N. Yung, R. S. Chen, and P. Y. Lau, “An efficient volume integral equation solution to EM scattering by complex bodies with inhomogeneous bi-isotropy,” *IEEE Trans. Antennas Propag.*, vol. 55, pp. 1970–1980, July 2007.

[84] G. Kobidze and B. Shanker, “Integral equation based analysis of scattering from 3-D inhomogeneous anisotropic bodies,” *IEEE Trans. Antennas Propag.*, vol. 52, pp. 2650–2658, Oct. 2004.

[85] J. Markkanen, P. Yla-Oijala, and A. Sihvola, “Discretization of volume integral equation formulations for extremely anisotropic materials,” *IEEE Trans. Antennas Propag.*, vol. 60, pp. 5195–5202, Nov. 2012.

[86] A. F. Peterson, “Volume integral equations for electromagnetic scattering from dielectric objects: observations and questions,” in *Forum for Electromagnetic Research Methods and Application Technologies (FERMAT)*, vol. 5, pp. 1–10, Sep.–Nov. 2014.

[87] S. Järvenpää, M. Taskinen, and P. Ylä-Oijala, “Singularity extraction technique for integral equation methods with higher order basis functions on plane triangles and tetrahedra,” *Int. J. Numer. Meth. Eng.*, vol. 58, pp. 1149–1165, Oct. 2003.