

NOVEL METHODS FOR FAST AND ACCURATE SOLUTIONS OF ELECTROMAGNETIC INTEGRAL EQUATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

By
Enes Koç
July 2025

Novel Methods for Fast and Accurate Solutions of Electromagnetic
Integral Equations

By Enes Koç

July 2025

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

Vakur Behçet Ertürk(Advisor)

Mert Kalfa(Co-Advisor)

Lale Alatan

Özlem Aydın Çivi

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

NOVEL METHODS FOR FAST AND ACCURATE SOLUTIONS OF ELECTROMAGNETIC INTEGRAL EQUATIONS

Enes Koç

M.S. in Electrical and Electronics Engineering

Advisor: Vakur Behçet Ertürk

Co-Advisor: Mert Kalfa

July 2025

As the scale and complexity of electromagnetic scattering problems continue to grow, solving frequency-domain integral equations (IEs) with high accuracy and computational efficiency presents increasing challenges. To address these, two novel methods are proposed, each offering robust alternatives to conventional approaches such as the multilevel fast multipole algorithm (MLFMA), which often suffers from the low-frequency breakdown (LFB) problem and inefficiencies in complex or elongated geometries. The first method introduces a broadband and highly parallelizable framework based on machine learning for accelerating the iterative solution of frequency-domain IEs. Far-zone interactions are evaluated through a group-by-group interaction model using a one-box buffer scheme. The scatterer is divided into uniformly sized boxes, and each subdomain basis functions within boxes are represented by a fixed number of Hertzian dipoles regarded as uniform basis functions. Neural networks trained on the dipole Green's function are employed to model far-field interactions, enabling accurate and reusable computations independent of problem geometry, provided that box sizes and relative distances are fixed. This framework eliminates the LFB problem and provides strong scalability in parallel environments. Its accuracy and efficiency are validated through comparisons with MLFMA and Mie series results, along with parallel scalability benchmarks. The second method focuses on improving far-zone interaction efficiency through a hierarchical strategy that incorporates Taylor expansions and virtual box decomposition. While maintaining the hierarchical tree structure of MLFMA, each box of size a is subdivided into virtual boxes of size $\sqrt{a}$, and interactions are approximated using Taylor expansions of the reduced Green's function. This setting ensures that the number of expansion

coefficients within virtual boxes remains constant with increasing box size, preserving $\mathcal{O}(N \log N)$ complexity even for electrically large or quasi-one-dimensional geometries where MLFMA struggles. Unlike matrix-decomposition-based methods, the proposed scheme relies entirely on algebraic operations. The method's robustness is demonstrated through numerical simulations that confirm its accuracy and efficiency relative to MLFMA.



Keywords: Integral equations, Fast solutions.

ÖZET

ELEKTROMANYETİK İNTEGRAL DENKLEMLERİN HIZLI VE DOĞRU ÇÖZÜMLERİ İÇİN YENİ YÖNTEMLER

Enes Koç

Elektrik ve Elektronik Mühendisliği, Yüksek Lisans

Tez Danışmanı: Vakur Behçet Ertürk

İkinci Tez Danışmanı: Mert Kalfa

Temmuz 2025

Elektromanyetik saçılma problemlerinin ölçeği ve karmaşıklığı arttıkça, frekans bölgesi integral denklemlerinin (İD) yüksek doğrulukta ve hesaplama açısından verimli şekilde çözülmesi giderek artan zorluklar ortaya çıkarmaktadır. Bu duruma yönelik olarak, geleneksel yöntemlere güçlü alternatifler sunan iki yeni yöntem önerilmektedir. Özellikle, Çok Seviyeli Hızlı Çokkutupsal Yöntemi (ÇSHÇY) gibi yaklaşımlar, düşük frekans bozulması (DFB) sorunu ve karmaşık ya da uzamış geometrilerde verimlilik kaybı yaşamakta ve bu yöntemler bu sınırlamaları aşmayı hedeflemektedir. Birinci yöntem, frekans bölgesi İD'lerinin yinelemeli çözümünü hızlandırmak amacıyla makine öğrenmesine dayalı, geniş frekans aralığında çalışan ve yüksek düzeyde paralelleştirilebilir bir çerçeve sunmaktadır. Uzak bölge etkileşimleri, tek kutulu tamponlama şeması kullanan grup-bazlı bir etkileşim modeli ile değerlendirilir. Saıcı yapı, eş boyutlu kutulara bölünür ve her kutudaki alt bölge taban fonksiyonları, sabit sayıda Hertz dipolü ile temsil edilir. Bu dipoller, düzgün taban fonksiyonları olarak kabul edilir. Hertz dipolünün Green fonksiyonu üzerinde eğitilmiş sinir ağları, uzak alan etkileşimlerini modellemek için kullanılır. Bu sayede, kutu boyutları ve göreceli uzaklıklar sabit tutulduğu sürece, geometri bağımsız doğru ve yeniden kullanılabilir hesaplamalar elde edilir. Yöntem, DFB problemini ortadan kaldırmakta ve paralel ortamlarda güçlü ölçeklenebilirlik sunmaktadır. Doğruluk ve verimlilik performansı, ÇSHÇY ve Mie serisi ile yapılan karşılaştırmalar ile paralel ölçeklenebilirlik deneyleri üzerinden doğrulanmıştır. İkinci yöntem, uzak bölge etkileşimlerinin verimliliğini artırmak amacıyla Taylor açılımları ve sanal kutu ayrıştırmasını içeren hiyerarşik bir strateji önermektedir. ÇSHÇY'nın

hiyerarşik ağaç yapısı korunmakta, her bir a boyutundaki kutu $\sqrt{a}$ boyutunda sanal kutulara ayrılmaktadır. Bu kutular arasındaki etkileşimler, indirgenmiş Green fonksiyonunun Taylor açılımları kullanılarak yaklaşık şekilde hesaplanmaktadır. Bu yapı sayesinde, sanal kutular içerisindeki açılım katsayılarının sayısı kutu boyutundan bağımsız sabit kalmakta ve böylece yöntem, ÇSHÇY'nın zorlandığı elektriksel olarak büyük veya yarı tek boyutlu geometrilere bile $\mathcal{O}(N \log N)$ karmaşıklığını koruyabilmektedir. Önerilen yaklaşım, matris ayrıştırmalarına dayanmamakta ve tamamen cebirsel işlemler kullanılarak uygulanmaktadır. Sayısal deneylerle yapılan değerlendirmeler, yöntemin ÇSHÇY'ya kıyasla doğruluk ve verimlilik açısından üstün bir performans sergilediğini göstermektedir.

Anahtar sözcükler: İntegral denklemler, Hızlı çözümler.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Vakur B. Ertürk, and my co-advisor, Dr. Mert Kalfa, for their invaluable guidance, support, and encouragement throughout my graduate studies. Their mentorship has not only helped shape the direction of this thesis but also made me truly enjoy the process of research. Their enthusiasm, insight, and support have been instrumental in fostering both my academic and personal growth, and I feel incredibly fortunate to have worked under their supervision.

I sincerely thank my thesis committee members, Assoc. Prof. Lale Alatan and Prof. Özlem A. Çivi, for their kind interest in my work and for taking the time to read and evaluate my thesis.

I am also grateful to Irmak E. Fitoz, whose contributions to the method presented in the second part of this thesis were highly valuable.

I would like to thank Prof. Ayhan Altıntaş for the many insightful and stimulating conversations we have had throughout my studies. His perspectives have greatly enriched my understanding and deepened my appreciation of the subject matter.

Special thanks to Prof. Tolga M. Duman for providing access to computational resources that were essential for the implementation and testing of the methods proposed in this thesis.

I would like to extend my heartfelt thanks to Artun, Cahit, Doğaç, Ege A., Ege K., Furkan, Onat, Seher, and Tuna, whose companionship, encouragement, and occasional distractions helped make the ups and downs of graduate school more manageable. Over the past three years, I have deeply enjoyed the intellectually stimulating discussions we had on a wide range of topics, as well as our hangouts and the many memorable moments we shared.

Finally, I am deeply thankful to my family for their unwavering love, patience,

and support throughout this journey and every stage of my life. Their constant encouragement, understanding, and belief in me, especially during the most challenging times, have been a source of immense strength and motivation. No matter the circumstances, they have always stood by my side, and for that, I am forever grateful.

In addition, I would like to gratefully acknowledge the financial support provided by Turkcell, as part of the 5G and Beyond Joint Graduate Support Programme coordinated by the Information and Communication Technologies Authority. I am also thankful to the TÜBİTAK 2210 Graduate Scholarship Program for its support throughout my graduate studies.

Contents

1	Introduction	1
2	Integral Equation Formulations and Method of Moments	8
2.1	Integral Equation Formulations	8
2.1.1	Maxwell's Equations in the Frequency Domain	8
2.1.2	Surface and Volume Equivalence Principles	10
2.1.3	Electric Field Integral Equation (EFIE) for PEC Surfaces .	12
2.1.4	Volume Electric Field Integral Equation (VEFIE)	13
2.2	Method of Moments (MoM)	14
2.2.1	Discretization of IEs	18
3	A Novel Highly Parallelizable ML-Based Method for the Fast Solution of IEs	19
3.1	Formulation	20
3.1.1	Mapping to Uniform Basis Functions	21

3.1.2	Translation	25
3.1.3	Inverse Mapping to Subdomain Functions	28
3.1.4	Discussions on the Merits of the Proposed Approach . . .	30
3.2	ML Implementation	31
3.3	Parallelization	35
3.4	Memory and Time Complexity of the Proposed Method	38
3.5	Numerical Results	40
4	A Novel Taylor Expansion and Virtual Box Based Hierarchical Method for the Fast Solution of IEs	50
4.1	Matrix-Vector Product Evaluation	52
4.2	Taylor Approximation of Green's Function	55
4.3	Virtual Box Structure	65
4.4	Implementation of the Proposed Method	71
4.4.1	Aggregation over Virtual Boxes	75
4.4.2	Translation over Virtual Boxes	76
4.4.3	Disaggregation over Virtual Boxes	77
4.5	Memory and Time Complexity of the Proposed Method	78
4.6	Numerical Results	81
5	Conclusion and Future Work	87

<i>CONTENTS</i>	xi
A Finding coefficients of Taylor Expansion	96
B Derivation of R Expansion	99
C An Example of Taylor Expansion	101



List of Figures

2.1	Surface equivalence diagram.	10
2.2	An RWG function.	16
2.3	An SWG function.	17
3.1	2D projection of a sample scatterer inside a mother box, which is divided into child boxes. For the selected box SB , NF boxes correspond to near interactions, and FF boxes correspond to far interactions.	21
3.2	Mapping from subdomain basis functions to uniform basis functions	22
3.3	Illustration of the translation stage.	26
3.4	Inverse mapping from uniform basis functions to subdomain functions	29
3.5	Illustration of pseudo-randomly surface set generation using a PEC sphere discretized with RWGs.	32
3.6	Training and validation losses at the first and second steps of the training for the $[3, 3, 3]^T$, $a = \lambda/8$ translation network.	34

3.7	Normalized execution time of the proposed method for the far-zone MVMs of a 0.6m diameter PEC sphere discretized with 114318 basis functions. Illumination is a unit amplitude uniform plane wave at 1 GHz.	37
3.8	Far-zone co-polarized electric fields scattering from a PEC sphere of diameter 0.6m (2λ) illuminated with a unit amplitude uniform plane wave at 1 GHz. The discretized surface consists of 114318 basis functions.	40
3.9	Far-zone co-polarized electric fields scattering from a 0.6m diameter dielectric sphere with $\epsilon_r = 2$, illuminated with a unit amplitude uniform plane wave at 250 MHz. The volume of the sphere is discretized with 38169 SWG functions.	41
3.10	Far-zone co-polarized electric fields scattering from a two-layer dielectric sphere with the core diameter 0.3 m ($\epsilon_r = 31$) and the thickness is 0.15 m ($\epsilon_r = 2$), illuminated with a unit amplitude uniform plane wave at 500 MHz. The entire volume is discretized with 259393 SWG functions.	42
3.11	Far-zone electric field along the E-plane ($\phi = 0^\circ$) scattered from a 0.6 m PEC Flamme model illuminated with a unit amplitude uniform plane wave impinging from $(\phi, \theta) = (0, 60^\circ)$, i.e., 30° elevation from the nose at 1 GHz. The surface of the Flamme is discretized with 51669 RWG functions.	43
3.12	Comparison of the induced current density magnitudes for the (a) proposed method and (b) 4-level MLFMA, when the PEC Flamme is illuminated with a unit amplitude uniform plane wave impinging from $(\phi, \theta) = (0, 60^\circ)$, i.e., 30° elevation from the nose at 1 GHz. .	45

3.13	Far-zone electric field along the E-plane scattered from a 0.6 m PEC NASA Almond model illuminated with a unit amplitude uniform plane at 1 GHz. The surface of the NASA Almond is discretized with 105159 RWG functions.	46
3.14	Comparison of the induced current density magnitudes for the (a) proposed method and (b) 4-level MLFMA, when the PEC NASA Almond is illuminated with a unit amplitude uniform plane at 1 GHz.	47
3.15	Far-zone co-polarized electric fields scattering from a PEC sphere of diameter 0.6m ($\lambda/64$) illuminated with a unit amplitude uniform plane wave at 7.8125 MHz. The discretized surface consists of 25995 basis functions.	48
4.1	Recursive division (2D view for an arbitrary 3D object)	53
4.2	Near-field (NF) and Far-field (FF) for a selected box (in red) . . .	54
4.3	Virtual box structure	66
4.4	Vectors for virtual box translation	67
4.5	Hierarchical tree (in black) and information flow between levels (in red and blue).	72
4.6	An example division of boxes into virtual boxes.	73
4.7	The possible child virtual boxes of a virtual box at level l	74
4.8	Far-zone co-polarized electric fields scattering from a PEC dipole scatterer with dimensions $(8, 0.04, 0.04) \times \lambda$ illuminated with a unit amplitude uniform plane wave at 118.812 MHz. The discretized surface consists of 38,592 basis functions.	83

- 4.9 Far-zone co-polarized electric fields scattering from a PEC plate with dimensions $(8, 8, 0.08) \times \lambda$ illuminated with a unit amplitude uniform plane wave at 120 MHz. The discretized surface consists of 183,060 basis functions. 84
- 4.10 Comparison of the induced current density magnitudes for the (a) proposed method and (b) 6-level MLFMA, when the PEC plate is illuminated with a unit amplitude uniform plane wave at 120 MHz. 86

List of Tables

3.1	ML Model Parameters	33
3.2	Test Set Errors of Several Networks	33
3.3	Memory Requirements and CPU Time Results for 2λ Sized Sphere. The L value is the same in MLFMA and the proposed method. . .	41
3.4	Memory Requirements and CPU Time Results for λ -Sized Two- Layer Dielectric Sphere. The L value is 4 in both MLFMA and the proposed method.	42
3.5	Memory Requirements and CPU Time Results 2λ Sized Flamme. The L value is 4 in MLFMA and 5 in the proposed method. . . .	44
3.6	Memory Requirements and CPU Time Results for 2λ Sized NASA Almond. The L value is 4 in MLFMA and 5 in the proposed method.	46
3.7	Memory Requirements and CPU Time Results for $\lambda/64$ Sized Sphere. The L value is 3 in the proposed method.	48
4.1	The CPU Time Results for dipole scatterer. The L value is 6 and the same in both MLFMA and the proposed method.	83
4.2	CPU Time Results for PEC plate. The L value is the same in MLFMA and the proposed method.	85

Chapter 1

Introduction

Electromagnetic (EM) radiation and scattering problems play a critical role in many scientific and engineering fields, such as wireless communications, radar systems, antenna design, electromagnetic compatibility, remote sensing, and biomedical imaging [1–6]. Accurately modeling these problems is essential for the reliable prediction and design of real-world systems. Among the various numerical approaches developed for such purposes, frequency-domain integral equation (IE) methods have become particularly prominent. One of their most important advantages is their ability to model open-region problems without requiring volumetric discretization of free space, leading to significantly reduced computational costs compared to differential-equation-based solvers. Instead, the unknown quantities, typically equivalent currents or fields, are confined to material interfaces, enabling compact representations that inherently satisfy the radiation condition at infinity.

The method of moments (MoM) has long been the standard technique for the numerical discretization of IEs [1]. In this method, the integral equations are converted into a dense system of linear equations by expressing the unknown fields in terms of a set of basis functions and enforcing the equations through testing procedures. MoM provides high accuracy and has become a foundational tool in computational electromagnetics. However, as the number of unknowns

increases due to fine geometrical features or high operating frequencies, the dense matrix structure of MoM becomes a significant limitation. The resulting matrix-vector equation can be solved either directly, which has a time complexity of $\mathcal{O}(N^3)$ where N is the number of unknowns, or iteratively using methods such as Krylov subspace solvers. In iterative methods, each matrix-vector multiplication (MVM) requires $\mathcal{O}(N^2)$ operations. In both approaches, the storage requirement is $\mathcal{O}(N^2)$, and the storage and computational time quickly become prohibitive for electrically large or complex geometries.

To overcome these limitations, a variety of fast algorithms have been introduced over the past decades. One of the most impactful developments in this regard is the fast multipole method (FMM) [7] and its hierarchical extension, the multilevel fast multipole algorithm (MLFMA) [8], which revolutionized the solution of large-scale scattering problems by dramatically reducing the computational complexity of MVMs to $\mathcal{O}(N^{3/2})$ with FMM and to $\mathcal{O}(N \log N)$ with MLFMA. MLFMA achieves this by organizing the computational domain into a hierarchy of clusters and using analytical approximations for interactions between distant groups of basis functions. Through a multilevel representation of the Green's function, MLFMA enables the efficient evaluation of far-field interactions while preserving exact computation of near-field interactions. Its hierarchical nature makes it particularly suitable for electrically large problems, allowing the simulation of geometries with millions of unknowns on relatively modest hardware.

Despite these advantages, MLFMA still faces challenges when applied to real-world problems. One of the most pressing issues is its efficiency when handling multiscale geometries. In such cases, the conventional MLFMA may produce trees with excessive depth or with uneven distributions of basis functions across clusters, leading to inefficient memory use and increased runtime. To address this issue, several modifications to the conventional MLFMA structure have been proposed. Notably, incomplete-leaf MLFMA (IL-MLFMA) introduces adaptivity into the hierarchical structure by allowing variable-depth clustering based on the local density of basis functions [9, 10]. This yields significant memory savings and computational efficiency, especially in problems involving a mixture of fine and

coarse features.

In parallel, researchers have developed additional techniques to further accelerate large-scale computations. Methods such as the Adaptive Integral Method (AIM) [11] and the Discrete Dipole Approximation (DDA) [12] represent the current distribution in the scatterer using uniformly spaced dipoles or point sources. They then leverage Fast Fourier Transform (FFT)-based accelerators, enabled by the regular 3D grid structure, to efficiently compute matrix-vector products, which are calculated as a whole over the global domain. Skeletonization techniques, for instance, exploit the fact that far-field interaction matrices are often numerically low-rank. By identifying a reduced set of representative basis functions (or “skeletons”), these techniques compress translation operators and reduce the number of computations required per interaction [13]. Such low-rank approximations enhance efficiency and help reduce memory consumption, an important consideration in simulations of large scatterers. Moreover, new parallelization strategies have been introduced to distribute computational workloads more evenly across multiple processors [14, 15], including hierarchical partitioning of the domain and dynamic scheduling to minimize idle time in high-performance computing environments. Nevertheless, achieving strong scalability for parallelization remains an ongoing challenge due to the inherently irregular distribution of computational work, often dictated by scatterer geometry.

Another major challenge in MLFMA is the low-frequency breakdown (LFB) problem, which arises when the separation between interacting groups of basis functions becomes small relative to the operating wavelength. In such cases, the assumptions underlying multipole expansions break down, leading to numerical instability and a loss of accuracy [16–18]. This is particularly problematic in multiscale problems, where parts of the geometry operate at very different electrical sizes. Various strategies have been proposed to address LFB, including alternative formulations of translation operators that better capture near-field effects [19, 20], and modified expansions that remain valid in the low-frequency regime [21, 22]. Other remedies involve adapting numerical integration paths to incorporate evanescent wave contributions [16]. While these techniques improve

low-frequency accuracy, they often increase algorithmic complexity and computational cost. Alternatively, multiple-precision arithmetic can be employed to mitigate numerical errors [23], though this typically requires specialized hardware.

A further limitation of the conventional MLFMA arises when dealing with elongated or quasi-one-dimensional geometries. In such scenarios, the asymptotic computational complexity of MLFMA may approach $\mathcal{O}(N^2)$, since its assumptions about spatial locality and balanced clustering break down. As a result, specialized algorithms have been developed to adapt MLFMA’s hierarchical structure and translation mechanisms to better suit elongated domains [24]. To efficiently treat these configurations, directional and adaptive techniques have been proposed. In [25], high-oscillatory integrals are evaluated using directional low-rank approximations that exploit the directional decay and smoothness of the Green’s function. A Chebyshev interpolation-based directional summation method [26] was developed to accelerate oscillatory kernel evaluations while maintaining high accuracy in anisotropic settings. In the spectral domain, Gaussian beam representations [27] enable angularly localized interactions, and their analytical diagonalization ensures both speed and precision. Extending this framework, the directional multilevel complex-space fast multipole algorithm (DMLCSFMA) [24] incorporates Gaussian beams into a multilevel hierarchy, achieving broadband, flexible simulations with $\mathcal{O}(N \log N)$ complexity. These methods collectively exploit directional smoothness and anisotropic interaction patterns to improve efficiency and robustness for elongated structures. Yet, there is still a need to develop better algorithms and methods to address these problems.

In recent years, the emergence of machine learning (ML) has opened new avenues for improving the efficiency and flexibility of computational EM solvers. Initially applied to antenna optimization and inverse scattering [28–31], ML techniques are now being integrated into the numerical solution of IEs. These efforts fall into two main categories. The first category uses ML as a surrogate model to directly predict EM responses, such as scattered fields or radar cross-sections, based on geometry or material parameters [32,33]. Once trained on large datasets, these models yield accurate predictions at negligible cost. The second category

embeds ML components into traditional solvers: neural networks (NNs) have been used to accelerate MVMs [34], enhance iterative solver convergence [35, 36], or approximate complex translation operators in MLFMA [37–39]. These hybrid strategies retain the physical accuracy of traditional methods while benefiting from ML’s speed and adaptability. A key advantage is generalizability: trained models can be reused across various simulations without retraining. These developments suggest that ML can play a transformative role in future EM solvers by addressing key limitations of existing algorithms. ML-based acceleration is especially promising for mitigating the parallelization and low-frequency bottlenecks of MLFMA. By learning patterns in Green’s function interactions and iterative behavior, ML can augment conventional techniques across diverse applications.

Consequently, the increasing demand to analyze more complex and large/multi-scale scatterers highlights the need for new, more efficient computational techniques. To address this necessity, in the first part of this thesis, a new, broadband, and highly parallelizable approach for the fast and accurate iterative solution of frequency-domain IEs is presented, designed to overcome the limitations discussed earlier. The core of this approach is an ML-based, group-by-group interaction framework that accurately accelerates the evaluation of far-zone interactions by utilizing a one-box buffer scheme during MVMs. The method starts by clustering the scatterer geometry into equally sized boxes, similar to what is done in FMM/MLFMA. Within each box, the subdomain basis functions used for discretizing the geometry are represented by a fixed number of arbitrarily oriented and uniformly distributed Hertzian dipoles, called uniform basis functions (other uniform basis function types can be used in the framework, but Hertzian dipoles are preferred due to the simplicity of their Green’s function expressions). These fixed number of uniform basis functions are placed within each box to enable the use of pre-trained ML models for far-zone interaction computation. This stands in contrast to AIM and DDA, which calculate interactions as a whole using FFT-based acceleration over globally defined dipole grids. Then, far-field interactions between these dipoles are computed in a group-wise fashion using ML models trained on the Hertzian dipole’s Green’s function (HDGF). This strategy offers

two main advantages: (A) Identical and workload-balanced box-to-box interactions ensure strong scalability on parallel architectures. (B) Incorporating the HDGF into the ML training eliminates the LFB problem, enabling an accurate and broadband solution applicable to multi-scale problems. Additionally, since training is performed offline, the trained models can be reused for different scatterers (both surface and volume IEs) without retraining, as long as the box sizes and relative distances remain the same. The proposed method’s accuracy and efficiency are validated by comparing its scattering results with those from the conventional MLFMA and, where applicable, Mie series solutions. The method’s robustness against LFB is also verified. Finally, scalability experiments confirm the method’s parallelization effectiveness.

The second part of this thesis introduces a novel hierarchical method based on virtual boxes and Taylor expansions to compute far-zone interactions for the fast and accurate iterative solution of frequency-domain IEs, particularly aimed at improving efficiency in electrically large and elongated problems. The foundation of the proposed method lies in two critical elements: the subdivision of each box into virtual boxes, a strategy independently developed in [25], and the use of Taylor expansions to approximate Green’s functions [40–42], as explored in several prior works in computational electromagnetics. These two components form the core of our hierarchical approach to computing far-zone interactions. The proposed method shares the same hierarchical tree structure as MLFMA, where the geometry is recursively partitioned into boxes that form the nodes of a tree. Each node represents a cluster of subdomain basis functions, and the tree is used to identify far-zone interactions between these boxes. Specifically, each box of size a is subdivided into virtual boxes of size $\sqrt{a}$, and the far-zone interaction between two boxes is computed as the sum of virtual box-to-virtual box interactions. These interactions are approximated using Taylor expansions of the reduced Green’s function (excluding the plane wave contribution). A key advantage of this setting is that, due to the small size of virtual boxes, the required number of Taylor expansion coefficients remains constant even as the electrical size of the parent boxes increases. This ensures that the method maintains $\mathcal{O}(N \log N)$ complexity. In the case of elongated geometries, the number of

virtual box pairs remains sparse. This structural sparsity allows box-to-box interactions to be evaluated efficiently via a reduced number of virtual box-to-virtual box interactions, thereby preserving the $\mathcal{O}(N \log N)$ complexity in such scenarios, where the conventional MLFMA suffers from keeping $\mathcal{O}(N \log N)$ complexity. Furthermore, since the Green's function is approximated purely via Taylor expansions, the method avoids matrix decompositions entirely, and the required translations are carried out using simple algebraic operations. Additionally, due to the algebraic form of the Taylor expansion terms, several of the resulting operations can be reformulated as multi-dimensional convolutions, which can be efficiently accelerated using FFT-based techniques. Finally, the accuracy and efficiency of the proposed method are validated through numerical experiments and benchmark comparisons with conventional MLFMA, including analyses of CPU time and scalability.

The remainder of this thesis is organized as follows: Chapter 2 reviews fundamental IE formulations in EM and introduces the basic principles of MoM. Chapter 3 presents a novel, highly parallelizable ML-based method for fast IE solutions, together with the numerical results to assess the method's accuracy and efficiency. Chapter 4 introduces a hierarchical virtual-box and Taylor-expansion method for fast EM scattering analysis for elongated & quasi-one-dimensional objects, and provides preliminary numerical results for these geometries. These preliminary results are compared with those of MLFMA to validate the method's accuracy and efficiency. Chapter 5 concludes the thesis with a summary of key findings and outlines future directions for fast and accurate IE solvers. Throughout this thesis, a time-harmonic convention with an implicit $\exp(j\omega t)$ time dependence is adopted, where $\omega = 2\pi f$ and f denotes the operating frequency. For clarity, this time dependence is omitted in the subsequent chapters.

Chapter 2

Integral Equation Formulations and Method of Moments

2.1 Integral Equation Formulations

2.1.1 Maxwell's Equations in the Frequency Domain

In an isotropic, linear medium with time-harmonic sources at angular frequency ω , Maxwell's equations in the frequency (phasor) domain are expressed as:

$$\nabla \times \bar{E}(\bar{r}) = -j\omega\bar{B}(\bar{r}) - \bar{M}(\bar{r}), \quad (2.1)$$

$$\nabla \times \bar{H}(\bar{r}) = j\omega\bar{D}(\bar{r}) + \bar{J}(\bar{r}), \quad (2.2)$$

$$\nabla \cdot \bar{D}(\bar{r}) = \rho_e(\bar{r}), \quad (2.3)$$

$$\nabla \cdot \bar{B}(\bar{r}) = \rho_m(\bar{r}), \quad (2.4)$$

where $\bar{J}$ and ρ_e represent electric current and charge densities, while $\bar{M}$ and ρ_m denote the fictitious magnetic current and charge densities, respectively. These equations are supplemented by the continuity equations between currents and

charges, which ensure local charge conservation:

$$\nabla \cdot \bar{J} = -j\omega\rho_e, \quad (2.5)$$

$$\nabla \cdot \bar{M} = -j\omega\rho_m. \quad (2.6)$$

Assuming a homogeneous medium, the electric and magnetic fields can be expressed in terms of vector and scalar potentials:

$$\bar{E}(\bar{r}) = -j\omega\bar{A}(\bar{r}) - \nabla\phi_e(\bar{r}) - \frac{1}{\varepsilon}\nabla \times \bar{F}(\bar{r}), \quad (2.7)$$

$$\bar{H}(\bar{r}) = -j\omega\bar{F}(\bar{r}) - \nabla\phi_m(\bar{r}) + \frac{1}{\mu}\nabla \times \bar{A}(\bar{r}). \quad (2.8)$$

Here, $\bar{A}$ and $\bar{F}$ denote the electric and magnetic vector potentials, and ϕ_e , ϕ_m are the respective scalar potentials. These potentials provide an elegant means of expressing the fields in a gauge-invariant manner.

To ensure the uniqueness of the potentials, the Lorenz gauge condition is imposed:

$$\nabla \cdot \bar{A}(\bar{r}) = -j\omega\varepsilon\mu\phi_e(\bar{r}), \quad (2.9)$$

$$\nabla \cdot \bar{F}(\bar{r}) = -j\omega\varepsilon\mu\phi_m(\bar{r}), \quad (2.10)$$

where ε is the permittivity and μ is the permeability of the medium. Substituting (2.7)–(2.8) into Maxwell's equations (2.1)–(2.4), and applying the Lorenz gauge, lead to Helmholtz-type equations for the potentials given by

$$\nabla^2 \bar{A}(\bar{r}) + k^2 \bar{A}(\bar{r}) = -\mu \bar{J}(\bar{r}), \quad (2.11)$$

$$\nabla^2 \phi_e(\bar{r}) + k^2 \phi_e(\bar{r}) = -\frac{\rho_e(\bar{r})}{\varepsilon}, \quad (2.12)$$

$$\nabla^2 \bar{F}(\bar{r}) + k^2 \bar{F}(\bar{r}) = -\varepsilon \bar{M}(\bar{r}), \quad (2.13)$$

$$\nabla^2 \phi_m(\bar{r}) + k^2 \phi_m(\bar{r}) = -\frac{\rho_m(\bar{r})}{\mu}, \quad (2.14)$$

where $k = \omega\sqrt{\mu\varepsilon}$ is the wavenumber of the medium.

The scalar Green's function for the Helmholtz equation in an unbounded homogeneous medium is given by

$$g(\bar{r}, \bar{r}') = \frac{e^{-jk|\bar{r}-\bar{r}'|}}{4\pi|\bar{r}-\bar{r}'|}. \quad (2.15)$$

Using this Green's function, the potential fields can be expressed in integral form as follows:

$$\phi_e(\bar{r}) = \frac{1}{\varepsilon} \int_V \rho_e(\bar{r}') g(\bar{r}, \bar{r}') d\bar{r}', \quad (2.16)$$

$$\phi_m(\bar{r}) = \frac{1}{\mu} \int_V \rho_m(\bar{r}') g(\bar{r}, \bar{r}') d\bar{r}', \quad (2.17)$$

$$\bar{A}(\bar{r}) = \mu \int_V \bar{J}(\bar{r}') g(\bar{r}, \bar{r}') d\bar{r}', \quad (2.18)$$

$$\bar{F}(\bar{r}) = \varepsilon \int_V \bar{M}(\bar{r}') g(\bar{r}, \bar{r}') d\bar{r}'. \quad (2.19)$$

2.1.2 Surface and Volume Equivalence Principles

The equivalence principle is foundational in computational electromagnetics. It provides the means to replace the actual scatterer with equivalent currents that reproduce the same fields in a region of interest.

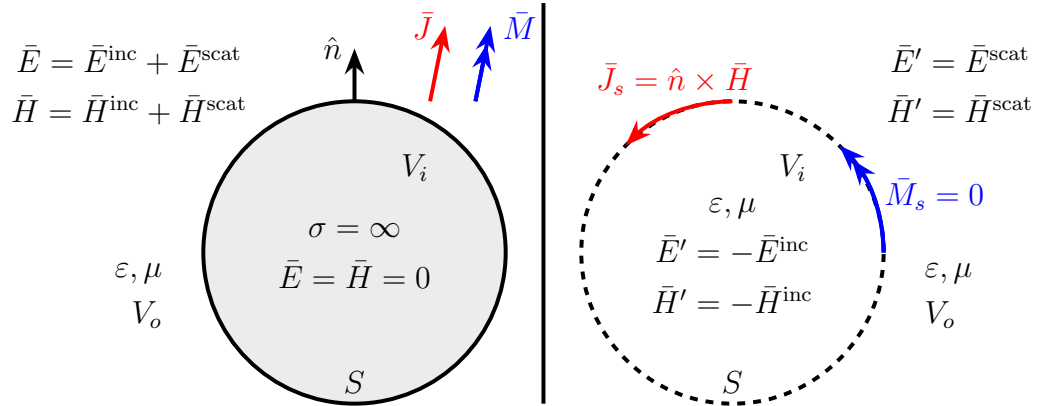


Figure 2.1: Surface equivalence diagram.

Surface Equivalence Principle: As illustrated in Fig. 2.1, consider a closed surface S that encloses a volume V_i containing a perfectly electrically conducting (PEC) scatterer (i.e., $\sigma = \infty$). The presence of the PEC object gives rise to scattered electric and magnetic fields, denoted by $\bar{E}^{\text{scat}}$ and $\bar{H}^{\text{scat}}$, respectively, in the exterior volume V_o , which is assumed to be a homogeneous medium characterized by permittivity ε and permeability μ . Let the impressed electric and magnetic

sources be represented by $\bar{J}$ and $\bar{M}$, which generate the incident fields $\bar{E}^{\text{inc}}$ and $\bar{H}^{\text{inc}}$ in the absence of the PEC scatterer. The total fields in the presence of the scatterer are given by $\bar{E} = \bar{E}^{\text{inc}} + \bar{E}^{\text{scat}}$ and $\bar{H} = \bar{H}^{\text{inc}} + \bar{H}^{\text{scat}}$. According to the surface equivalence principle, the PEC scatterer inside S can be replaced by equivalent surface current densities on S that reproduce the same external field. These equivalent surface currents are defined as

$$\bar{J}_s = \hat{n} \times \bar{H} = \hat{n} \times (\bar{H}^{\text{inc}} + \bar{H}^{\text{scat}}), \quad (2.20)$$

$$\bar{M}_s = \bar{E} \times \hat{n} = (\bar{E}^{\text{inc}} + \bar{E}^{\text{scat}}) \times \hat{n} = 0, \quad (2.21)$$

where $\hat{n}$ is the unit normal vector pointing outward from the surface S . In the equivalent problem, the interior volume V_i is treated as the same homogeneous medium as the exterior region V_o (i.e., with the same ε and μ). In this scenario, the surface current $\bar{J}_s$ placed on S generate fields $\bar{E}'$ and $\bar{H}'$ in V_o that exactly reproduce the original scattered fields $\bar{E}^{\text{scat}}$ and $\bar{H}^{\text{scat}}$, while in V_i , the fields are given by $\bar{E}' = -\bar{E}^{\text{inc}}$ and $\bar{H}' = -\bar{H}^{\text{inc}}$. As a result, they satisfy the boundary conditions. This fundamental equivalence underlies the formulation of surface IEs.

Volume Equivalence Principle: In electromagnetic problems involving inhomogeneous dielectric media, equivalent volume currents are introduced to account for the spatial variations in material properties. Starting from Maxwell's equations in a region with position-dependent permittivity $\varepsilon(\bar{r})$, the field behavior is governed by:

$$\nabla \times \bar{E}(\bar{r}) = -j\omega\mu_0\bar{H}(\bar{r}) - \bar{M}(\bar{r}), \quad (2.22)$$

$$\nabla \times \bar{H}(\bar{r}) = j\omega\varepsilon(\bar{r})\bar{E}(\bar{r}) + \bar{J}(\bar{r}), \quad (2.23)$$

where μ_0 is the permeability of free space/background. To reformulate the problem in terms of free-space Green's functions and simplify the use of IEs, an equivalent polarization (or contrast) current density is defined as:

$$\bar{J}_{\text{eq}}(\bar{r}) = j\omega[\varepsilon(\bar{r}) - \varepsilon_0]\bar{E}(\bar{r}), \quad (2.24)$$

where ε_0 denotes the permittivity of free space/background.

2.1.3 Electric Field Integral Equation (EFIE) for PEC Surfaces

Surface IEs offer an efficient reduction of Maxwell's equations for perfect electric conductor (PEC) objects by converting a volumetric problem into a surface-only formulation. In a PEC object placed on a simple medium (with parameters: ε, μ, k) scenario, the total tangential electric field must vanish on the boundary surface S of the object:

$$\hat{n} \times \bar{E}^{\text{total}}(\bar{r}) = \bar{0}, \quad \bar{r} \in S. \quad (2.25)$$

This boundary condition allows representing the PEC object via an equivalent electric surface current density $\bar{J}(\bar{r})$, as magnetic surface currents are 0 due to a null tangential $\bar{E}$ -field on S .

The scattered electric field by this equivalent surface current is given by:

$$\bar{E}^{\text{scat}}(\bar{r}) = -j\omega\mu \int_S \left[\bar{J}(\bar{r}') + \frac{1}{k^2} \nabla' \cdot \bar{J}(\bar{r}') \nabla \right] g(\bar{r}, \bar{r}') d\bar{r}'. \quad (2.26)$$

The total field is composed of the incident and scattered components:

$$\bar{E}^{\text{total}}(\bar{r}) = \bar{E}^{\text{inc}}(\bar{r}) + \bar{E}^{\text{scat}}(\bar{r}). \quad (2.27)$$

Applying the boundary condition yields:

$$-\hat{n} \times \bar{E}^{\text{scat}}(\bar{r}) = \hat{n} \times \bar{E}^{\text{inc}}(\bar{r}), \quad \bar{r} \in S. \quad (2.28)$$

This is known as the *normal Electric Field Integral Equation* (N-EFIE).

Alternatively, enforcing the condition via a tangential projection leads to the *tangential Electric Field Integral Equation* (T-EFIE):

$$-\hat{n} \times \hat{n} \times \bar{E}^{\text{scat}}(\bar{r}) = \hat{n} \times \hat{n} \times \bar{E}^{\text{inc}}(\bar{r}). \quad (2.29)$$

In this thesis, the T-EFIE is used as the primary surface integral formulation, as it is compatible with the widely adopted Rao-Wilton-Glisson (RWG) basis functions and facilitates accurate and stable numerical solutions [6].

2.1.4 Volume Electric Field Integral Equation (VEFIE)

Volume integral formulations are crucial for modeling electromagnetic scattering from inhomogeneous dielectric objects. These formulations retain the original geometry and utilize volume currents to express field interactions within the object.

Let the object occupy volume V with spatially varying permittivity $\varepsilon(\bar{r})$, embedded in a homogeneous background with parameters ε_0 and μ_0 . The incident electric field induces a contrast (or polarization) current:

$$\bar{J}_{eq}(\bar{r}) = j\omega[\varepsilon(\bar{r}) - \varepsilon_0]\bar{E}^{\text{total}}(\bar{r}), \quad (2.30)$$

$$= j\omega \frac{\varepsilon(\bar{r}) - \varepsilon_0}{\varepsilon(\bar{r})} \bar{D}^{\text{total}}(\bar{r}), \quad (2.31)$$

$$= j\omega\kappa(\bar{r})\bar{D}^{\text{total}}(\bar{r}) \quad (2.32)$$

where κ is the permittivity contrast function defined as

$$\kappa(\bar{r}) = 1 - \frac{\varepsilon_0}{\varepsilon(\bar{r})}. \quad (2.33)$$

Substituting the equivalent current defined in (2.32) into the inhomogeneous Maxwell equation (2.23), and expressing the total fields as the sum of incident and scattered components, the curl equation becomes

$$\nabla \times (\bar{H}^{\text{inc}}(\bar{r}) + \bar{H}^{\text{scat}}(\bar{r})) = j\omega\varepsilon_0 (\bar{E}^{\text{inc}}(\bar{r}) + \bar{E}^{\text{scat}}(\bar{r})) + \bar{J}(\bar{r}) + \bar{J}_{eq}(\bar{r}), \quad (2.34)$$

$$\nabla \times \bar{H}^{\text{inc}}(\bar{r}) = j\omega\varepsilon_0 \bar{E}^{\text{inc}}(\bar{r}) + \bar{J}(\bar{r}), \quad (2.35)$$

$$\nabla \times \bar{H}^{\text{scat}}(\bar{r}) = j\omega\varepsilon_0 \bar{E}^{\text{scat}}(\bar{r}) + \bar{J}_{eq}(\bar{r}), \quad (2.36)$$

where the incident fields $\bar{E}^{\text{inc}}$, $\bar{H}^{\text{inc}}$ are generated by the impressed sources $\bar{J}$ and $\bar{M}$, and the scattered fields $\bar{E}^{\text{scat}}$, $\bar{H}^{\text{scat}}$ are generated by the equivalent source $\bar{J}_{eq}$. A similar procedure can be applied to the Maxwell equation (2.22), yielding the magnetic field counterparts of (2.35) and (2.36). This reformulation effectively recasts the original inhomogeneous medium problem as an equivalent homogeneous background (characterized by ε_0 and μ_0) with embedded equivalent sources. The resulting equivalent problem, where $\bar{J}_{eq}$ radiates in background, produces the same scattered fields $\bar{E}^{\text{scat}}$ and $\bar{H}^{\text{scat}}$ as the original inhomogeneous

configuration. Thus, the scattered $\bar{E}$ -field from these volume currents is:

$$\bar{E}^{\text{scat}}(\bar{r}) = -j\omega\mu_0 \int_V \left[\bar{J}_{eq}(\bar{r}')g(\bar{r}, \bar{r}') - \frac{1}{k_0^2} \nabla [\bar{J}_{eq}(\bar{r}') \cdot \nabla' g(\bar{r}, \bar{r}')] \right] d\bar{r}'. \quad (2.37)$$

Then, the total field inside the object becomes:

$$\bar{E}^{\text{total}}(\bar{r}) = \bar{E}^{\text{inc}}(\bar{r}) + \bar{E}^{\text{scat}}(\bar{r}), \quad \bar{r} \in V. \quad (2.38)$$

Combining the equations above, we obtain the VEFIE:

$$\bar{E}^{\text{inc}} = \bar{E}^{\text{total}}(\bar{r}) + j\omega\mu_0 \int_V \left[\bar{J}_{eq}(\bar{r}')g(\bar{r}, \bar{r}') - \frac{1}{k_0^2} \nabla [\bar{J}_{eq}(\bar{r}') \cdot \nabla' g(\bar{r}, \bar{r}')] \right] d\bar{r}'. \quad (2.39)$$

Furthermore, we can express all parameters in terms of $\bar{D}$ -fields as

$$\bar{E}^{\text{inc}} = \frac{\bar{D}^{\text{total}}(\bar{r})}{\varepsilon(\bar{r})} + j\omega\mu_0 \int_V \left[\bar{J}_{eq}(\bar{r}')g(\bar{r}, \bar{r}') - \frac{1}{k_0^2} \nabla [\bar{J}_{eq}(\bar{r}') \cdot \nabla' g(\bar{r}, \bar{r}')] \right] d\bar{r}'. \quad (2.40)$$

This equation is known as the $\bar{D}$ -type formulation of volume integral equation (VIE). In this thesis, all VIE problems are solved using this approach.

2.2 Method of Moments (MoM)

The Method of Moments (MoM) is a widely used numerical technique for solving IEs that arise in computational electromagnetics. It transforms a continuous operator equation into a finite system of linear equations, which can be solved using standard linear algebra techniques.

Consider a linear operator equation of the form:

$$\mathcal{L}\{\bar{f}(\bar{r})\} = \bar{g}(\bar{r}), \quad (2.41)$$

where $\mathcal{L}\{\cdot\}$ is a linear operator acting on an unknown vector function $\bar{f}(\bar{r})$, and $\bar{g}(\bar{r})$ is a known excitation function (e.g., the incident electric field).

To discretize this equation, the unknown function is approximated by a finite series expansion using N known vector basis functions $\{\bar{b}_n(\bar{r})\}$:

$$\bar{f}(\bar{r}) \approx \sum_{n=1}^N \alpha_n \bar{b}_n(\bar{r}), \quad (2.42)$$

where α_n are unknown scalar coefficients to be determined.

Substituting this expansion into the operator equation, we obtain:

$$\sum_{n=1}^N \alpha_n \mathcal{L}\{\bar{b}_n(\bar{r})\} = \bar{g}(\bar{r}). \quad (2.43)$$

To solve for the coefficients α_n , we test both sides of the equation with a set of linearly independent testing functions $\{\bar{t}_m(\bar{r})\}$, yielding the following system through the method of weighted residuals:

$$\int \bar{t}_m(\bar{r}) \cdot \left(\sum_{n=1}^N \alpha_n \mathcal{L}\{\bar{b}_n(\bar{r})\} \right) d\bar{r} = \int \bar{t}_m(\bar{r}) \cdot \bar{g}(\bar{r}) d\bar{r}, \quad m = 1, 2, \dots, N. \quad (2.44)$$

Rearranging the sum and integral, we obtain the matrix form:

$$\sum_{n=1}^N \alpha_n Z_{mn} = w_m, \quad m = 1, 2, \dots, N, \quad (2.45)$$

with matrix entries and excitation vector defined as:

$$Z_{mn} = \int \bar{t}_m(\bar{r}) \cdot \mathcal{L}\{\bar{b}_n(\bar{r})\} d\bar{r}, \quad (2.46)$$

$$w_m = \int \bar{t}_m(\bar{r}) \cdot \bar{g}(\bar{r}) d\bar{r}. \quad (2.47)$$

This discretization leads to a dense linear system of the form $\bar{\bar{Z}}\bar{\alpha} = \bar{w}$, where $\bar{\bar{Z}} \in \mathbb{C}^{N \times N}$ is the impedance matrix, $\bar{\alpha}$ is the vector of unknown coefficients α_n , and $\bar{w}$ is the excitation vector. Solving this system directly via Gaussian elimination or LU decomposition leads to a computational cost of $\mathcal{O}(N^3)$ and a memory requirement of $\mathcal{O}(N^2)$, which becomes prohibitive for large N . Alternatively, iterative solvers such as generalized minimal residual method (GMRES) can be employed, offering a per-iteration cost of $\mathcal{O}(N^2)$ in both time and memory.

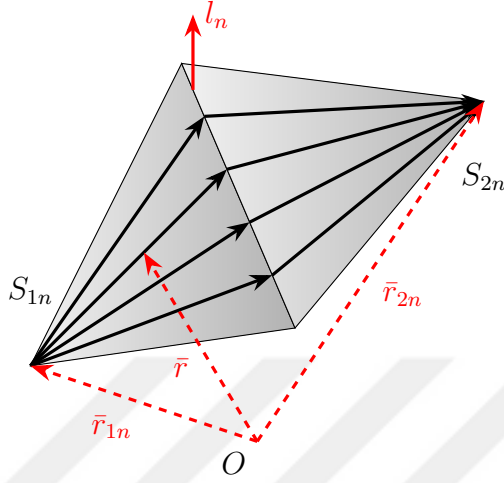


Figure 2.2: An RWG function.

These iterative methods are particularly advantageous when combined with fast algorithms or preconditioners that reduce the number of required iterations.

Before the discretization of the IEs, the geometry of the problem must first be meshed. In surface IEs, this is typically done using triangular surface elements, while volume integral formulations require a tetrahedral volume discretization. This is because triangles and tetrahedra are the optimal shapes that can accurately approximate arbitrary surfaces and volumes, respectively, with minimal geometric error. Once the domain is meshed, the IEs are discretized by expressing the unknowns using appropriate basis functions and enforcing the equations with corresponding testing functions. This results in the final matrix system being suitable for numerical solutions.

For surface IE problems, the Rao-Wilton-Glisson (RWG) basis functions are commonly used due to their divergence-conforming properties and compatibility with triangular surface meshes. As shown in Fig. 2.2, each RWG basis function is defined on a pair of adjacent triangles S_{1n} and S_{2n} sharing a common edge of length l_n . The function is expressed as:

$$\bar{b}_n(\bar{r}) = \begin{cases} \frac{l_n}{2A_{1n}}(\bar{r} - \bar{r}_{1n}), & \bar{r} \in S_{1n}, \\ \frac{l_n}{2A_{2n}}(\bar{r}_{2n} - \bar{r}), & \bar{r} \in S_{2n}, \\ 0, & \text{otherwise,} \end{cases} \quad (2.48)$$

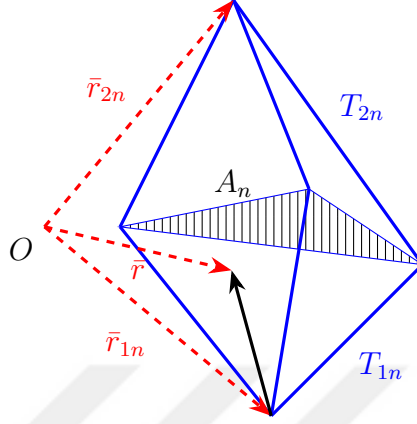


Figure 2.3: An SWG function.

where A_{1n} and A_{2n} are the areas of S_{1n} and S_{2n} , and $\bar{r}_{1n}$ and $\bar{r}_{2n}$ are the position vectors of the vertices opposite to the shared edge.

The divergence of each RWG function is piecewise constant and satisfies:

$$\nabla \cdot \bar{b}_n(\bar{r}) = \begin{cases} \frac{l_n}{A_{1n}}, & \bar{r} \in S_{1n}, \\ -\frac{l_n}{A_{2n}}, & \bar{r} \in S_{2n}, \\ 0, & \text{otherwise.} \end{cases} \quad (2.49)$$

RWG functions ensure local charge conservation and continuity of normal components across triangle edges, making them ideal for EFIE discretization.

In volumetric formulations, Schaubert-Wilton-Glisson (SWG) functions are employed to represent $\bar{D}$ -fields in tetrahedral volume meshes. As shown in Fig. 2.3, each SWG basis function is associated with an internal surface (with area of A_n) shared by two tetrahedra T_{1n} and T_{2n} . The function is expressed as:

$$\bar{b}_n(\bar{r}) = \begin{cases} \frac{A_n}{3V_{1n}}(\bar{r} - \bar{r}_{1n}), & \bar{r} \in T_{1n}, \\ \frac{A_n}{3V_{2n}}(\bar{r}_{2n} - \bar{r}), & \bar{r} \in T_{2n}, \\ 0, & \text{otherwise.} \end{cases} \quad (2.50)$$

where V_{1n} and V_{2n} are the volumes of T_{1n} and T_{2n} , and $\bar{r}_{1n}$ and $\bar{r}_{2n}$ are the position vectors of the vertices opposite to the shared surface.

The divergence of each SWG function is piecewise constant and satisfies:

$$\bar{b}_n(\bar{r}) = \begin{cases} \frac{A_n}{V_{1n}}, & \bar{r} \in T_{1n}, \\ -\frac{A_n}{V_{2n}}, & \bar{r} \in T_{2n}, \\ 0, & \text{otherwise,} \end{cases} \quad (2.51)$$

SWG functions are divergence-conforming, which means their divergence is constant within each tetrahedron, preserving Gauss's law. These features make SWG functions well-suited for discretizing volume IEs ($\bar{D}$ -type) in inhomogeneous dielectric media.

2.2.1 Discretization of IEs

For the surface IEs with PEC object under T-EFIE formulation, approximated unknown function $\bar{f}$ in (2.42) is $\bar{J}$ such that we expand it as $\bar{J}(\bar{r}) \approx \sum_{n=1}^N \alpha_n \bar{b}_n(\bar{r})$, where $\bar{b}_n$'s are RWG functions. Then, the Z_{mn} entries given in (2.46) are

$$Z_{mn} = -j\omega\mu \int \int \bar{t}_m(\bar{r}) \cdot \left[\bar{b}_n(\bar{r}') + \frac{1}{k^2} \nabla' \cdot \bar{b}_n(\bar{r}') \nabla \right] g(\bar{r}, \bar{r}') d\bar{r}' d\bar{r}, \quad (2.52)$$

and the elements w_m are written as

$$w_m = - \int \bar{t}_m(\bar{r}) \cdot \bar{E}^{\text{inc}}(\bar{r}) d\bar{r}. \quad (2.53)$$

For the volume IEs with $\bar{D}$ -type formulation, the approximated function $\bar{f}$ is $-j\omega\bar{D}$ such that we expand it as $-j\omega\bar{D}(\bar{r}) \approx \sum_{n=1}^N \alpha_n \bar{b}_n(\bar{r})$, where $\bar{b}_n$'s are SWG functions. Then, the Z_{mn} entries given in (2.46) are

$$\begin{aligned} Z_{mn} = & -\frac{1}{j\omega\varepsilon_0} \int (1 - \kappa(\bar{r})) \bar{t}_m(\bar{r}) \cdot \bar{b}_n(\bar{r}) d\bar{r} \\ & - j\omega\mu_0 \int \int \kappa(\bar{r}') \bar{t}_m(\bar{r}) \cdot \bar{b}_n(\bar{r}') g(\bar{r}, \bar{r}') d\bar{r}' d\bar{r} \\ & - \frac{1}{j\omega\varepsilon_0} \int \int \bar{t}_m(\bar{r}) \cdot \nabla [\kappa(\bar{r}') \bar{b}_n(\bar{r}') \cdot \nabla' g(\bar{r}, \bar{r}')] d\bar{r}' d\bar{r} \end{aligned} \quad (2.54)$$

and the elements w_m are written as

$$w_m = \int \bar{t}_m(\bar{r}) \cdot \bar{E}^{\text{inc}}(\bar{r}) d\bar{r}. \quad (2.55)$$

In each integral, since the integration domain is bounded due to the subdomain basis function choice, Gaussian quadrature for numerical integration is used.

Chapter 3

A Novel Highly Parallelizable ML-Based Method for the Fast Solution of IEs

In this chapter, a novel, broadband, and highly parallelizable method is presented for the fast and accurate iterative solution of frequency-domain IEs. The core of this method is an ML-based, group-by-group interaction framework that accelerates the evaluation of far-zone interactions by employing a one-box buffer scheme during MVMs. The approach begins by partitioning the scatterer geometry into uniformly sized boxes, following a strategy similar to that of MLFMA. Within each box, the basis functions are represented by a fixed set of uniform basis functions. Far-field interactions between these functions are then evaluated in a group-wise manner using ML models. As a result, strong scalability for parallelization is achieved without encountering the LFB problem.

3.1 Formulation

As outlined in Section 2.2, the MoM approach transforms a volume or surface IE into the matrix equation as follows:

$$\bar{\bar{Z}}\bar{\alpha} = \bar{V}. \quad (3.1)$$

In (3.1), $\bar{\bar{Z}} \in \mathbb{C}^{N \times N}$ denotes the impedance matrix corresponding to N unknowns, and $\bar{V} \in \mathbb{C}^{N \times 1}$ represents the excitation vector. The vector $\bar{\alpha} \in \mathbb{C}^{N \times 1}$ holds the unknown coefficients of the subdomain basis functions, which are to be computed iteratively using the novel method introduced in this work.

In a manner analogous to FMM, the proposed method begins by enclosing the entire object of size D within a mother box (cube), which is then subdivided into 8^L equally sized child boxes, where L is a positive integer. Each child box has an edge length of a , making the total side length of the mother box equal to $a2^L$, as illustrated in Fig. 3.1. For any object, there exists some L such that the inequality $a2^{L-1} < D \leq a2^L$ holds. Next, applying the one-box-buffer scheme (see Fig. 3.1) and considering only non-empty boxes, the MVM operations are decomposed into near-field (NF) and far-field (FF) interactions as follows:

$$\bar{\bar{Z}}^{NF}\bar{\alpha} + \bar{\bar{Z}}^{FF}\bar{\alpha} = \bar{V}. \quad (3.2)$$

In (3.2), $\bar{\bar{Z}}^{NF}$ represents the near-field interaction matrix, which is computed directly and stored. On the other hand, $\bar{\bar{Z}}^{FF}$ corresponds to the far-field interaction matrix, and the product $\bar{\bar{Z}}^{FF}\bar{\alpha}$ is evaluated using the proposed group-by-group interaction computation scheme. This scheme consists of three main stages: *A*) mapping to uniform basis functions, *B*) translation, and *C*) inverse mapping to test (subdomain) functions. These steps are detailed in the subsections that follow.

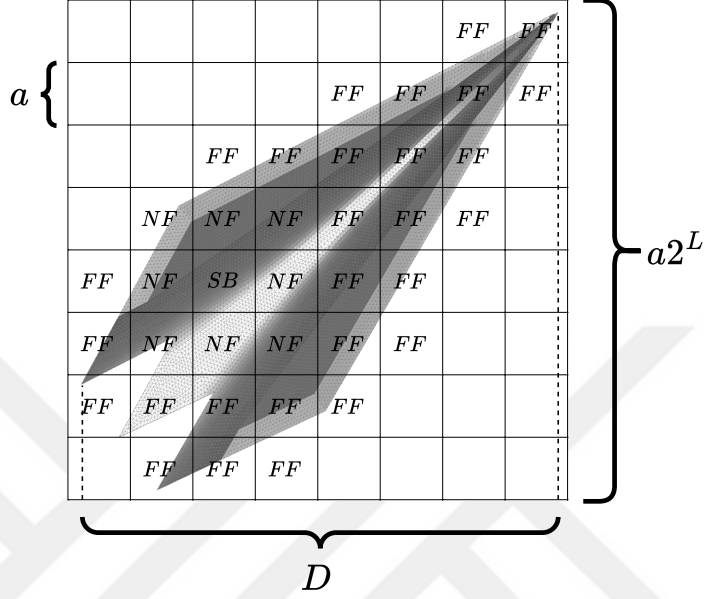


Figure 3.1: 2D projection of a sample scatterer inside a mother box, which is divided into child boxes. For the selected box SB , NF boxes correspond to near interactions, and FF boxes correspond to far interactions.

3.1.1 Mapping to Uniform Basis Functions

The objective of this stage is to express the subdomain basis functions, used to discretize an arbitrary scatterer with any material properties, within each box in terms of a fixed number of uniformly distributed and arbitrarily oriented basis functions, referred to as uniform basis functions. In this work, the chosen uniform basis functions are arbitrarily oriented Hertzian dipoles, as depicted in Fig. 3.2.

Let $\bar{J}(\bar{r})$ denote the current distribution within one of the child boxes, expressed as

$$\bar{J}(\bar{r}) = \sum_{n=1}^{N_b} \alpha_n \bar{b}_n(\bar{r}), \quad (3.3)$$

where $\bar{b}_n(\bar{r})$ denotes the original subdomain basis functions, α_n are their corresponding coefficients (updated at each iteration), and N_b is the total number of

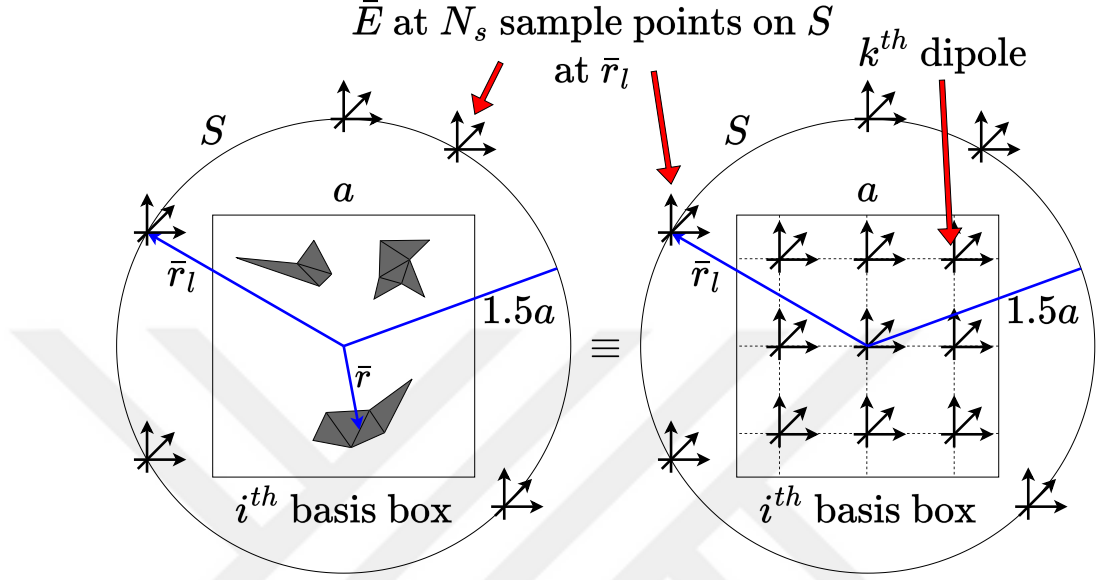


Figure 3.2: Mapping from subdomain basis functions to uniform basis functions

subdomain basis functions in the selected box. This current distribution generates the electric field

$$\bar{E}_o(\bar{r}) = \int \bar{\bar{D}}(\bar{r}, \bar{r}') \cdot \bar{J}(\bar{r}') d\bar{r}', \quad (3.4)$$

where $\bar{\bar{D}}(\bar{r}, \bar{r}')$ is the free-space Dyadic Green's function, defined as

$$\begin{aligned} \bar{\bar{D}}(\bar{r}, \bar{r}') = & -jk\eta \frac{\exp(-jkR)}{4\pi R} \left\{ \bar{\bar{I}}_{3 \times 3} \left(1 - \frac{j}{kR} - \frac{1}{k^2 R^2} \right) \right. \\ & \left. - \hat{R} \hat{R} \left(1 - \frac{3j}{kR} - \frac{3}{k^2 R^2} \right) \right\}. \end{aligned} \quad (3.5)$$

In (3.5), the vectors and parameters are defined as $\bar{R} = \bar{r} - \bar{r}'$, $R = \|\bar{R}\|$, $\hat{R} = \bar{R}/R$, with k and η denoting the free-space wave number and intrinsic impedance, respectively.

Similarly, let $\bar{J}_u(\bar{r})$ represent the current distribution of the uniform basis functions given as

$$\bar{J}_u(\bar{r}) = \sum_{\hat{a} \in \{\hat{x}, \hat{y}, \hat{z}\}} \sum_{k=1}^{N_d} \beta_{\hat{a},k} \bar{p}_{\hat{a},k}(\bar{r}). \quad (3.6)$$

Here, $\bar{p}_{\hat{a},k}$ denotes the position-fixed uniform basis function oriented along the direction of $\hat{a}$. That is, $\bar{p}_{\hat{x},k}$ points in the $\hat{x}$ direction, $\bar{p}_{\hat{y},k}$ in the $\hat{y}$ direction, and

$\bar{p}_{\hat{z},k}$ in the $\hat{z}$ direction. The associated unknown coefficient, $\beta_{\hat{a},k}$, is determined at the end of this stage in each iteration. N_d denotes the number of position-fixed uniform basis functions within each box. Since each basis function has three vector components (along $\hat{x}$, $\hat{y}$, and $\hat{z}$), the total number of uniform basis function components per box is given by $N_u = 3N_d$. The fields generated by $\bar{J}_u(\bar{r})$ can similarly be written as

$$\bar{E}_u(\bar{r}) = \sum_{\hat{a} \in \{\hat{x}, \hat{y}, \hat{z}\}} \sum_{k=1}^{N_d} \beta_{\hat{a},k} \int \bar{\bar{D}}(\bar{r}, \bar{r}') \cdot \bar{p}_{\hat{a},k}(\bar{r}') d\bar{r}'. \quad (3.7)$$

According to the uniqueness theorem, if two source distributions produce identical $\bar{E}$ -fields on a closed surface S , then the resulting $\bar{E}$ -fields must also be identical outside of S . Based on this principle, we aim to determine the optimal coefficients for the uniform basis functions, i.e., $\beta_{\hat{a},k}$ for $k = 1, \dots, N_d$, $\hat{a} \in \{\hat{x}, \hat{y}, \hat{z}\}$, by minimizing the $\bar{E}$ -field error on a chosen surface S . This surface is selected as a sphere of radius $1.5a$, as shown in Fig. 3.2. The reason for selecting this radius is to ensure that every far-zone interaction point lies outside the surface S , while still avoiding overlap with test boxes under the one-box buffer scheme. Therefore, the objective becomes finding the set of $\beta_{\hat{a},k}$ values that minimize the field mismatch on S :

$$\operatorname{argmin}_{\beta_{\hat{a},k}} \int_{\bar{r} \in S} \|\bar{E}_o(\bar{r}) - \bar{E}_u(\bar{r})\|^2 d\bar{r}, \quad (3.8)$$

which can be approximated by discretizing the surface using N_s sampling points on S , leading to the expression:

$$\operatorname{argmin}_{\beta_{\hat{a},k}} \sum_{l=1}^{N_s} \|\bar{E}_o(\bar{r}_l) - \bar{E}_u(\bar{r}_l)\|^2 = \operatorname{argmin}_{\beta_{\hat{a},k}} \|\bar{B} - \bar{\bar{A}}\bar{C}\|^2. \quad (3.9)$$

In (3.9), the vector $\bar{B} \in \mathbb{C}^{3N_s \times 1}$ contains the components of $\bar{E}_o$ at each sample point $\bar{r}_l \in S$; the matrix $\bar{\bar{A}} \in \mathbb{C}^{3N_s \times N_u}$ is constructed from 3×3 blocks $\bar{\bar{A}}_{l,k}$, which represent the free-space Dyadic Green's function $\bar{\bar{D}}(\bar{r}_l, \bar{r}_k)$ between the k^{th} Hertzian dipole and the l^{th} sample point; and $\bar{C} \in \mathbb{C}^{N_u \times 1}$ is the vector of unknown coefficients $\beta_{\hat{a},k}$. The least-squares solution to (3.9) is given by

$$\bar{C} = \bar{\bar{A}}^\dagger \bar{B} \quad (3.10)$$

where $\bar{\bar{A}}^\dagger$ denotes the pseudo-inverse of $\bar{\bar{A}}$.

Within each child box, a total of n_d Hertzian dipoles are uniformly placed along each of the x , y , and z directions, resulting in a total dipole count of $N_d = n_d^3$ per box. The selection of N_s and N_u is guided by a similar strategy used in MLFMA. Specifically, depending on the box size a (or its electrical size ka), N_d is chosen based on the following rule:

$$N_d = \begin{cases} 2(14.14d_0 - 7.17)^2 & \text{if } a < \lambda/8; [43] \\ 2(1.73ka + 2.16(d_0)^{\frac{2}{3}}(ka)^{\frac{1}{3}})^2 & \text{else; [44].} \end{cases} \quad (3.11)$$

In (3.11), d_0 specifies the desired number of accurate digits in the Green's function evaluation. The number of sampling points N_s is then empirically set to twice the value of N_d , providing a favorable balance between computational cost and accuracy. For the sampling scheme on surface S , spherical Fibonacci lattice points [45] are employed due to their near-uniform angular distribution. In standard spherical coordinates (r, θ, ϕ) , the Fibonacci lattice defines the sampling points as $r_l = (1.5a, \theta_l, \phi_l)$ with $\theta_l = \arccos(1 - (2l + 1)/N_s)$ and $\phi_l = 2\pi l/\varphi$, where φ is the golden ratio.¹

The pseudocode presented in Algorithm 1 is used to implement the mapping stage, including its preprocessing step, during MVMs. In this algorithm, each $\bar{C}'_n$ for $n = 1, \dots, N$ corresponds to the uniform basis function coefficients associated with the n^{th} subdomain basis function of unit amplitude. These vectors are computed in the preprocessing step using (3.10) and stored. Then, during the actual MVM operation, the coefficient vector $\bar{C}$ for a given box is obtained as the linear superposition of all $\bar{C}'_n$ vectors corresponding to the subdomain basis functions within that box. This strategy is valid for any child box due to the identity

$$\bar{C} = \bar{A}^\dagger \bar{B} = \bar{A}^\dagger \left(\sum_n \alpha_n \bar{B}_n \right) = \sum_n \alpha_n \bar{A}^\dagger \bar{B}_n = \sum_n \alpha_n \bar{C}'_n. \quad (3.12)$$

In (3.12), each $\bar{B}_n$ denotes the electric field at the sample points on S due to the n^{th} subdomain basis function with unit amplitude, and the summation runs over all subdomain basis functions inside the given child box. This formulation enables an efficient computation of the mapping step.

¹The golden ratio is defined as $\varphi = 0.5 + 0.5\sqrt{5}$.

Algorithm 1 Mapping to Uniform Basis Functions

Preprocessing Stage:Compute $\bar{\bar{A}}^\dagger$ in (3.10)**for** $i \in \text{Non-empty Child Box Indices}$ **do** **for** $n \in \text{Subdomain Functions in Box } i$ **do** Compute $\bar{B}_n$ for n^{th} subdomain function. $\bar{C}'_n = \bar{\bar{A}}^\dagger \bar{B}_n$ (via (3.10)). **end for****end for****return and store** $\bar{C}'_n$'s

During MVM Computation for each Iteration:**for** $i \in \text{Non-empty Child Box Indices}$ **do** Initialize $\bar{C}_i = 0$ for the i^{th} box **for** $n \in \text{Subdomain Functions in Box } i$ **do** $\bar{C}_i = \bar{C}_i + \alpha_n \bar{C}'_n$ (via (3.12)). **end for****end for****return** $\bar{C}_i$'s

3.1.2 Translation

After completing the first stage, all boxes are rendered structurally identical, each containing the same number of Hertzian dipoles positioned at fixed locations. Now, consider two boxes that lie in each other's far-zone, as illustrated in Fig. 3.3. Let $\bar{r}_i$ and $\bar{r}_j$ denote the centers of the i^{th} basis box and the j^{th} test box, respectively. The translation vector from the basis box to the test box is given by $\bar{w}_{ij} = \bar{r}_j - \bar{r}_i$. These translation vectors $\bar{w}$ take the form $a(\hat{x}n_1 + \hat{y}n_2 + \hat{z}n_3)$, which is compactly represented as $[n_1, n_2, n_3]^T$, where n_1 , n_2 , and n_3 are integers in the range $(-2^L, 2^L)$.

Also in Fig. 3.3, let $\bar{r}_{dk}$ denote the position of the k^{th} dipole relative to the

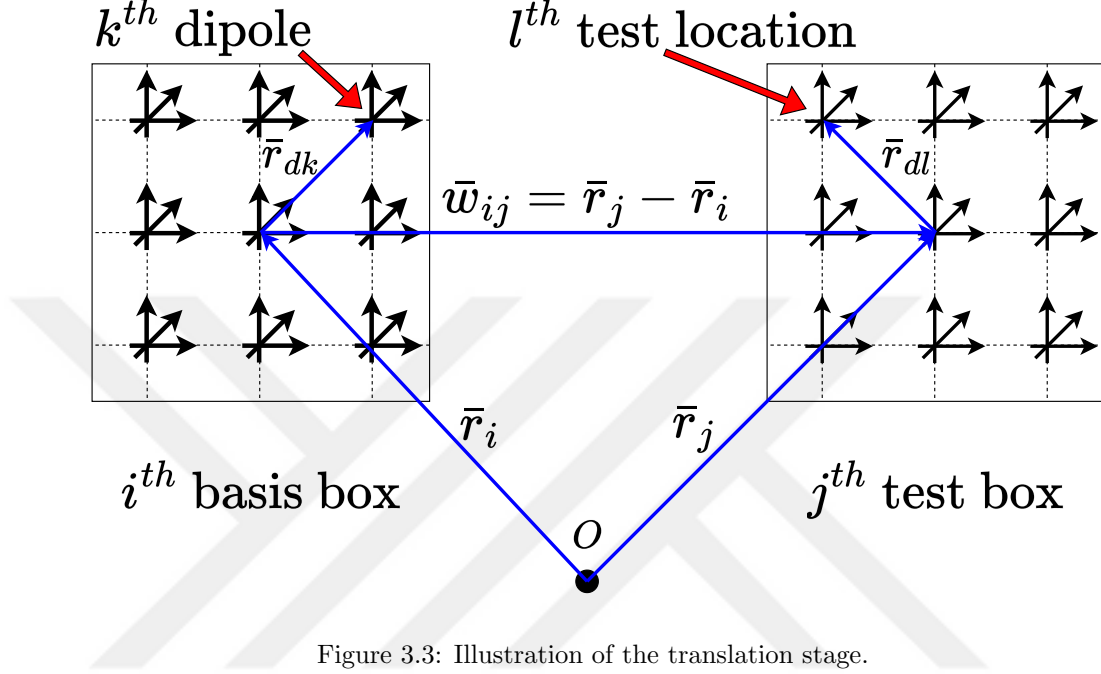


Figure 3.3: Illustration of the translation stage.

center of its own box (the i^{th} basis box), and let $\bar{r}_{dl}$ denote the position of the l^{th} dipole relative to the center of its box (the j^{th} test box). The field observed at the l^{th} dipole location in the test box, due to the k^{th} dipole in the basis box, is given by

$$\bar{E}(\bar{r}_{dl}) = \bar{\bar{D}}(\bar{w}_{ij} + \bar{r}_{dl}, \bar{r}_{dk}) \cdot \bar{p}_{a,k}. \quad (3.13)$$

By generalizing (3.13) to include all dipoles and test locations in the i^{th} basis box and the j^{th} test box, we obtain the following matrix equation:

$$\bar{E}_{i-j}(\bar{r}_{dl}) = \bar{\bar{G}}_{\bar{w}_{ij}}(\bar{w}_{ij} + \bar{r}_{dl}, \bar{r}_{dk}) \cdot \bar{C}_i(\bar{r}_{dk}), \quad (3.14)$$

where $\bar{\bar{G}}_{\bar{w}_{ij}} \in \mathbb{C}^{3N_d \times 3N_d}$ is the translation matrix corresponding to the vector $\bar{w}_{ij}$. Each column of $\bar{\bar{G}}_{\bar{w}_{ij}}$ represents the electric field at all test locations in the j^{th} test box due to a unit dipole component located in the i^{th} basis box. The vector $\bar{C}_i \in \mathbb{C}^{3N_d \times 1}$ contains the uniform basis function coefficients computed in Algorithm 1 via (3.10), and $\bar{E}_{i-j} \in \mathbb{C}^{3N_d \times 1}$ represents the resulting field vector at the test dipole locations. The total far-zone field at the test locations of the j^{th} test box, denoted by $\bar{E}_j$, is then obtained by summing the contributions from all

far-zone boxes as

$$\bar{E}_j = \sum_{i \in \mathbb{F}_j} \bar{E}_{i-j}, \quad (3.15)$$

where $\mathbb{F}_j$ denotes the set of indices corresponding to boxes in the far-zone of the j^{th} test box.

Since all boxes have identical structure, many translation matrices $\bar{\bar{G}}_{\bar{w}}$ are related by geometric symmetries. Specifically, consider two translation matrices $\bar{\bar{G}}_{\bar{w}_1}$ and $\bar{\bar{G}}_{\bar{w}_2}$ with associated vectors $\bar{w}_1$ and $\bar{w}_2$. If $\bar{w}_2$ is obtained from $\bar{w}_1$ through 90° rotations about the x -, y -, or z -axes, or reflections across these axes, then the matrices are related by

$$\bar{\bar{G}}_{\bar{w}_2} = \bar{\bar{P}}^T \bar{\bar{\Sigma}} \bar{\bar{G}}_{\bar{w}_1} \bar{\bar{\Sigma}} \bar{\bar{P}}, \quad (3.16)$$

where $\bar{\bar{P}}$ is a permutation matrix and $\bar{\bar{\Sigma}}$ is a diagonal matrix with entries in $\{-1, 1\}$. Under these transformations, the relative grid of dipole and test locations remains unchanged, but the indexing is permuted (reflected in $\bar{\bar{P}}$), and the field and dipole directions are appropriately rotated (reflected in $\bar{\bar{\Sigma}}$ and $\bar{\bar{P}}$). As a result, it suffices to compute and store only a small subset of translation matrices. For example, when $L = 3$, the one-box-buffer scheme results in 316 distinct translation vectors, but only 16 unique translation matrices need to be computed directly; the others are generated by applying symmetry transformations. This approach to modeling far-zone interactions using $\bar{\bar{G}}_{\bar{w}}$ matrices is referred to as the direct Green's function approach (DGFA) in this work. DGFA provides high accuracy because it avoids both the truncated multipole expansions and the series truncation used in MLFMA. However, using DGFA in a dipole-wise fashion results in $\mathcal{O}(N_d^2)$ complexity for a single box-to-box interaction. To mitigate this cost, we employ ML models to accurately emulate the translation matrices, allowing efficient prediction of dipole-to-dipole interactions across far-zone boxes in a group-wise manner. This novel approach enhances efficiency significantly. The complete procedure for executing the translation stage at each iteration using DGFA is summarized in Algorithm 2. Details on the ML implementation that replaces the DGFA evaluation in (3.14) are provided in Section 3.2.

Algorithm 2 Translation using DGFA

```
Initialize  $\bar{E}_j = 0$  for  $j = 1, \dots, N_B$  ( $N_B$  : # of Boxes)
for  $i \in$  Non-empty Child Box Indices do
  for  $j \in$  Non-empty Child Box Indices do
    if  $\|\bar{r}_j - \bar{r}_i\| = \|\bar{w}_{ij}\| \geq 2a$  then
      Calculate  $\bar{E}_{i-j}$  defined in (3.14)
       $\bar{E}_j = \bar{E}_j + \bar{E}_{i-j}$ .
    end if
  end for
end for
return  $\bar{E}_j$ 's
```

3.1.3 Inverse Mapping to Subdomain Functions

At the end of the translation stage, the fields at arbitrary test locations can be obtained using (3.15). These test points may correspond to the interaction points associated with the original subdomain test functions. However, to ensure a formulation that remains independent of the geometry of the scatterer, we adopt a mapping approach analogous to that used in the first stage. To this end, we consider the j^{th} test box and distribute N_s Hertzian dipoles, referred to as virtual dipoles, on a spherical surface S at locations given by the spherical Fibonacci lattice points, previously defined as $\bar{r}_l$, and illustrated in Fig. 3.4.

For the j^{th} test box, the fields at internal test locations $\bar{r}_{dl}$, induced by its far-zone interacting boxes, can be expressed as

$$\bar{E}_{vd}(\bar{r}_{dl}) = \sum_{\hat{a} \in \{\hat{x}, \hat{y}, \hat{z}\}} \sum_{l=1}^{N_d} \bar{D}(\bar{r}_{dl}, \bar{r}_l) \cdot q_{\hat{a},l} \bar{p}_{\hat{a},l}(\bar{r}_l), \quad (3.17)$$

where $\bar{p}_{\hat{a},l}$ denotes the virtual dipole located at $\bar{r}_l$ on S , and $q_{\hat{a},l}$ is its unknown excitation coefficient to be determined. Meanwhile, the total field at these test locations is already known from the translation stage, as described by (3.15), and denoted by $\bar{E}_j(\bar{r}_{dl})$.

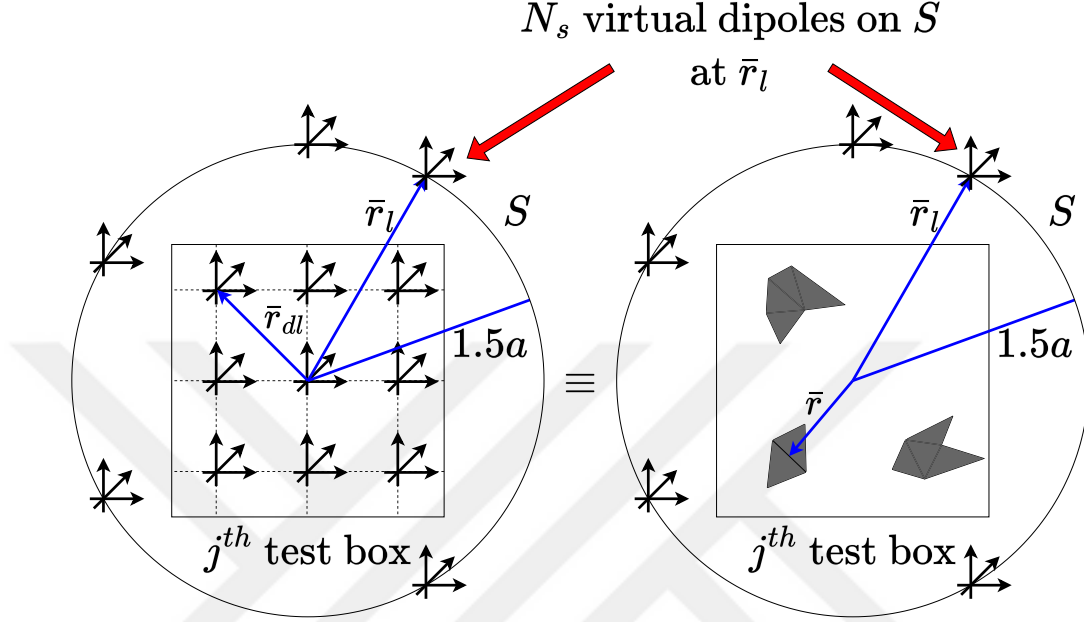


Figure 3.4: Inverse mapping from uniform basis functions to subdomain functions

Determining the optimal set of $q_{\hat{a},l}$ coefficients becomes a least-squares problem, analogous to the one in the first stage, and is formulated as

$$\operatorname{argmin}_{q_{\hat{a},l}} \sum_{dl=1}^{N_d} |\bar{E}_j(\bar{r}_{dl}) - \bar{E}_{vd}(\bar{r}_{dl})|^2 = \operatorname{argmin}_{q_{\hat{a},l}} \|\bar{E}_j - \bar{A}'\bar{Q}\|^2. \quad (3.18)$$

In (3.18), $\bar{A}' = \bar{A}^T$ is the transpose of the same matrix defined in (3.9), and $\bar{Q} \in \mathbb{C}^{3N_s \times 1}$ is the vector of unknown virtual dipole coefficients. The least-squares solution for $\bar{Q}$ is given by

$$\bar{Q} = (\bar{A}^T)^\dagger \bar{E}_j. \quad (3.19)$$

Once $\bar{Q}$ is computed, the resulting $\bar{E}$ -fields can be evaluated anywhere within the j^{th} test box using (3.17). To complete the far-zone interaction term $\bar{Z}^{FF} \bar{\alpha}$ in (3.2), we proceed as follows. Let $\bar{D}_{jn} \in \mathbb{C}^{3N_s \times 1}$ be the vector resulting from testing the expression in (3.17) at the integration points of the n^{th} subdomain function within the j^{th} test box (where $\bar{r}_{dl}$ becomes $\bar{r}$, as illustrated on the right of Fig. 3.4). Using (3.19), we obtain

$$(\bar{Z}^{FF} \bar{\alpha})_n = \bar{D}_{jn}^T \bar{Q} = \bar{D}_{jn}^T (\bar{A}^T)^\dagger \bar{E}_j = (\bar{D}'_n)^T \bar{E}_j. \quad (3.20)$$

The transpose vectors $(\bar{D}'_n)^T$ are precomputed and stored during the preprocessing stage, analogous to the $\bar{C}'_n$ vectors used in the first stage. Algorithm 3 outlines

the full implementation of this mapping stage.

Algorithm 3 Inverse Mapping to Subdomain Functions

Preprocessing Stage:

Compute $(\bar{\bar{A}}^T)^\dagger$ in (3.19)

for $j \in \text{Non-empty Child Box Indices}$ **do**

for $n \in \text{Subdomain Functions in Box } j$ **do**

 Compute $\bar{D}_{jn}$ for n^{th} subdomain function.

$(\bar{D}'_n)^T = \bar{D}_{jn}^T (\bar{\bar{A}}^T)^\dagger$ (via (3.20)).

end for

end for

return and store $\bar{D}'_n$'s

During MVM Computation for each Iteration:

Initialize $\bar{\bar{Z}}^{FF} \bar{\alpha} = \bar{0}$

for $j \in \text{Non-empty Child Box Indices}$ **do**

for $n \in \text{Subdomain Functions in Box } j$ **do**

$(\bar{\bar{Z}}^{FF} \bar{\alpha})_n = (\bar{D}'_n)^T \bar{E}_j$ (via (3.20)).

end for

end for

return $\bar{\bar{Z}}^{FF} \bar{\alpha}$

3.1.4 Discussions on the Merits of the Proposed Approach

- The proposed method exhibits strong scalability for parallelization, as each group-by-group interaction corresponds to a separate portion of the impedance matrix $\bar{\bar{Z}}$. This design eliminates the need for inter-node communication during parallel execution. Furthermore, since all boxes are structurally identical after the mapping stage, the computational workload is evenly distributed across processing nodes. With the mapping vectors (i.e., $\bar{C}'_n$ and $\bar{D}'_n$) computed and stored in the preprocessing stage, parallelization can focus solely on evaluating group-by-group interactions.
- The proposed framework inherently avoids the LFB issue faced by MLFMA,

irrespective of whether DGFA or ML models emulating DGFA are employed.

- Once the offline training phase is completed, the method becomes universally applicable to any IE formulation involving scattering from both conducting and material bodies, provided the geometry fits within the mother box. For larger domains, only additional translation networks must be trained. Furthermore, the approach may potentially enable fast solutions to coupled volume-surface IEs [46], which arise in the analysis of composite dielectric-conducting structures.
- The method is extendable to a multilevel framework, analogous to the hierarchical progression from FMM to MLFMA. A tree-based structure can be established in which group-by-group interactions are evaluated at each level, complemented by corresponding mapping stages. Nonetheless, the choice of uniform basis function locations should be revisited for optimal efficiency. While achieving strong scalability may be challenging in the multilevel extension, the proposed method is expected to offer better load balancing than MLFMA due to its uniform box structure at every level, although further investigation is needed to validate this claim.

3.2 ML Implementation

The translation stage constitutes the primary computational load in each MVM iteration. To address this efficiently, we utilize simple feed-forward complex-valued fully connected neural networks (CVFCNNs) [47] capable of accurately predicting $\bar{\bar{G}}_{\bar{w}_{ij}}$ in (3.14). Each CVFCNN used in the proposed method comprises a single hidden layer of size $\mathbb{C}^{H \times 1}$. It accepts the input $\bar{C}_i \in \mathbb{C}^{3N_d \times 1}$ (normalized using its ℓ_∞ norm) and produces the output $\bar{O}_{i-j} \in \mathbb{C}^{3N_d \times 1}$, which serves as an estimate of the electric field vector $\bar{E}_{i-j}$ defined in (3.14). As shown in earlier work [48], employing more complex architectures such as convolutional neural networks offers no tangible benefits and can even degrade the efficiency of the proposed scheme. All CVFCNNs are implemented using the complex-valued

neural network (CVNN) library built upon TensorFlow [49], with the Cartesian ReLU activation function employed in the hidden layer.

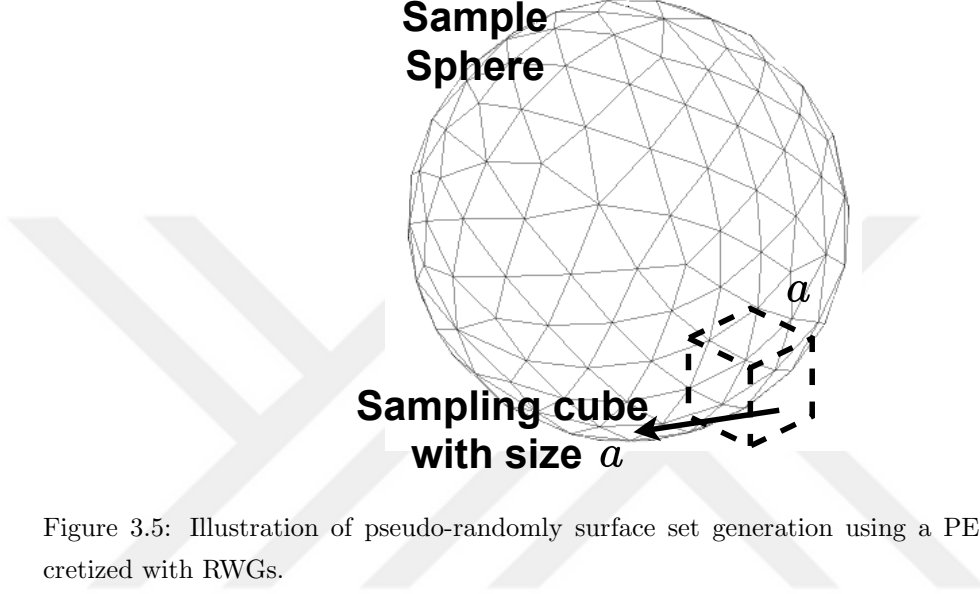


Figure 3.5: Illustration of pseudo-randomly surface set generation using a PEC sphere discretized with RWGs.

To improve training efficiency and accuracy, a two-stage training approach is adopted. In the first stage, training data are generated by randomly assigning real and imaginary parts of the dipole coefficients from a uniform distribution over $[-1, 1]$. This training phase continues until the relative ℓ_2 -norm error,

$$e = \frac{\|\bar{O}_{i-j} - \bar{E}_{i-j}\|_2}{\|\bar{E}_{i-j}\|_2}, \quad (3.21)$$

falls below 10^{-3} . Here, $\bar{O}_{i-j}$ denotes the neural network output, while $\bar{E}_{i-j}$ corresponds to the actual field computed via DGFA in (3.14), both evaluated over the test set generated from uniformly distributed samples.

Following the first stage, the networks are further trained using a more realistic dataset to improve generalization and reduce total training time. This second-stage training set is constructed from Hertzian dipole coefficients derived from pseudo-randomly generated PEC surfaces discretized using RWG basis functions, as illustrated in Fig. 3.5. A sampling box, sized identically to the child boxes, is swept across a PEC sphere with a step size of $0.5a$. At each sampling location,

Table 3.1: ML Model Parameters

ML Model Parameter Names	Values
Hidden Layer Dimension (H)	50
Input and Output Dimensions	192
Loss Function	Mean Square Error (MSE)
Optimizer	Adam
Epochs in the First Training Stage	~ 500
Epochs in the Second Training Stage	~ 600
1st Batch Size	200
2nd Batch Size	1000
3rd Batch Size	10000
4th Batch Size	100000

only boxes containing RWG functions are retained. For each retained box, complex coefficients (with real and imaginary parts drawn uniformly from $[-1, 1]$) are assigned to the RWG functions within it. These RWG coefficients are then mapped to their corresponding Hertzian dipole representations using the procedure outlined in Section 3.1. The resulting Hertzian dipole coefficients form the input samples for the second training and validation set. As in the first stage, training continues until the relative ℓ_2 -norm error defined in (3.21) falls below 10^{-3} . In this context, $\bar{O}_{i-j}$ now represents the predicted output for test data originating from the pseudo-random spherical surface distribution.

Table 3.2: Test Set Errors of Several Networks

Translation Distance & Input Box Length	Mean of Relative ℓ_2 Error of Output	Std. of Relative ℓ_2 Error of Output
$[0, 0, 3]^T, a = \lambda/8$	2.0677e-04	1.2839e-04
$[3, 3, 3]^T, a = \lambda/8$	2.8932e-04	2.2387e-04
$[0, 0, 2]^T, a = \lambda/256$	4.8532e-04	3.0749e-04
$[0, 2, 2]^T, a = \lambda/256$	1.1256e-04	6.2188e-05

In Table 3.1, the ML model parameters are given. The Adam optimizer [50] is employed throughout both training stages, along with the mean squared error as the loss function. For demonstration purposes, the CVFCNNs are trained

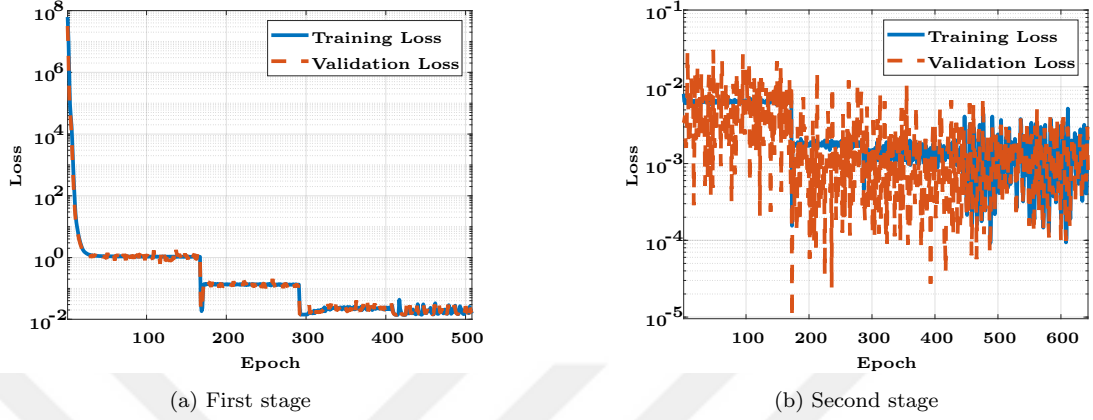


Figure 3.6: Training and validation losses at the first and second steps of the training for the $[3, 3, 3]^T$, $a = \lambda/8$ translation network.

for two different child box sizes: $a = \lambda/8$ for solving structures up to 2λ in extent, and $a = \lambda/256$ for structures up to $\lambda/16$, where conventional MLFMA often experiences the LFB problem. In both cases, the hidden layer width is set to 50, providing a good compromise between accuracy and computational efficiency. The input-output dimension of each network is $3N_d = 192$, given that $N_d = 64$ corresponding to a $4 \times 4 \times 4$ dipole grid in each child box. Additionally, the batch size is adaptively adjusted during training to further reduce error. Specifically, the batch size is increased in four steps: starting from 200, then increased to 1000, 10000, and finally 100000. This progression is illustrated in Fig. 3.6, where the noticeable drops in the loss curve correspond to the points of batch size increase. These changes improve the stability of gradient estimates and contribute to more consistent convergence. The first and second training stages typically require approximately 500 and 600 epochs, respectively. Fig. 3.6 shows the training and validation loss curves for $a = \lambda/8$ and a translation vector of $[3, 3, 3]^T$. While the first stage exhibits a smooth loss decrease, the second stage achieves significantly lower errors, albeit with mild oscillations, indicating that training with only uniformly distributed data is insufficient for high accuracy. The two-stage scheme thus enhances both efficiency and accuracy. Finally, Table 3.2 summarizes the relative ℓ_2 errors for several trained networks, confirming that all achieve error levels below 10^{-3} , a typical tolerance threshold in iterative solvers. These trained networks are used in the numerical experiments to solve a variety of scattering problems, including volumetric cases with inhomogeneous dielectric

materials.

3.3 Parallelization

As discussed in Section 3.1, once the mapping to uniform basis functions is complete, all boxes become structurally identical. This uniformity enables strong scalability for parallel computation, regardless of whether DGFA or ML networks are used for dipole-to-dipole interaction evaluations. In this section, we outline the parallelization strategy designed to distribute the computational workload evenly across multiple workers. Notably, all steps described here are part of the preprocessing phase.

Let $\mathbb{W}$ denote the set of distinct translation vectors, where any two vectors in the set are not related by rotational or mirror symmetries. Define $\mathbb{W}_r \subseteq \mathbb{W}$ as the subset of translation vectors not yet assigned to any worker. For each $\bar{w} \in \mathbb{W}$, let $n_{\bar{w}}$ represent the total number of box-to-box interactions associated with $\bar{w}$ or its symmetry-equivalent variants. Additionally, let t_m be the number of translation operations already allocated to worker m , out of a total of M_w workers. To achieve balanced load distribution and minimize memory usage, particularly relevant for DGFA or ML model storage, we implement the following three-step scheduling procedure:

1. Translation vectors with the fewest associated interactions (i.e., $\text{argmin}_{\bar{w} \in \mathbb{W}_r} n_{\bar{w}}$) are assigned to workers in a cyclic sequential order. Specifically, translations are distributed across workers 1 through M_w , looping back to the first worker after each full cycle. This continues until the total number of remaining unassigned interactions, denoted as $r = \sum_{\bar{w} \in \mathbb{W}_r} n_{\bar{w}}$, falls below the imbalance among workers:

$$r < \sum_{\tilde{m}} (\max\{t_m \mid \forall m\} - t_{\tilde{m}}). \quad (3.22)$$

If this condition is met, the step concludes before assigning the last translation vector. Importantly, since equivalent translations are kept exclusive to

specific workers, memory duplication (e.g., storing ML weights) is avoided.

2. The remaining unassigned vectors in $\mathbb{W}_r$ are sorted in ascending order of their interaction counts $n_{\bar{w}}$ and distributed to workers. Starting from the worker with the lowest load $t_{\tilde{m}}$, translation vectors are assigned until the count of added operations remains below the gap $\max\{t_m \mid \forall m\} - t_{\tilde{m}}$. This reassignment is performed once for all workers, except the one already having the maximum workload. This step may introduce minor memory overhead due to more uniform distribution.
3. Any remaining translations are then distributed among all workers in a final balancing pass to further minimize inhomogeneity. Given the small size of the remaining translations, this step results in negligible additional memory consumption.

In our current implementation, all translation vectors utilize identical ML network structures, allowing scheduling to depend solely on translation counts. However, for optimal performance in more generalized settings, network complexity variations across translation directions should also be factored into the scheduling process.

Algorithm 4 MVM Part

procedure MVM STAGE(x)

Obtain uniform basis function values $\bar{C}_i$'s for each box with x and applying Algorithm 1.

parfor $m = 1, \dots, M_w$ **do**

for $\tilde{t} = 1, \dots, t_m$ **do**

 Calculate $\tilde{t}^{\text{th}}$ box-to-box translation (which gives output as $\bar{E}_{i-j}$) in m^{th} worker with ML networks and obtain $\bar{E}_j$'s.

end for

end parfor

Calculate test function value in the related box with $\bar{E}_j$'s and applying Algorithm 3.

end procedure

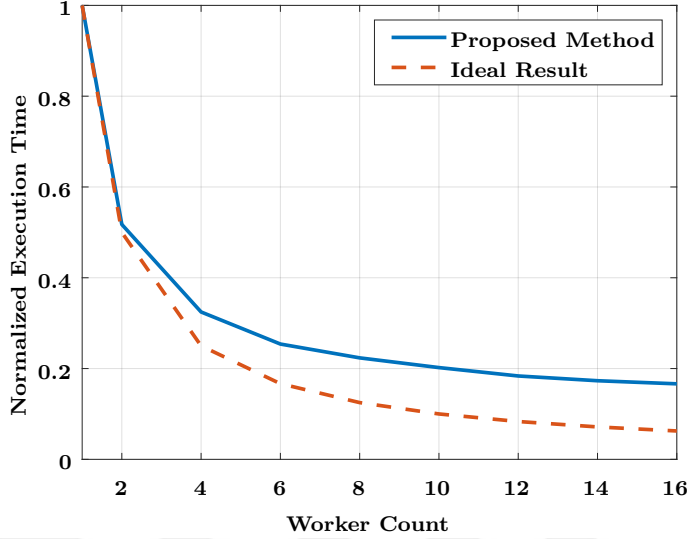


Figure 3.7: Normalized execution time of the proposed method for the far-zone MVMs of a 0.6m diameter PEC sphere discretized with 114318 basis functions. Illumination is a unit amplitude uniform plane wave at 1 GHz.

Finally, the MVM procedure used in the iterative solver is summarized in Algorithm 4, where each worker performs its assigned translations in parallel using the pre-trained ML networks.

To validate the strong scalability of the proposed method using ML networks for translations, we measure the average MVM time for computing far-zone interactions of a PEC sphere. The sphere is discretized into 114,318 RWG basis functions and partitioned into 8^4 child boxes, matching the box size a of the leaf level in a 5-level MLFMA. Figure 3.7 illustrates the normalized runtime as a function of the number of parallel workers (up to 16), and compares it with the ideal linear scaling trend. All measurements are performed using an Intel Xeon Silver 4210R processor.

While the proposed parallelization scheme demonstrates strong scalability, its efficiency begins to diverge from the ideal trend as the number of workers increases. This deviation arises because certain parts of the algorithm, specifically, the mapping stages, remain serial. As more workers are added, the relative contribution of these serial components to the overall runtime becomes more pronounced, thus limiting scalability. One possible remedy is to incorporate the

mapping operations into the parallel workflow. Although this modification can improve strong scaling efficiency, it may also lead to increased memory usage and overhead. Nonetheless, in this example, the proposed method completes all far-zone interactions in 15.87 seconds using only 4 workers, outperforming the 5-level MLFMA, which takes 16.91 seconds. The runtime data reported in Fig. 3.7 represents the average of 10 independent runs.

3.4 Memory and Time Complexity of the Proposed Method

In this section, we present the memory and time complexity analysis for computing far-field interactions using the proposed method. The memory and time costs associated with each stage of MVM iterations are detailed separately, and the preprocessing memory costs for the first and third stages are also discussed.

In the first stage, the uniform basis function coefficients for each basis box are obtained using the precomputed mapping vectors $\bar{C}'_n$ corresponding to each subdomain basis function. As each $\bar{C}'_n$ has a size of $3N_d$, the memory complexity of this stage is $\mathcal{O}(NN_d)$. The time complexity is also $\mathcal{O}(NN_d)$, since each coefficient α_n is multiplied by $\bar{C}'_n$ to obtain the equivalent Hertzian dipole representation, as described in (3.12).

In the second stage, the dominant memory consideration arises from the storage of neural networks (NNs). Each NN requires $\mathcal{O}(N_d H)$ memory, where H is the hidden layer dimension. Given that there are $|\mathbb{W}|$ unique NNs (corresponding to the distinct translation vectors), the total memory complexity of this stage is $\mathcal{O}(|\mathbb{W}|N_d H)$. The time complexity for translation using NNs is $\mathcal{O}(N_B^2 N_d H)$, where N_B denotes the number of child boxes.

The third stage, being the inverse mapping of the first stage, incurs the same memory and time complexity as the first stage, i.e., $\mathcal{O}(NN_d)$. Therefore, the total memory complexity for far-field MVM operations per iteration is $\mathcal{O}(N_d(N +$

$|\mathbb{W}|H))$, and the overall time complexity is $\mathcal{O}(N_B^2 N_d H + N N_d)$.

$$\mathcal{O}(N_d(N + |\mathbb{W}|H)) \quad (3.23)$$

It is important to note that near-field matrix elements are computed identically to those in MLFMA, provided the child box size in the proposed method equals the leaf-level box size in MLFMA. Hence, the memory and time complexities for near-field interactions remain the same as in MLFMA. For preprocessing, the memory required to store the mapping vectors $\bar{C}'_n$ and $\bar{D}'_n$ (used in the first and third stages, respectively) is also $\mathcal{O}(N N_d)$.

The total time complexity of an MVM operation becomes:

$$\mathcal{O}(N_B^2 N_d H + N N_d + N N_b), \quad (3.24)$$

where $N_b = N/N_B$ denotes the average number of subdomain basis functions per child box, and $\mathcal{O}(N N_b)$ accounts for near-field computations. By applying a standard FMM-like complexity analysis for SIEs, and assuming $\mathcal{O}(N_B) = \mathcal{O}(N_b) = \mathcal{O}(\sqrt{N})$, the complexity simplifies to:

$$\mathcal{O}(N N_d H + N^{1.5}). \quad (3.25)$$

Using the relation $\mathcal{O}(N_d) = \mathcal{O}((ka)^2) = \mathcal{O}(N_b) = \mathcal{O}(\sqrt{N})$ from (3.11), we obtain an overall complexity of $\mathcal{O}(N^{1.5} H)$. If H is treated as a constant, this yields an asymptotic time complexity of $\mathcal{O}(N^{1.5})$, consistent with FMM for SIE problems.

Two additional remarks are warranted. (i) Due to the strong scalability of the translation stage, N_B can be chosen larger than $\sqrt{N}$. In such cases, the increase in translation time is offset by reduced costs in the mapping stages and fewer near-field computations. Moreover, the gain in parallel efficiency can lead to total complexities lower than $\mathcal{O}(N^{1.5})$. (ii) For certain classes of scatterers (e.g., elongated structures as shown in the *Numerical Results* section), the required number of NNs is significantly smaller due to geometric symmetries, thereby reducing the CPU time. While the asymptotic time complexity remains $\mathcal{O}(N^{1.5})$, the associated constant can be significantly lower.

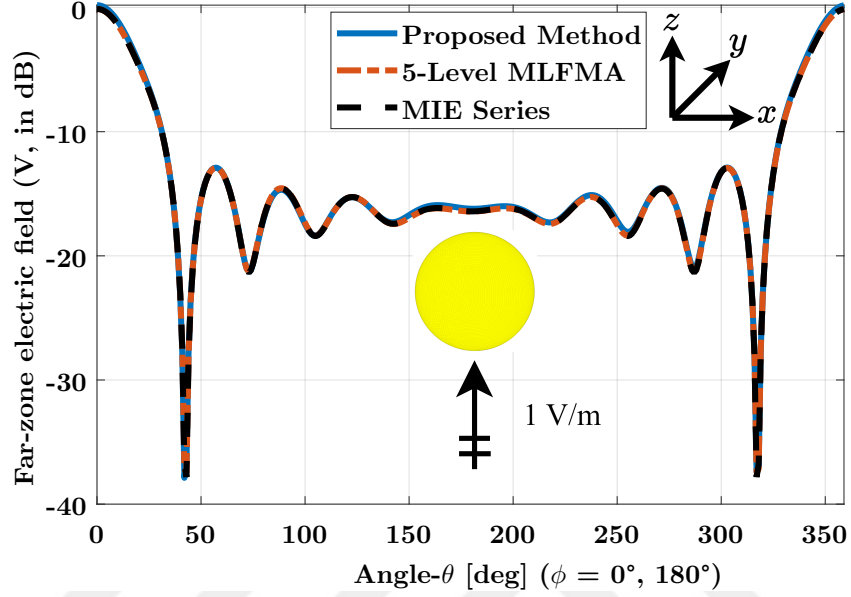


Figure 3.8: Far-zone co-polarized electric fields scattering from a PEC sphere of diameter 0.6λ (2λ) illuminated with a unit amplitude uniform plane wave at 1 GHz. The discretized surface consists of 114318 basis functions.

3.5 Numerical Results

To evaluate the performance of the proposed method, numerical simulations are conducted for both PEC and penetrable objects, discretized using RWG and SWG basis functions, respectively. EFIE is used in all test cases, where the D-type formulation [51] is employed for penetrable objects. The linear systems are solved using the generalized minimal residual (GMRES) iterative solver without any preconditioner, with the following solver parameters: an error tolerance of 10^{-3} , a restart value of 100, and a maximum of 1000 iterations. For numerical integration, a second-order Gaussian quadrature rule is applied over the surfaces associated with basis functions, while a first-order Gaussian quadrature is used for testing function-related surfaces.

The benchmark scattering problems include PEC spheres, dielectric spheres, the PEC NASA Almond model, and the PEC Flamme model [52]. All models, except the Flamme, are excited by an x -polarized uniform plane wave of unit amplitude (1 V/m) traveling in the z direction. The Flamme model is illuminated

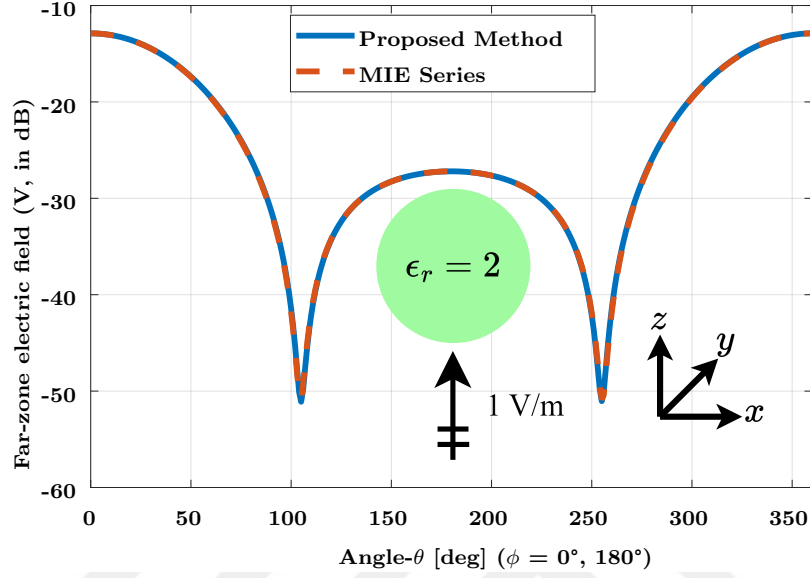


Figure 3.9: Far-zone co-polarized electric fields scattering from a 0.6m diameter dielectric sphere with $\epsilon_r = 2$, illuminated with a unit amplitude uniform plane wave at 250 MHz. The volume of the sphere is discretized with 38169 SWG functions.

by a y -polarized plane wave of unit amplitude impinging from $(\phi, \theta) = (0, 60^\circ)$, corresponding to a 30° elevation angle relative to the nose of the object. CPU timing measurements are performed using an Intel Xeon w5-3423 processor, with the exception of the two-layer dielectric case, which is executed on an Intel Xeon Gold 6230 CPU. For accuracy, each CPU time is computed as the average over 10 independent runs. In the tables presented throughout this section, NF and FF

Table 3.3: Memory Requirements and CPU Time Results for 2λ Sized Sphere. The L value is the same in MLFMA and the proposed method.

2λ Sized Sphere	5-Level MLFMA	Proposed Method 1-Worker	Proposed Method 4-Worker
One MVM (FF) CPU Time (sec.)	4.9	13.5	4.6
One MVM (NF) CPU Time (sec.)	8.5		
(FF) Memory (GB)	0.02	0.95	
(NF) Memory (GB)	2.84		
Iteration Count	689	654	
Total Time (Hour)	2.56	4.00	2.38

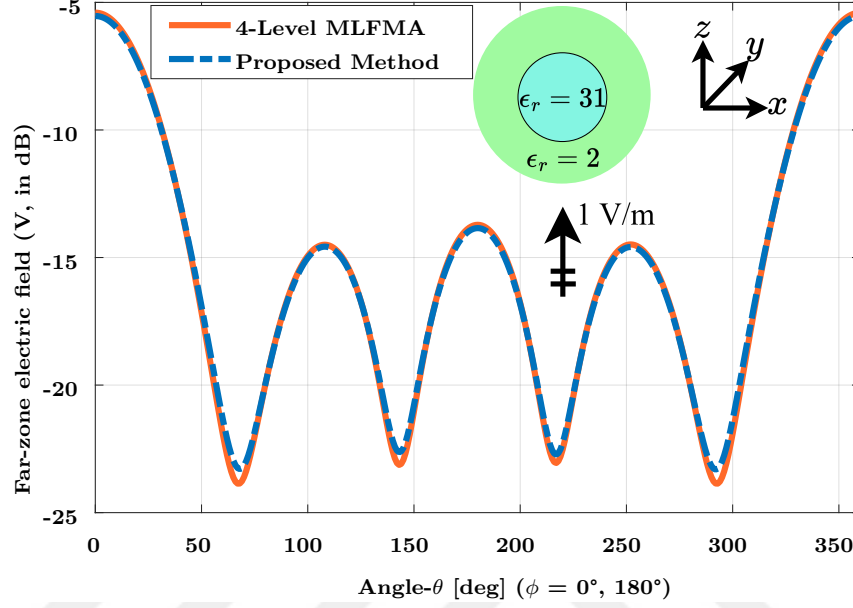


Figure 3.10: Far-zone co-polarized electric fields scattering from a two-layer dielectric sphere with the core diameter 0.3 m ($\epsilon_r = 31$) and the thickness is 0.15 m ($\epsilon_r = 2$), illuminated with a unit amplitude uniform plane wave at 500 MHz. The entire volume is discretized with 259393 SWG functions.

denote near-field and far-field interactions, respectively, when reporting memory usage and computation time.

As the first example, we consider a PEC sphere with a diameter of 0.6 m (2λ) at 1 GHz, discretized using 114,318 RWG basis functions. It is worth noting that this same geometry is also employed in the two-stage training process described in Section 3.2 to train the ML networks. The PEC sphere is partitioned into 8^4

Table 3.4: Memory Requirements and CPU Time Results for λ -Sized Two-Layer Dielectric Sphere. The L value is 4 in both MLFMA and the proposed method.

λ -Sized Two-Layer Dielectric Sphere	4-Level MLFMA	Proposed Method 1-Worker
One MVM (FF) CPU Time (sec.)	10	8.9
One MVM (NF) CPU Time (sec.)	80.1	
(FF) Memory (GB)	0.03	1.63
(NF) Memory (GB)	291	
Iteration Count	1070	1065
Total Time (Hour)	26.78	26.33

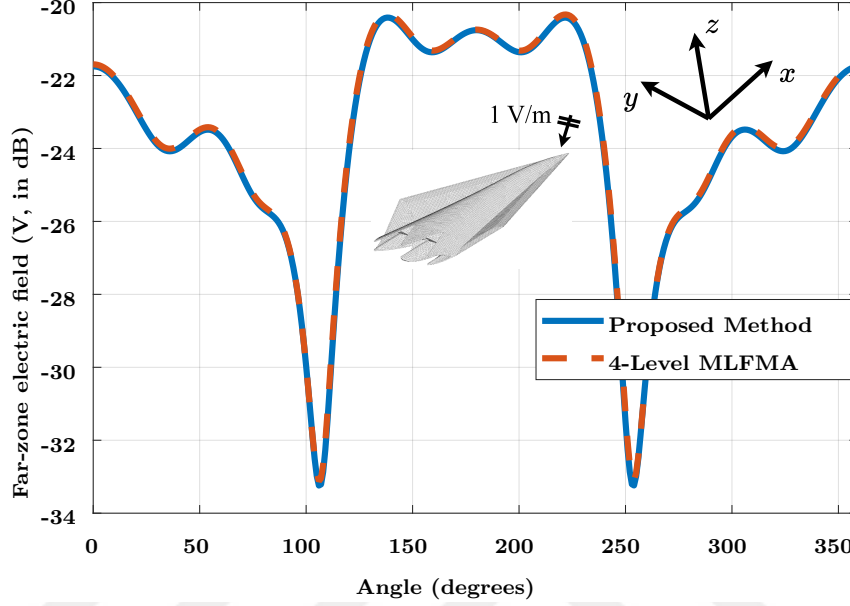


Figure 3.11: Far-zone electric field along the E-plane ($\phi = 0^\circ$) scattered from a 0.6 m PEC Flamme model illuminated with a unit amplitude uniform plane wave impinging from $(\phi, \theta) = (0, 60^\circ)$, i.e., 30° elevation from the nose at 1 GHz. The surface of the Flamme is discretized with 51669 RWG functions.

boxes, resulting in a child box size of $a = \lambda/8$, which coincides with the leaf-level box size used in a 5-level MLFMA. Figure 3.8 shows the far-zone scattered electric field computed with the proposed method, alongside the results obtained using the 5-level MLFMA and the analytical Mie series solution. The proposed method exhibits perfect agreement with both MLFMA and Mie series results. Table 3.3 summarizes the corresponding memory requirements and CPU time. According to these results, the proposed method achieves performance comparable to MLFMA when four workers are employed. Although the FF memory consumption is higher for the proposed method compared to MLFMA, the overall memory usage is still primarily determined by the NF contributions.

It is important to highlight that, for all subsequent numerical examples fitting within a mother box of size 2λ and using a child box size of $a = \lambda/8$, the same ML network weights are employed throughout. To validate this claim, even in the context of dielectric objects and VIE formulations, two dielectric sphere examples are examined. First, a dielectric sphere with a diameter of 0.6 m ($\lambda/2$ at 250 MHz)

and relative permittivity $\epsilon_r = 2$ is considered. This object is discretized using 38,169 SWG basis functions, and its far-zone scattered electric field computed by the proposed method is compared to the analytical Mie series solution. As depicted in Fig. 3.9, the results are in excellent agreement. Next, a two-layer dielectric sphere is analyzed. The inner core has a diameter of 0.3 m ($\lambda/2$ at 500 MHz) with $\epsilon_r = 31$, while the surrounding layer has $\epsilon_r = 2$ and a thickness of 0.15 m, resulting in a total diameter of 0.6 m (λ at 500 MHz). The full structure is discretized using 259,393 SWG basis functions. Far-zone scattered electric field results from the proposed method and a 4-level MLFMA are shown in Fig. 3.10, once again demonstrating excellent agreement. Memory consumption and CPU time measurements are summarized in Table 3.4. Since both methods use the same number of levels L , the NF contributions are identical and dominate the overall memory and runtime. In terms of FF computations, the proposed method achieves slightly lower CPU time than MLFMA, even with a single worker, albeit with increased FF memory usage.

The next set of numerical experiments involves the PEC Flamme model and the PEC NASA Almond model, both with an electrical size of 2λ at 1 GHz. As in previous examples, the same ML network weights are utilized for both cases. The Flamme model is discretized using 51,669 RWG basis functions, while the NASA Almond model employs 105,159 RWG basis functions. For both scatterers, the E-plane far-zone scattered electric field and surface current density are computed using the proposed method and compared against the results obtained from the 4-level MLFMA. Figures 3.11 and 3.12 display the comparisons for the Flamme

Table 3.5: Memory Requirements and CPU Time Results 2λ Sized Flamme. The L value is 4 in MLFMA and 5 in the proposed method.

2λ Sized Flamme	4-Level MLFMA	Proposed Method 1-Worker
One MVM (FF) CPU Time (sec.)	2.5	0.6
One MVM (NF) CPU Time (sec.)	44.2	15.1
(FF) Memory (GB)	0.01	0.38
(NF) Memory (GB)	15.7	5.32
Iteration Count	1200	1200
Total Time (Hour)	15.57	5.23

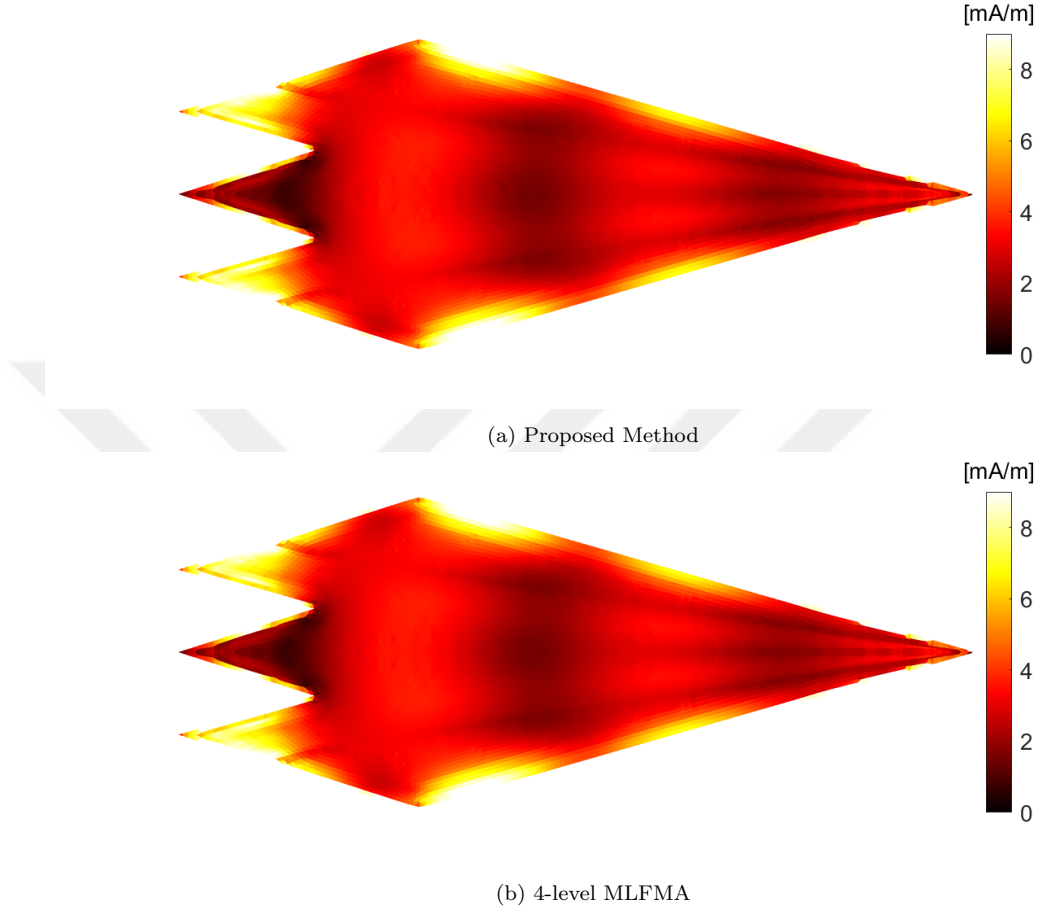


Figure 3.12: Comparison of the induced current density magnitudes for the (a) proposed method and (b) 4-level MLFMA, when the PEC Flamme is illuminated with a unit amplitude uniform plane wave impinging from $(\phi, \theta) = (0, 60^\circ)$, i.e., 30° elevation from the nose at 1 GHz.

model in terms of the far-zone scattered field and surface current distribution, respectively, showing excellent agreement between the two methods. Memory usage and CPU timing results for the Flamme model are reported in Table 3.5. In both the proposed method and MLFMA, NF memory and NF CPU time are the dominant contributors. Notably, the number of levels L is set to 5 in the proposed method and 4 in MLFMA due to the LFB problem of MLFMA. As a result, the proposed method exhibits lower NF memory and NF CPU time than MLFMA. Additionally, the FF computation time in the proposed method is significantly reduced compared to that of the 4-level MLFMA, even when executed with a single worker.

Similarly, Figs. 3.13 and 3.14 present the comparisons of the E-plane far-zone

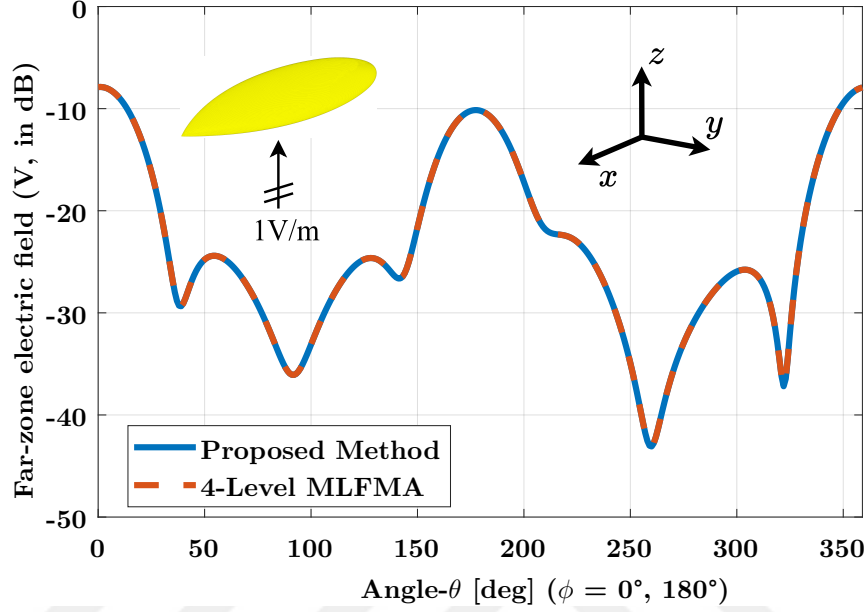


Figure 3.13: Far-zone electric field along the E-plane scattered from a 0.6 m PEC NASA Almond model illuminated with a unit amplitude uniform plane at 1 GHz. The surface of the NASA Almond is discretized with 105159 RWG functions.

scattered electric field and surface current density, respectively, for the NASA Almond model. Once again, very good agreement is observed between the results of the proposed method and those of the 4-level MLFMA. Table 3.6 provides the corresponding memory requirements and CPU time measurements, which follow a similar trend to those reported for the Flamme example.

Table 3.6: Memory Requirements and CPU Time Results for 2λ Sized NASA Almond. The L value is 4 in MLFMA and 5 in the proposed method.

2λ Sized NASA Almond	4-Level MLFMA	Proposed Method 1-Worker
One MVM (FF) CPU Time (sec.)	5.1	1.2
One MVM (NF) CPU Time (sec.)	143	44.9
(FF) Memory (GB)	0.02	0.71
(NF) Memory (GB)	51.52	16
Iteration Count	1913	1912
Total Time (Hour)	78.70	24.48

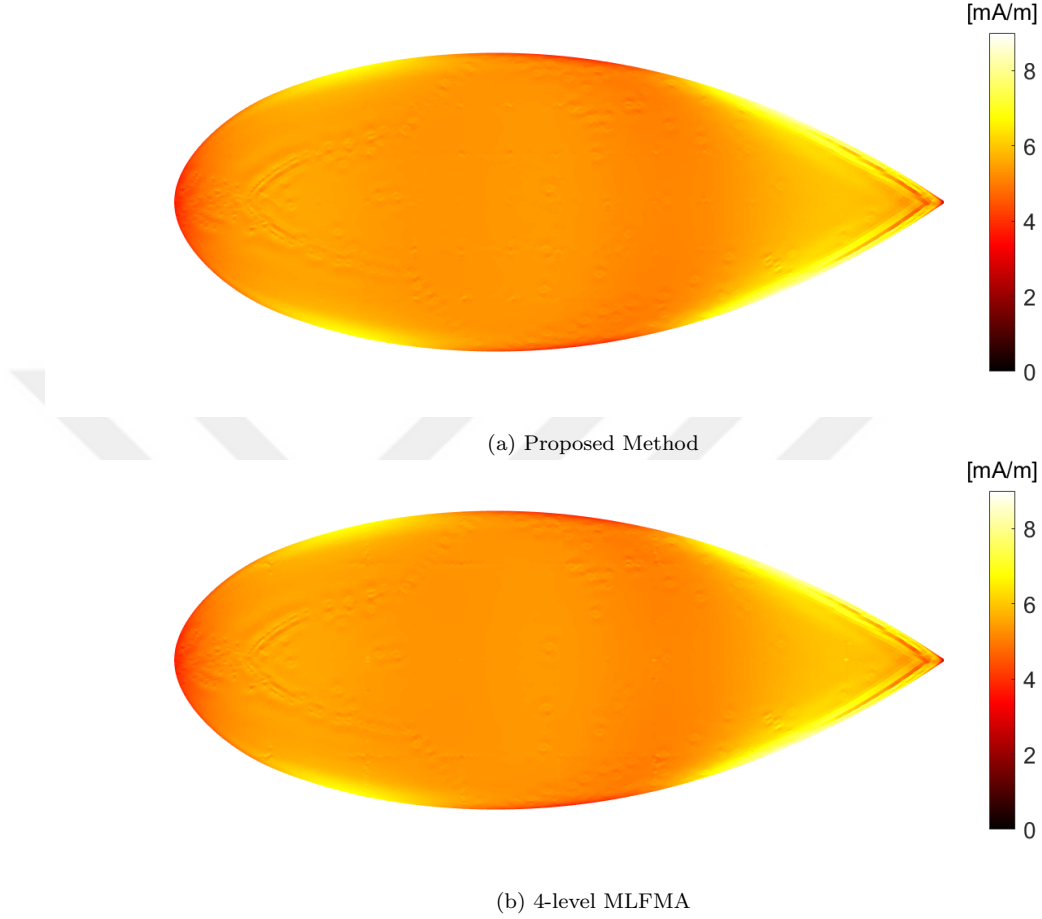


Figure 3.14: Comparison of the induced current density magnitudes for the (a) proposed method and (b) 4-level MLFMA, when the PEC NASA Almond is illuminated with a unit amplitude uniform plane at 1 GHz.

For both the NASA Almond and Flamme models, the proposed method requires less NF memory and NF CPU time, both of which are the dominant contributors, compared to the 4-level MLFMA. This reduction is primarily attributed to the use of a smaller child box size in the proposed method relative to the leaf-level box size in MLFMA, which is constrained by the LFB problem. Although the proposed method performs a greater number of FF computations than MLFMA, the total FF CPU time remains significantly lower, even when only a single worker is used in both numerical examples. This improvement is due to the high aspect ratio of the scatterers, which causes inefficiencies in the hierarchical structure of MLFMA, while the proposed method continues to exploit symmetries to retain computational efficiency in such cases. In contrast to

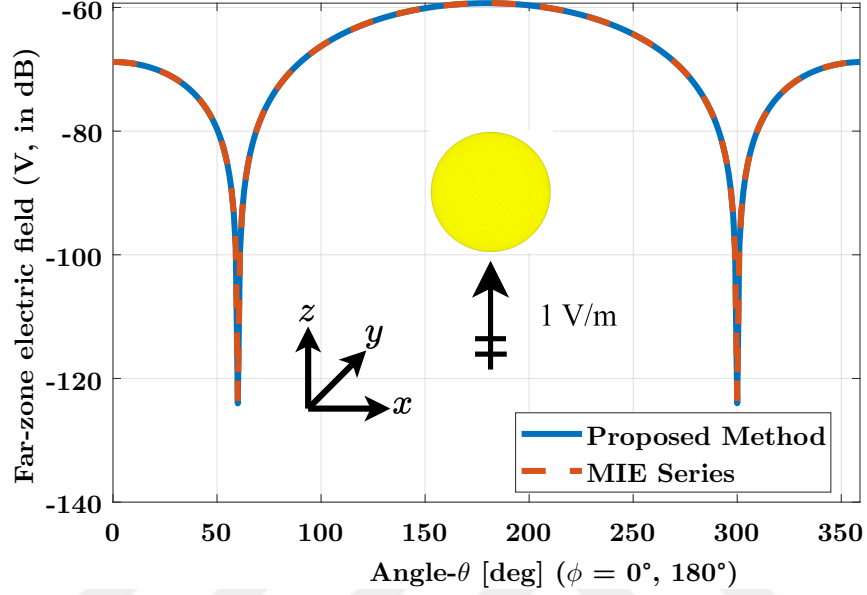


Figure 3.15: Far-zone co-polarized electric fields scattering from a PEC sphere of diameter 0.6m ($\lambda/64$) illuminated with a unit amplitude uniform plane wave at 7.8125 MHz. The discretized surface consists of 25995 basis functions.

the NASA Almond and Flamme examples, where MLFMA struggles due to the high aspect ratio, the λ -sized two-layer dielectric sphere presents a different configuration. Even in this case, the proposed method slightly outperforms MLFMA in terms of FF CPU time when $L = 4$ and a single worker is used. These results lead to the following conclusion: although the proposed method shares the same asymptotic time complexity as FMM, namely $\mathcal{O}(N^{1.5})$, it benefits from a smaller

Table 3.7: Memory Requirements and CPU Time Results for $\lambda/64$ Sized Sphere. The L value is 3 in the proposed method.

$\lambda/64$ Sized Sphere	MoM	Proposed Method 1-Worker
One MVM (FF) CPU Time (sec.)	-	0.15
One MVM (NF) CPU Time (sec.)	30.9	7.15
(FF) Memory (GB)	-	0.16
(NF) Memory (GB)	10.82	2.5
Iteration Count	79	81
Total Time (Hour)	0.68	0.16

multiplicative constant compared to FMM or 4-level MLFMA, particularly depending on the geometry of the scatterer. Across all examples, while the proposed method incurs higher FF memory costs, overall memory usage continues to be dominated by NF memory.

Finally, the last numerical example highlights the proposed method’s robustness against the LFB problem, which severely affects conventional MLFMA. For this demonstration, the same PEC sphere with a diameter of 0.6 m is used, but the frequency is reduced to 7.8125 MHz. At this frequency, the sphere becomes electrically small ($\lambda/64$), and it is partitioned into 8^2 boxes, resulting in a child box size of $a = \lambda/256$. The ML networks are again trained using the two-stage training strategy described earlier. Figure 3.15 shows the far-zone scattered electric field results obtained using the proposed method in comparison with the Mie series solution, demonstrating perfect agreement. As anticipated, conventional MLFMA fails to solve this problem due to LFB. Table 3.7 reports the memory and CPU time requirements of both the proposed method and MoM.

It is important to note that the proposed method is not hierarchical in nature. As a result, direct efficiency comparisons with N -level MLFMA for $N \geq 4$ may not be entirely equitable, particularly when both methods are executed using a single worker for electrically large problems. Nevertheless, across all the presented examples, as well as several additional cases not included here for brevity, the proposed method consistently outperforms 3-level MLFMA when executed on a single core. Furthermore, its efficiency becomes comparable to that of N -level MLFMA (with $N \geq 4$) when a sufficiently large number of workers is employed.

Chapter 4

A Novel Taylor Expansion and Virtual Box Based Hierarchical Method for the Fast Solution of IEs

The second part of this thesis presents a novel hierarchical method for efficiently computing far-zone interactions in frequency-domain IEs, particularly suited for electrically large and elongated problems.

In computational electromagnetics, the accurate and efficient analysis of elongated structures (objects with high aspect ratios), such as missiles, ships, aircraft, wires, antennas, and waveguides, is crucial due to their widespread use in practical applications. However, MLFMA faces substantial limitations when applied to such geometries. Specifically, it struggles to maintain its desirable computational complexity of $\mathcal{O}(N \log N)$ without sacrificing accuracy [24]. This inefficiency arises from the isotropic directional sampling used in MLFMA's plane-wave expansions: although only a limited set of directions is physically relevant for elongated or quasi-one-dimensional structures, MLFMA enforces computations across a full directional spectrum, resulting in significant redundant effort

and computational overhead.

To overcome this inefficiency, we propose a directional method particularly for elongated problems, leveraging a specially constructed *reduced Green’s function*, which is a plane-wave-extracted form of the homogeneous scalar Green’s function. This function refines the information carried by the plane wave for a specified translation between groups of source and observation points, ensuring improved accuracy over the reduced direction set. The dominant component of the reduced kernel is a Gaussian-wave term, and by applying a control scheme to this term, the entire expansion, including higher-order contributions, follows this controlled behavior, enabling accurate representation as the frequency increases. To ensure that the cost of evaluating this Gaussian component remains independent of frequency, we introduce a novel *virtual box structure*, which serves as localized expansion regions. Each box of size a is partitioned into $\sqrt{a}$ -sized virtual boxes, thereby enabling frequency-invariant control over computational complexity and ensuring the desired $\mathcal{O}(N \log N)$ scaling.

Moreover, since the reduced Green’s function does not admit a separable expansion using known orthogonal bases, such as spherical harmonics, we utilize a *truncated Taylor series expansion* to capture its directional dependence. Each term in this expansion acts as a correction to the underlying plane-wave term, enabling systematic control over the approximation error by tuning the expansion order. This strategy provides a flexible and accurate representation of far-field interactions specific to directional geometries.

To fully harness these components in a hierarchical framework, we integrate the reduced kernel and the virtual box scheme into a hierarchical tree structure (analogous to that used in MLFMA, yet fundamentally different in implementation) to efficiently separate and compute far-zone interactions. This involves aggregation and disaggregation stages, adapted to operate over virtual boxes nested within boxes in the hierarchical tree. The result is an efficient and hierarchical algorithm with $\mathcal{O}(N \log N)$ complexity, specifically crafted for the challenges posed by elongated scatterers. This approach offers a robust and accurate alternative to MLFMA in scenarios where traditional isotropic sampling becomes

computationally burdensome.

4.1 Matrix-Vector Product Evaluation

As explained before, the MoM discretization of a SIE leads to a dense linear system in the form of

$$\bar{\bar{Z}}\bar{\alpha} = \bar{V}, \quad (4.1)$$

where $\bar{\bar{Z}} \in \mathbb{C}^{N \times N}$ is the impedance matrix, $\bar{\alpha} \in \mathbb{C}^{N \times 1}$ contains the unknown coefficients associated with the basis functions, and $\bar{V} \in \mathbb{C}^{N \times 1}$ is the excitation vector. The proposed method focuses on accelerating the MVM involving $\bar{\bar{Z}}$ and iteratively refined estimates of $\bar{\alpha}$, which is typically the computational bottleneck in iterative solvers. Hence, it uses a tree structure very similar to MLFMA. Basically, the computational domain is enclosed within a large cubic box and recursively subdivided into smaller subboxes, forming a hierarchical tree structure as depicted in Fig. 4.1. Each stage of this subdivision defines a level of the tree. The root corresponds to the entire domain at level 1, and deeper levels contain progressively smaller boxes, each covering a subset of the geometry. This multi-level organization is key to enabling efficient far-zone computations by specifying how group-by-group interactions are managed within the hierarchical structure. Only non-empty boxes (those containing basis and test functions) are retained in the tree, and interactions are processed in a level-wise fashion. At each level, box interactions are categorized as either near-field (NF) or far-field (FF), based on a one-box-buffer criterion. This leads to the matrix decomposition given by

$$\bar{\bar{Z}}^{NF}\bar{\alpha} + \bar{\bar{Z}}^{FF}\bar{\alpha} = \bar{V}. \quad (4.2)$$

As illustrated in Fig. 4.2, at level l , the NF region (light blue) includes adjacent or buffer-touching boxes whose contributions are computed either directly and stored in $\bar{\bar{Z}}^{NF}$ at the finest level or combined from finer-level calculations. In contrast, the FF region (purple) consists of well-separated groups whose interactions are approximated via the hierarchical translation framework based on

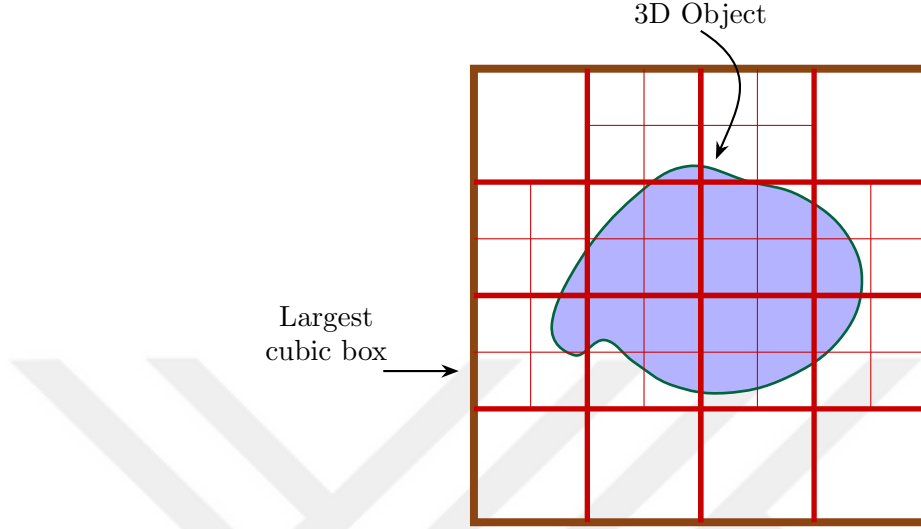


Figure 4.1: Recursive division (2D view for an arbitrary 3D object)

Taylor series expansions and the virtual box structure, which are completely novel for this method.

To evaluate $\bar{\bar{Z}}^{FF}\bar{\alpha}$, the proposed method employs a hierarchical interaction strategy that processes the interactions between clusters of basis and test functions within *virtual boxes*, which are subregions defined inside the boxes of the hierarchical tree. These virtual boxes serve as localized expansion centers at each box on each level, enabling the outgoing fields to be compactly represented through directional monomials formed via truncated Taylor series expansions.

The overall MVM is performed hierarchically through three main operations, which have similar names but completely different implementations than MLFMA, repeated across all levels of the tree:

Aggregation over Virtual Boxes: Each basis function group inside a virtual box inside of a box in the hierarchical tree collects the contributions of its basis functions into a local expansion defined over the associated virtual box. This expansion is constructed using directional Taylor monomials aligned with observation directions and truncated based on the desired accuracy. In the hierarchical framework, virtual boxes at level $l + 2$ are aggregated into virtual boxes at level l , as discussed in Section 4.4.

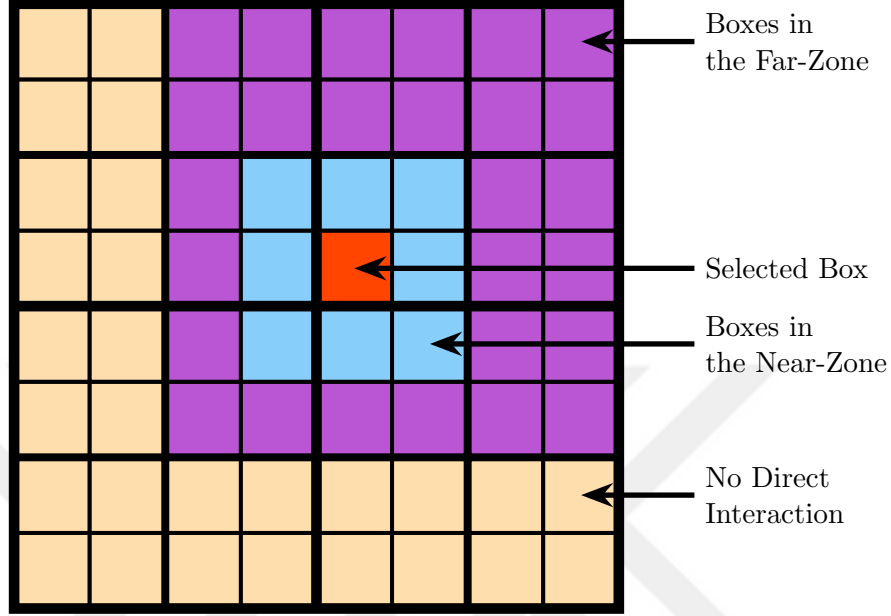


Figure 4.2: Near-field (NF) and Far-field (FF) for a selected box (in red)

Translation over Virtual Boxes: The aggregated expansions are transferred to the FF test boxes using precomputed translation operators between each pair of basis and test virtual boxes. These operators depend only on the relative positions of basis and test boxes at the same level and exploit the directional structure of the virtual box expansions.

Disaggregation over Virtual Boxes: The incoming expansions at each test box are first accumulated at the centers of its virtual boxes. Similar to the aggregation stage, virtual boxes at level l are disaggregated into virtual boxes at level $l + 2$, as discussed in Section 4.4. At the finest level, these expansions are evaluated on the testing functions located within the test virtual boxes to find the contributions from distant source groups.

This hierarchical level-by-level processing minimizes redundant calculations by collecting incoming field expansions at each level and propagating them downward, allowing efficient interaction evaluations. The directional structure introduced by the virtual boxes and the multilevel organization ensures scalability, particularly for electrically large and elongated geometries. Consequently, the computational complexity of the MVM operation is reduced from the naive $\mathcal{O}(N^2)$

to approximately $\mathcal{O}(N \log N)$, enabling the fast and memory-efficient solution of large-scale electromagnetic problems that would otherwise be intractable using conventional dense matrix methods.

4.2 Taylor Approximation of Green's Function

Let us consider the homogeneous scalar Green's function defined as

$$g(\bar{r}, \bar{r}'_1, \bar{r}'_2) = \frac{\exp(-jk|\bar{r} + \bar{r}'_1 + \bar{r}'_2|)}{4\pi|\bar{r} + \bar{r}'_1 + \bar{r}'_2|}, \quad (4.3)$$

where $r > |\bar{r}'_1 + \bar{r}'_2|$. For the rest of the analysis, we consider $\bar{r}$ as a fixed vector, and define

$$\bar{r}' = \bar{r}'_1 + \bar{r}'_2.$$

Then, the Green's function in (4.3) can be separated as

$$g(\bar{r}, \bar{r}'_1, \bar{r}'_2) = g(\bar{r}, \bar{r}') = G \exp(jk \hat{r} \cdot \bar{r}') p(\hat{r} \cdot \bar{r}', \hat{\theta} \cdot \bar{r}', \hat{\phi} \cdot \bar{r}'), \quad (4.4)$$

where $\hat{r}, \hat{\theta}, \hat{\phi}$ are unit vectors in the corresponding directions, $G = \frac{\exp(-jkr)}{4\pi r}$, and we refer to the function p as the reduced Green's function in the context of this formulation. This reduced function does not contain the plane wave component of the Green's function, which becomes the dominant term as the ratio $r/|\bar{r}'|$ increases.

To obtain a proper Taylor expansion of p in terms of $(\bar{r}'_1, \bar{r}'_2)$, there are two main approaches:

1. Taylor expansion with respect to $\bar{r}'$:

We treat $\bar{r}'$ as a single variable and expand p as a d -order Taylor series:

$$\begin{aligned} p(\hat{r} \cdot \bar{r}', \hat{\theta} \cdot \bar{r}', \hat{\phi} \cdot \bar{r}') &\approx p_d(\hat{r} \cdot \bar{r}', \hat{\theta} \cdot \bar{r}', \hat{\phi} \cdot \bar{r}') \\ &= \sum_{d_1=0}^d \sum_{d_2=0}^{d-d_1} \sum_{d_3=0}^{d-d_1-d_2} c_{d_1, d_2, d_3} (\hat{r} \cdot \bar{r}')^{d_1} (\hat{\theta} \cdot \bar{r}')^{d_2} (\hat{\phi} \cdot \bar{r}')^{d_3}, \end{aligned} \quad (4.5)$$

where $c_{d_1, d_2, d_3} \in \mathbb{C}$ and p_d is the d -order Taylor approximation of p with respect to $\bar{r}'$.

Using the identity $\bar{r}' = \bar{r}'_1 + \bar{r}'_2$ and applying the multinomial expansion, we obtain the separated form as follows:

$$\begin{aligned}
p_d(\hat{r} \cdot \bar{r}'_1, \hat{\theta} \cdot \bar{r}'_1, \hat{\phi} \cdot \bar{r}'_1, \hat{r} \cdot \bar{r}'_2, \hat{\theta} \cdot \bar{r}'_2, \hat{\phi} \cdot \bar{r}'_2) = \\
\sum_{d_1=0}^d \sum_{d_2=0}^{d-d_1} \sum_{d_3=0}^{d-d_1-d_2} \sum_{d'_1=0}^{d_1} \sum_{d'_2=0}^{d_2} \sum_{d'_3=0}^{d_3} c_{d_1, d_2, d_3} \binom{d_1}{d'_1} \binom{d_2}{d'_2} \binom{d_3}{d'_3} \\
\times (\hat{r} \cdot \bar{r}'_1)^{d'_1} (\hat{r} \cdot \bar{r}'_2)^{d_1-d'_1} (\hat{\theta} \cdot \bar{r}'_1)^{d'_2} (\hat{\theta} \cdot \bar{r}'_2)^{d_2-d'_2} (\hat{\phi} \cdot \bar{r}'_1)^{d'_3} (\hat{\phi} \cdot \bar{r}'_2)^{d_3-d'_3},
\end{aligned} \tag{4.6}$$

where $\binom{d_1}{d'_1} = \frac{d_1!}{d'_1!(d_1-d'_1)!}$ is the binomial coefficient.

Furthermore, a method to calculate coefficients c_{d_1, d_2, d_3} 's is given in Appendix A.

2. Taylor expansion with respect to $(\bar{r}'_1, \bar{r}'_2)$:

Alternatively, p can be directly expanded in both variables simultaneously. For a d -degree expansion in each, we write:

$$\begin{aligned}
p(\hat{r} \cdot \bar{r}', \hat{\theta} \cdot \bar{r}', \hat{\phi} \cdot \bar{r}') &\approx p'_d(\hat{r} \cdot \bar{r}'_1, \hat{\theta} \cdot \bar{r}'_1, \hat{\phi} \cdot \bar{r}'_1, \hat{r} \cdot \bar{r}'_2, \hat{\theta} \cdot \bar{r}'_2, \hat{\phi} \cdot \bar{r}'_2) \\
&= \sum_{d_1=0}^d \sum_{d_2=0}^{d-d_1} \sum_{d_3=0}^{d-d_1-d_2} \sum_{d'_1=0}^d \sum_{d'_2=0}^{d-d'_1} \sum_{d'_3=0}^{d-d'_1-d'_2} c'_{d_1, d_2, d_3, d'_1, d'_2, d'_3} \\
&\quad \times (\hat{r} \cdot \bar{r}'_1)^{d_1} (\hat{r} \cdot \bar{r}'_2)^{d'_1} (\hat{\theta} \cdot \bar{r}'_1)^{d_2} (\hat{\theta} \cdot \bar{r}'_2)^{d'_2} (\hat{\phi} \cdot \bar{r}'_1)^{d_3} (\hat{\phi} \cdot \bar{r}'_2)^{d'_3},
\end{aligned} \tag{4.7}$$

where $c'_{d_1, d_2, d_3, d'_1, d'_2, d'_3} \in \mathbb{C}$.

There is also a connection between the coefficients obtained via the two expansion approaches. Specifically, each monomial term of the form

$$(\hat{r} \cdot \bar{r}'_1)^{d_1} (\hat{r} \cdot \bar{r}'_2)^{d'_1} (\hat{\theta} \cdot \bar{r}'_1)^{d_2} (\hat{\theta} \cdot \bar{r}'_2)^{d'_2} (\hat{\phi} \cdot \bar{r}'_1)^{d_3} (\hat{\phi} \cdot \bar{r}'_2)^{d'_3}$$

has a unique coefficient in the Taylor expansion of p , which originates from the corresponding mixed partial derivatives. Therefore, the nonzero terms in the two expansions must represent the same set of monomials, although grouped

differently. By comparing the coefficients in (4.7) with those in (4.6), we obtain the following relationship:

$$c'_{d_1, d_2, d_3, d'_1, d'_2, d'_3} = c_{d_1+d'_1, d_2+d'_2, d_3+d'_3} \binom{d_1+d'_1}{d'_1} \binom{d_2+d'_2}{d'_2} \binom{d_3+d'_3}{d'_3}. \quad (4.8)$$

However, it is important to note a subtle distinction. In (4.7), the maximum degree for each variable (in both $\bar{r}'_1$ and $\bar{r}'_2$) is limited to d , which means that the total degree of some terms may exceed d . In contrast, the expansion in (4.6) constrains the total combined degree $d_1+d_2+d_3$ (and hence also $d'_1+d'_2+d'_3$) to be at most d . Consequently, while every term in (4.7) maps to a corresponding term in (4.6), the reverse is not true when both expansions are truncated at the same order d . That is, some higher-degree monomials in (4.6) will not be captured in (4.7) under this constraint. To ensure that all terms of (4.7) are covered by (4.6), one can instead use a total degree $2d$ in the latter. In that case, (4.6) fully spans the support of (4.7), although it will include some additional terms not present in the direct expansion.

In practical applications, when the truncation order d is chosen sufficiently large to meet a desired approximation accuracy, the omitted higher-order terms (those present in one expansion but not the other) typically contribute negligibly to the overall error. Thus, the two approaches are asymptotically equivalent in representing the reduced Green's function p and g , with the choice between them guided by computational convenience or the structure of the problem at hand.

These two approaches to computing the reduced Green's function p and the original one $g(\bar{r}, \bar{r}'_1, \bar{r}'_2)$ in (4.4) can be compactly expressed in the form of a *vector-matrix-vector multiplication* by adopting a structured representation of the Taylor series. Let us define a vector, called the basis monomials, of directional monomials associated with $\bar{r}'_1$ as

$$\bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1) := \exp(jk \hat{r} \cdot \bar{r}'_1) \left[(\hat{r} \cdot \bar{r}'_1)^{d_1} (\hat{\theta} \cdot \bar{r}'_1)^{d_2} (\hat{\phi} \cdot \bar{r}'_1)^{d_3} \right], \quad (4.9)$$

where each entry corresponds to a unique combination of non-negative integers (d_1, d_2, d_3) satisfying $d_1 + d_2 + d_3 \leq d$. The entries are ordered in a block fashion:

d_3 varies first, followed by d_2 , and d_1 varies last. That is, the monomials are arranged lexicographically in increasing order of (d_1, d_2, d_3) .

Similarly, define the vector, called the test monomials, associated with $\bar{r}'_2$ as

$$\bar{\mathfrak{T}}(\hat{r}, \bar{r}'_2) := \exp(jk \hat{r} \cdot \bar{r}'_2) \left[(\hat{r} \cdot \bar{r}'_2)^{d'_1} (\hat{\theta} \cdot \bar{r}'_2)^{d'_2} (\hat{\phi} \cdot \bar{r}'_2)^{d'_3} \right], \quad (4.10)$$

with the same block ordering applied to indices (d'_1, d'_2, d'_3) , subject to $d'_1 + d'_2 + d'_3 \leq d$.

We also define a coefficient matrix $\bar{\bar{T}} \in \mathbb{C}^{D \times D}$, where D , the total number of such monomial combinations up to degree d , is given by

$$D = \binom{d+3}{3}. \quad (4.11)$$

Each entry of $\bar{\bar{T}}$ corresponds to the coefficient $c'_{d_1, d_2, d_3, d'_1, d'_2, d'_3}$ (or those in (4.6)) arranged consistently with the ordering of entries in $\bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1)$ and $\bar{\mathfrak{T}}(\hat{r}, \bar{r}'_2)$. A pseudocode to construct it by using (4.7) is given in Algorithm 5.

With these definitions, both of the Taylor expansions in (4.6) and (4.7) can be expressed in compact matrix equation form:

$$\begin{aligned} g(\bar{r}, \bar{r}'_1, \bar{r}'_2) &= G \exp(jk \hat{r} \cdot \bar{r}') p'_d(\hat{r} \cdot \bar{r}'_1, \hat{\theta} \cdot \bar{r}'_1, \hat{\phi} \cdot \bar{r}'_1, \hat{r} \cdot \bar{r}'_2, \hat{\theta} \cdot \bar{r}'_2, \hat{\phi} \cdot \bar{r}'_2) \\ &\approx G \bar{\mathfrak{T}}(\hat{r}, \bar{r}'_2)^T \bar{\bar{T}} \bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1). \end{aligned} \quad (4.12)$$

The computational complexity of evaluating (4.12) through direct matrix-vector multiplication is $\mathcal{O}(D^2) = \mathcal{O}(d^6)$, where D is the number of monomial terms up to total degree d . However, when the expansion in (4.6) is used instead of the fully separated form in (4.7), the total number of coefficients required for the same expansion order d is significantly reduced. This results in a much smaller leading constant for the $\mathcal{O}(d^6)$ operation count, albeit at the cost of reduced approximation accuracy due to the smaller functional bases.

In the numerical experiments presented later in this chapter, we employ either dense matrix-vector multiplication for (4.7), or sparse matrix-vector multiplication in the case of (4.6), taking advantage of the sparsity induced by the lower

Algorithm 5 Construction of the Translation Matrix $\bar{\bar{T}}$

Input: Taylor order d

Output: Translation matrix $\bar{\bar{T}}$ of size $D \times D$

Initialize $\bar{\bar{T}} \leftarrow \mathbf{0}_{D \times D}$

Find Taylor coefficients c_{d_1, d_2, d_3} for $2d$ order using procedure given in Appendix A

for all multi-indices (d_1, d_2, d_3) with $d_1 + d_2 + d_3 \leq d$ **do**

for all multi-indices (d'_1, d'_2, d'_3) with $d'_1 + d'_2 + d'_3 \leq d$ **do**

$m \leftarrow \text{INDEX}(d_1, d_2, d_3)$

$n \leftarrow \text{INDEX}(d'_1, d'_2, d'_3)$

 Set:

$$\bar{\bar{T}}_{m,n} = c_{d_1+d'_1, d_2+d'_2, d_3+d'_3} \binom{d_1+d'_1}{d_1} \binom{d_2+d'_2}{d_2} \binom{d_3+d'_3}{d_3}$$

end for

end for

return $\bar{\bar{T}}$

function $\text{INDEX}(d_1, d_2, d_3)$

$$\text{return } n = \sum_{k=0}^{d_1+d_2+d_3-1} \binom{k+2}{2} + \sum_{\ell=0}^{d_1+d_2-1} \binom{\ell+1}{1} + d_3$$

end function

number of nonzero coefficients. In the remainder of this section, we introduce further acceleration techniques specifically aimed at improving the efficiency of (4.12), which are not implemented in the numerical experiments.

Up to this point, the expansions and representations described above rely solely on the Taylor expansion of the reduced Green's function p in (4.4), without exploiting any additional analytical properties of p . Such purely algebraic expansions can in fact be applied to arbitrary smooth functions, as discussed in works such as [40–42]. However, the function p possesses an additional structure such that it satisfies a differential equation that can be used to improve both analytical understanding and numerical efficiency. For simplicity, we define

$$z' := \hat{r} \cdot \bar{r}', \quad y' := \hat{\theta} \cdot \bar{r}', \quad x' := \hat{\phi} \cdot \bar{r}',$$

and express p as a function of these scalar projections: $p = p(x', y', z')$. Recalling that the full Green's function satisfies the scalar Helmholtz equation in the primed

coordinates:

$$\nabla'^2 g(\bar{r}, \bar{r}') + k^2 g(\bar{r}, \bar{r}') = 0, \quad \text{for } r' < r, \quad (4.13)$$

and using the decomposition

$$g(\bar{r}, \bar{r}') = \frac{\exp(-jkr)}{4\pi r} \exp(jkz') p(x', y', z'), \quad (4.14)$$

we apply the Laplacian operator to g in the primed coordinate to find

$$\nabla'^2 g = \frac{\exp(-jkr)}{4\pi r} \exp(jkz') \left(\nabla'^2 p + 2jk \frac{\partial p}{\partial z'} - k^2 p \right). \quad (4.15)$$

Substituting this into the Helmholtz equation, we obtain

$$\nabla'^2 g + k^2 g = \frac{\exp(-jkr)}{4\pi r} \exp(jkz') \left(\nabla'^2 p + 2jk \frac{\partial p}{\partial z'} \right) = 0, \quad (4.16)$$

which implies that the reduced Green's function satisfies the following inhomogeneous partial differential equation (PDE):

$$\nabla'^2 p(x', y', z') + 2jk \frac{\partial p(x', y', z')}{\partial z'} = 0. \quad (4.17)$$

The differential equation in (4.17) remains valid even when $\bar{r}'$ is replaced by either $\bar{r}'_1$ or $\bar{r}'_2$, owing to the shift-invariance of the reduced Green's function p . Since p depends only on the relative position vector $\bar{r}'$, any rigid translation of the source coordinates does not alter the form of the governing PDE. This invariance enables us to incorporate the differential structure of p directly into the matrix-vector representation introduced in (4.12).

Let $\bar{\mathfrak{C}}$ be the coefficient vector obtained by the matrix-vector product

$$\bar{\mathfrak{C}} = \bar{T} \bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1). \quad (4.18)$$

This vector encodes the Taylor expansion coefficients corresponding to the test monomials represented in $\bar{\mathfrak{T}}(\hat{r}, \bar{r}'_2)$, where each entry is indexed by the triplet (d'_1, d'_2, d'_3) . Specifically, the monomial associated with each index is given by

$$(\hat{r} \cdot \bar{r}'_2)^{d'_1} (\hat{\theta} \cdot \bar{r}'_2)^{d'_2} (\hat{\phi} \cdot \bar{r}'_2)^{d'_3} = z'^{d'_1} y'^{d'_2} x'^{d'_3}, \quad (4.19)$$

and the full Taylor expansion matrix-vector form in (4.12) can be expressed as

$$\sum_{d'_1=0}^d \sum_{d'_2=0}^{d-d'_1} \sum_{d'_3=0}^{d-d'_1-d'_2} G \bar{\mathfrak{C}}_{d'_1, d'_2, d'_3} z'^{d'_1} y'^{d'_2} x'^{d'_3} \exp(jkz') = G \bar{\mathfrak{T}}(\hat{r}, \bar{r}'_2)^T \bar{T} \bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1). \quad (4.20)$$

Due to the governing PDE in (4.17), the coefficients of these monomials must satisfy a recurrence relation that captures the effect of the Laplacian and the directional derivative with respect to z' . In particular, for a given triple (d'_1, d'_2, d'_3) , the coefficients associated with the monomials are in the form of

$$x'^{d'_3+2} y'^{d'_2} z'^{d'_1}, \quad x'^{d'_3} y'^{d'_2+2} z'^{d'_1}, \quad x'^{d'_3} y'^{d'_2} z'^{d'_1+1}, \quad x'^{d'_3} y'^{d'_2} z'^{d'_1+2}$$

and they must satisfy the identity (4.17):

$$\begin{aligned} & (d'_3 + 2)(d'_3 + 1) \bar{\mathfrak{C}}_{d'_1, d'_2, d'_3+2} + (d'_2 + 2)(d'_2 + 1) \bar{\mathfrak{C}}_{d'_1, d'_2+2, d'_3} \\ & + 2jk(d'_1 + 1) \bar{\mathfrak{C}}_{d'_1+1, d'_2, d'_3} + (d'_1 + 2)(d'_1 + 1) \bar{\mathfrak{C}}_{d'_1+2, d'_2, d'_3} = 0. \end{aligned} \quad (4.21)$$

This recurrence relation encodes the requirement that the coefficients in $\bar{\mathfrak{C}}$ correspond to a function that satisfies the reduced Green's function PDE. However, since we are working with a truncated expansion up to total degree d , not all of the terms appearing in (4.21) are necessarily contained in the expansion. Specifically, if any of the resulting indices (e.g., $d'_1 + 2$ or $d'_3 + 2$) exceed d , the corresponding terms lie outside the valid coefficient domain and must be excluded. To accommodate this, we instead apply the recurrence in reverse to express higher-order coefficients in terms of lower-order ones. For multi-indices satisfying $d'_1 + d'_2 + d'_3 \leq d - 2$, equation (4.21) can be rearranged to solve for $\bar{\mathfrak{C}}_{d'_1+2, d'_2, d'_3}$ as:

$$\begin{aligned} \bar{\mathfrak{C}}_{d'_1+2, d'_2, d'_3} = & -\frac{(d'_3 + 2)(d'_3 + 1)}{(d'_1 + 2)(d'_1 + 1)} \bar{\mathfrak{C}}_{d'_1, d'_2, d'_3+2} \\ & -\frac{(d'_2 + 2)(d'_2 + 1)}{(d'_1 + 2)(d'_1 + 1)} \bar{\mathfrak{C}}_{d'_1, d'_2+2, d'_3} - \frac{2jk}{d'_1 + 2} \bar{\mathfrak{C}}_{d'_1+1, d'_2, d'_3}. \end{aligned} \quad (4.22)$$

This backward recurrence enables the reconstruction of all higher-order coefficients in the $\hat{r}$ direction (z') starting from the base cases where $d'_1 = 0$ or $d'_1 = 1$. In other words, once the coefficients for these base indices are known, the entire vector $\bar{\mathfrak{C}}$ can be generated recursively. Moreover, this recurrence can be cast into

Algorithm 6 Construction of the Recurrence Matrix $\bar{\bar{R}}$

Input: Maximum order d

Output: Recurrence matrix $\bar{\bar{R}}$

Initialize $\bar{\bar{R}} \leftarrow [\mathbf{I}_{D' \times D'} \quad \mathbf{0}_{D' \times D-D'}]$

for $d_1 = 2$ to d **do**

for all multi-indices (d_1, d_2, d_3) with $d_2 + d_3 \leq d - d_1$ **do**

$n \leftarrow \text{INDEX}(d_1, d'_2, d'_3)$

$n_1 \leftarrow \text{INDEX}(d_1 - 2, d_2, d_3 + 2)$

$n_2 \leftarrow \text{INDEX}(d_1 - 2, d_2 + 2, d_3)$

$n_3 \leftarrow \text{INDEX}(d_1 - 1, d_2, d_3)$

 Compute recurrence coefficients:

$$\alpha = -\frac{(d_3 + 2)(d_3 + 1)}{(d_1)(d_1 - 1)}, \quad \beta = -\frac{(d_2 + 2)(d_2 + 1)}{(d_1)(d_1 - 1)}, \quad \gamma = -\frac{2jk}{d_1}$$

 Update matrix column n :

$$\bar{\bar{R}}_{:,n} = \alpha \bar{\bar{R}}_{:,n_1} + \beta \bar{\bar{R}}_{:,n_2} + \gamma \bar{\bar{R}}_{:,n_3}$$

end for

end for

return $\bar{\bar{R}}$

function $\text{INDEX}(d_1, d_2, d_3)$

$$\text{return } n = \sum_{k=0}^{d_1+d_2+d_3-1} \binom{k+2}{2} + \sum_{\ell=0}^{d_1+d_2-1} \binom{\ell+1}{1} + d_3$$

end function

a sparse linear transformation. That is, the full coefficient vector $\bar{\mathfrak{C}}$ can be computed from a reduced set of coefficients, denoted as $\bar{\mathfrak{C}}_{\text{red}}$, which includes only the components with $d'_1 = 0$ or $d'_1 = 1$. This transformation can be written as

$$\bar{\mathfrak{C}} = \bar{\bar{R}}^T \bar{\mathfrak{C}}_{\text{red}}, \quad (4.23)$$

where $\bar{\bar{R}}$ is a sparse recurrence matrix encoding the PDE-consistent dependencies between monomial coefficients. A pseudocode to construct it is given in Algorithm 6.

A similar recurrence construction can be applied to the basis side via reciprocity, enabling the full matrix-vector formulation in (4.12) to be expressed using reduced coefficient and basis spaces. Let $\bar{\bar{T}}_{\text{pre-red}}$ denote the truncated submatrix of the full coefficient matrix $\bar{\bar{T}}$, containing only entries corresponding to

test-side index combinations with $d'_1 = 0$ or 1. From the definition of $\bar{\mathfrak{C}}_{\text{red}}$, we then have

$$\bar{\mathfrak{C}}_{\text{red}} = \bar{\bar{T}}_{\text{pre-red}} \bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1), \quad (4.24)$$

since this formulation restricts the evaluation to index combinations with $d'_1 = 0$ or 1. Combining equations (4.20), (4.23), and (4.24), we obtain

$$\begin{aligned} \bar{\mathfrak{Z}}(\hat{r}, \bar{r}'_2)^T \bar{\bar{T}} \bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1) &= \bar{\mathfrak{Z}}(\hat{r}, \bar{r}'_2)^T \bar{\mathfrak{C}} \\ &= \bar{\mathfrak{Z}}(\hat{r}, \bar{r}'_2)^T \bar{\bar{R}}^T \bar{\mathfrak{C}}_{\text{red}} \\ &= \bar{\mathfrak{Z}}(\hat{r}, \bar{r}'_2)^T \bar{\bar{R}}^T \bar{\bar{T}}_{\text{pre-red}} \bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1). \end{aligned} \quad (4.25)$$

By applying the same reasoning to the basis side via reciprocity, the fully reduced expression becomes

$$\bar{\mathfrak{Z}}(\hat{r}, \bar{r}'_2)^T \bar{\bar{T}} \bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1) = \bar{\mathfrak{Z}}(\hat{r}, \bar{r}'_2)^T \bar{\bar{R}}^T \bar{\bar{T}}_{\text{red}} \bar{\bar{R}} \bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1), \quad (4.26)$$

where $\bar{\bar{T}}_{\text{red}} \in \mathbb{C}^{D' \times D'}$ denotes the truncated submatrix of $\bar{\bar{T}}$ restricted to both test- and basis-side indices with $d_1 = 0$ or 1 and $d'_1 = 0$ or 1, and D' , the number of them, is given by

$$D' = \binom{d+2}{2} + \binom{d+1}{2}. \quad (4.27)$$

Moreover, we now define two modified vectors that incorporate the recurrence relationships imposed on the monomial bases as follows:

$$\bar{\mathfrak{Z}}_{\text{mod}}(\hat{r}, \bar{r}'_2) = \bar{\bar{R}} \bar{\mathfrak{Z}}(\hat{r}, \bar{r}'_2), \quad (4.28)$$

$$\bar{\mathfrak{B}}_{\text{mod}}(\hat{r}, \bar{r}'_1) = \bar{\bar{R}} \bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1). \quad (4.29)$$

In (4.28) and (4.29), $\bar{\mathfrak{Z}}_{\text{mod}}$ and $\bar{\mathfrak{B}}_{\text{mod}}$ encapsulate the recurrence relations satisfied by the test and basis monomials, respectively. Substituting into (4.26) yields the final reduced form:

$$\bar{\mathfrak{Z}}(\hat{r}, \bar{r}'_2)^T \bar{\bar{T}} \bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1) = \bar{\mathfrak{Z}}_{\text{mod}}(\hat{r}, \bar{r}'_2)^T \bar{\bar{T}}_{\text{red}} \bar{\mathfrak{B}}_{\text{mod}}(\hat{r}, \bar{r}'_1). \quad (4.30)$$

This reformulation enables the incorporation of the PDE-implied recurrence structure directly into the matrix-vector product, thereby reducing the number of required degrees of freedom while maintaining the fidelity of the functional approximation. Consequently, both storage and computational complexity are

significantly reduced by using the modified vectors and reduced matrix, bringing the overall complexity down to $\mathcal{O}(D'^2) = \mathcal{O}(d^4)$.

Furthermore, the matrices $\bar{\bar{T}}$ and $\bar{\bar{T}}_{\text{red}}$ exhibit a block-Hankel structure that becomes evident when appropriate diagonal scalings and zero-padding are applied. This structural property arises from the coefficient form given in (4.8), where each matrix entry is constructed as

$$c_{d_1+d'_1, d_2+d'_2, d_3+d'_3} \frac{(d_1+d'_1)!}{d_1! d'_1!} \frac{(d_2+d'_2)!}{d_2! d'_2!} \frac{(d_3+d'_3)!}{d_3! d'_3!}. \quad (4.31)$$

To reveal the underlying Hankel structure more clearly, we define a diagonal matrix $\bar{\bar{\mathfrak{D}}}$ whose entries are given by

$$\bar{\bar{\mathfrak{D}}}_{(d_1, d_2, d_3), (d'_1, d'_2, d'_3)} = d_1! d_2! d_3!. \quad (4.32)$$

This matrix serves to normalize the factorial terms appearing in the binomial coefficients inherent in the structure of $\bar{\bar{T}}$ (or $\bar{\bar{T}}_{\text{red}}$). Specifically, consider the transformation given by

$$\bar{\bar{H}} = \bar{\bar{\mathfrak{D}}} \bar{\bar{T}} \bar{\bar{\mathfrak{D}}}. \quad (4.33)$$

Then, each entry of $\bar{\bar{H}}$ takes the form

$$\bar{\bar{H}}_{(d_1, d_2, d_3), (d'_1, d'_2, d'_3)} = c_{d_1+d'_1, d_2+d'_2, d_3+d'_3} (d_1+d'_1)! (d_2+d'_2)! (d_3+d'_3)!, \quad (4.34)$$

which depends only on the sums $d_1 + d'_1$, $d_2 + d'_2$, and $d_3 + d'_3$. As a result, the matrix $\bar{\bar{H}}$ (or its reduced counterpart $\bar{\bar{H}}_{\text{red}}$) exhibits a multidimensional block-Hankel structure, where each entry depends only on the sums $d_1 + d'_1$, $d_2 + d'_2$, and $d_3 + d'_3$. That is, blocks of the matrix are determined by constant total index sums across basis and test monomials. In order to fully realize this block-Hankel form, however, the indexing of both the basis and test monomials must span a complete tensor-product grid (i.e., each index d_i and d'_i must independently range from 0 to d , rather than being constrained by total-degree bounds such as $d_1 + d_2 + d_3 \leq d$). To enforce this structure, the vectors $\bar{\mathfrak{B}}(\hat{r}, \hat{r}'_1)$ and $\bar{\mathfrak{T}}(\hat{r}, \hat{r}'_2)$ must be augmented with zero entries for the missing monomial combinations, and the corresponding rows and columns of $\bar{\bar{H}}$ and $\bar{\bar{H}}_{\text{red}}$ must likewise be padded with zeros. Once this uniform indexing is established, the resulting matrix assumes a Toeplitz-like block-Hankel structure, which enables fast matrix-vector multiplication using

multidimensional Fast Fourier Transforms [53]. As a result, the computational complexity can be reduced from $\mathcal{O}(d^4)$ to $\mathcal{O}(d^2 \log d)$ for $\bar{\bar{T}}_{\text{red}}$ and from $\mathcal{O}(d^6)$ to $\mathcal{O}(d^3 \log d)$ for $\bar{\bar{T}}$, owing to the underlying block-Hankel structure. While such structured matrices may introduce numerical instability, this challenge can be effectively addressed through suitable scaling strategies, as demonstrated in [54].

An example of Taylor expansion with the topics mentioned here is given in the Appendix C.

4.3 Virtual Box Structure

To efficiently compute far-zone box-to-box interactions, the proposed method introduces a *virtual box* structure, illustrated in Fig. 4.3, which is independently studied in [25]. In this setting, the required Taylor expansion order d can be kept fixed across all virtual boxes in all boxes and levels, regardless of the physical box size. Furthermore, in geometrically sparse scenarios, such as quasi-1D elongated objects, the number of non-empty virtual box pairs, and hence the number of required computations for far-zone interactions, is significantly reduced, further lowering the overall computational cost. At each level l of the hierarchical tree, a box of edge length a_l is subdivided into smaller, equally sized virtual boxes, each having an edge length $a_{v,l} \sim \sqrt{a_l}$. This choice is motivated by the Gaussian-wave component of the reduced Green's function, which takes the form

$$\exp\left(-jk\frac{r'^2}{2r}\left[1 - (\hat{r} \cdot \hat{r}')^2\right]\right) = \exp\left(-jk\frac{r'^2}{2r}\left[(\hat{\phi} \cdot \hat{r}')^2 + (\hat{\theta} \cdot \hat{r}')^2\right]\right), \quad (4.35)$$

where the notation follows that of equation (4.4). To maintain a constant Taylor expansion order for this term across all virtual box and box sizes, the ratio $\frac{r'^2}{r}$ must remain constant. This condition leads to the scaling relation $a_{v,l} \sim \sqrt{a_l}$. It is discussed at the end of this section. These virtual boxes serve as localized expansion centers at each box on each level, enabling the outgoing fields to be compactly represented through directional monomials formed via truncated Taylor series expansions. These virtual boxes serve as localized regions for field expansion and enable finer-grained approximation of interactions.

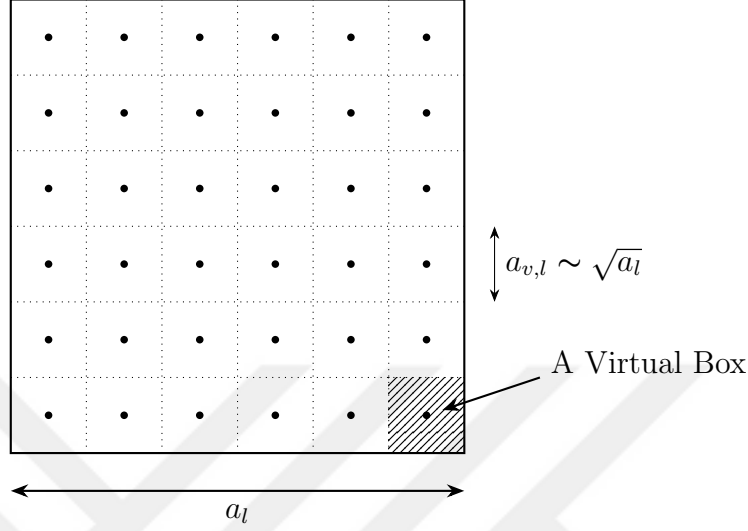
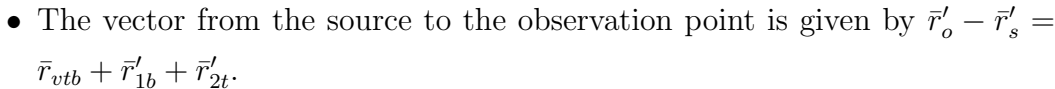


Figure 4.3: Virtual box structure

To evaluate the translation between two boxes in the far zone, the interaction is not computed directly at the box level. Instead, it is decomposed into the cumulative interactions between all pairs of source and observation virtual boxes contained within the two parent boxes. Each of these pairwise interactions is approximated using a truncated Taylor expansion, as detailed in the preceding section. This strategy enables accurate and efficient modeling of the Green's function behavior over spatially separated domains using local expansions around the virtual box centers.

As illustrated in Fig. 4.4, let O denote the origin, let $\bar{r}'_s$ denote a source point located inside the basis virtual box centered at $\bar{r}_{vb}$, and let $\bar{r}'_o$ denote an observation point located within the test virtual box centered at $\bar{r}_{vt}$. Then, according to the representation in (4.4), we aim to compute $g(\bar{r}_{vtb}, \bar{r}'_{1b}, \bar{r}'_{2t})$, where:

- $\bar{r}'_{1b} = \bar{r}_{vb} - \bar{r}'_s$ is the vector from the source point to the center of the basis virtual box.
- $\bar{r}'_{2t} = \bar{r}'_o - \bar{r}_{vt}$ is the vector from the test virtual box to the observation point.
- $\bar{r}_{vtb} = \bar{r}_{vt} - \bar{r}_{vb}$ is the vector from the center of the basis virtual box to the center of the test virtual box. It is called the translation vector.



1. Construct the basis monomial vector $\tilde{\mathbf{B}}(\hat{r}_{vtb}, \bar{r}'_{1b})$ defined in (4.9) on the source (basis) side, where the direction $\hat{r}_{vtb}$ connects the center of the basis virtual box to the center of the test virtual box.
2. Multiply this vector from the right by the matrix $G\bar{T}$, as described in (4.18), to obtain the coefficient vector $\tilde{\mathbf{C}}$ associated with the test monomials.
3. Construct the test monomial vector $\tilde{\mathbf{T}}(\hat{r}_{vtb}, \bar{r}_{2t})$ defined in (4.10) on the observation (test) side.
4. Multiply the coefficient vector from the left by the test monomial vector to evaluate the final interaction, as expressed in (4.12).

If multiple source points reside within the same basis virtual box, let each source point be denoted by its local displacement vector $\bar{r}'_{ib}$ from the center of the basis virtual box, where i indexes the individual source points. For each such point, the corresponding basis monomial vector $\bar{\mathfrak{B}}(\hat{r}_{vtb}, \bar{r}'_{ib})$ is constructed. Due to the linearity of the formulation, the cumulative contribution of all these source points to the observation point can be approximated as

$$\sum_i \beta_i g(\bar{r}_{vtb}, \bar{r}'_{ib}, \bar{r}'_{2t}) \approx G \bar{\mathfrak{T}}(\hat{r}_{vtb}, \bar{r}_{2t})^T \bar{T} \left(\sum_i \beta_i \bar{\mathfrak{B}}(\hat{r}_{vtb}, \bar{r}'_{ib}) \right), \quad (4.36)$$

where β_i are the coefficients associated with the i^{th} source point. We define the combined basis monomial vector as

$$\bar{\mathfrak{B}}_{\text{comb}}(\hat{r}_{vtb}) = \sum_i \beta_i \bar{\mathfrak{B}}(\hat{r}_{vtb}, \bar{r}'_{ib}). \quad (4.37)$$

Using this definition, the interaction in (4.36) becomes

$$\begin{aligned} \sum_i \beta_i g(\bar{r}_{vtb}, \bar{r}'_{ib}, \bar{r}'_{2t}) &\approx G \bar{\mathfrak{T}}(\hat{r}_{vtb}, \bar{r}_{2t})^T \bar{T} \bar{\mathfrak{B}}_{\text{comb}}(\hat{r}_{vtb}) \\ &\approx \bar{\mathfrak{T}}(\hat{r}_{vtb}, \bar{r}_{2t})^T \bar{\mathfrak{C}}_{\text{comb}}(\hat{r}_{vtb}), \end{aligned} \quad (4.38)$$

where the directional combined coefficient vector $\bar{\mathfrak{C}}_{\text{comb}}(\hat{r}_{vtb})$ represents the quantity $G\bar{\mathfrak{C}}$, as defined previously, but evaluated along the specific direction $\hat{r}_{vtb}$. If the objective is to compute the expression given in (4.36), the required modifications to the previously described steps are as follows:

- In step (1), instead of computing $\bar{\mathfrak{B}}(\hat{r}_{vtb}, \bar{r}'_{ib})$ for each source point, we construct the directional combined basis vector $\bar{\mathfrak{B}}_{\text{comb}}(\hat{r}_{vtb})$.
- In step (2), we denote the coefficients as the directional combined coefficient vector $\bar{\mathfrak{C}}_{\text{comb}}(\hat{r}_{vtb})$.

In the T-EFIE formulation, the matrix entries Z_{mn} defined in (2.52) can be transformed into the form [6]

$$\begin{aligned} Z_{mn} = & -jk\eta \int_{S_m} d\bar{r} \bar{t}_m(\bar{r}) \cdot \int_{S_n} d\bar{r}' g(\bar{r}, \bar{r}') \bar{b}_n(\bar{r}') \\ & + \frac{j\eta}{k} \int_{S_m} d\bar{r} \nabla \cdot \bar{t}_m(\bar{r}) \int_{S_n} d\bar{r}' g(\bar{r}, \bar{r}') \nabla' \cdot \bar{b}_n(\bar{r}'), \end{aligned} \quad (4.39)$$

where $\bar{t}_m(\bar{r})$ is the m th test function with support over the surface patch S_m , and $\bar{b}_n(\bar{r})$ is the n th basis function supported on S_n . For RWG basis and test functions, by inserting their definitions from (2.48) and (2.49), (4.39) can be written as:

$$Z_{mn} = -j\omega\mu \sum_{a=1}^2 \sum_{b=1}^2 \frac{l_m l_n}{4A_{am}A_{bn}} (-1)^{a+b} \times \left(\int_{S_m} d\bar{r} (\bar{r} - \bar{r}_{am}) \cdot \int_{S_n} d\bar{r}' (\bar{r}' - \bar{r}_{bn}) g(\bar{r}, \bar{r}') - \frac{4}{k^2} \int_{S_m} d\bar{r} \int_{S_n} d\bar{r}' g(\bar{r}, \bar{r}') \right). \quad (4.40)$$

As seen in (4.40), each matrix entry for a given pair of RWG basis and test functions results in four distinct integrals involving the scalar Green's function $g(\bar{r}, \bar{r}')$. The first three arise from the dot product structure in the first term, while the fourth corresponds to the scalar interaction in the second term. Consequently, in the proposed method, four separate interaction terms β_i are computed for each basis-test pair. These computations are mutually independent and have been parallelized in the numerical implementation to improve efficiency.

In the implementation of this procedure with a single level (this corresponds to the previously proposed method or FMM), for each basis virtual box, their combined basis monomials vector $\bar{\mathfrak{B}}_{\text{comb}}(\hat{r}_{vtb})$ for each virtual box to virtual box translation should be constructed. Similarly, for each test virtual box and its test locations, the test monomials vector $\bar{\mathfrak{T}}(\hat{r}_{vtb}, \bar{r}_{2t})$ should be constructed for each virtual box to virtual box translation. Instead of calculating these vectors in every possible direction $\hat{r}_{vtb}$, we assume that these vectors are precomputed along a sufficiently dense set of $N_{s,l}$ sampling directions $\hat{r}_m$, $m = 1, \dots, N_{s,l}$ at level l . This allows the combined directional basis monomials vector $\bar{\mathfrak{B}}_{\text{comb}}(\hat{r}_{vtb})$ to be efficiently approximated at arbitrary directions $\hat{r}$ via local Lagrange interpolation [6] over the sampled grid. On the test side, a similar strategy is employed: rather than evaluating incoming directional expansions for all possible directions, we use interpolation to project the incoming coefficients ($\bar{\mathfrak{C}}$) onto the same sampling grid. This is conceptually analogous to the approach taken in MLFMA, where interpolation leverages the interpolation grid to reconstruct field contributions efficiently.

For the selection of sampling directions $\hat{r}_m$, where $m = 1, \dots, N_{s,l}$, spherical

Gauss-Legendre quadrature points are employed [6] when the virtual boxes are radiating or receiving energy from all possible directions. In such cases, the number of sampling directions $N_{s,l}$ scales as $\mathcal{O}(a_{v,l}^2)$ due to the need for angular resolution over the entire sphere. However, if the object exhibits lower-dimensional characteristics, the required sampling can be reduced accordingly. For instance, if the object is approximately two-dimensional and lies on a planar surface, it suffices to sample directions along a circle, resulting in $N_{s,l} = \mathcal{O}(a_{v,l})$. Furthermore, if the object is effectively one-dimensional, resembling a linear structure, sampling can be limited to a small set of directions aligned with the object's orientation. In such cases, $N_{s,l}$ remains almost constant, resulting in $\mathcal{O}(1)$.

In this setting, the required degree d of the Taylor expansion is determined by the asymptotic behavior of the reduced Green's function in the far field. Crucially, this degree depends on the ratio $a_{v,l}^2/a_l \sim a_l/a_l \sim 1$, which remains constant across hierarchical levels. Therefore, the expansion order d can be kept fixed as the box size a_l increases with the level. To support this, we consider the series expansion of the reduced Green's function p given by

$$p(\hat{r} \cdot \bar{r}', \hat{\theta} \cdot \bar{r}', \hat{\phi} \cdot \bar{r}') = \sum_{\ell=0}^{\infty} \frac{r'^{\ell}}{r^{\ell}} P_{\ell}(-\hat{r} \cdot \hat{r}') \times \exp \left(-jk \sum_{\ell=1}^{\infty} \frac{r'^{\ell+1}}{(2\ell+1)r^{\ell}} (P_{\ell-1}(-\hat{r} \cdot \hat{r}') - P_{\ell+1}(-\hat{r} \cdot \hat{r}')) \right), \quad (4.41)$$

which arises from the combination of the power series for $1/R$ and the exponential representation of $\exp(-jkR)$. To justify the expansion in (4.41), we examine the analytical structure of the scalar Green's function by expanding both the Euclidean distance $R = |\bar{r} + \bar{r}'|$ and its reciprocal $1/R$, assuming that $r > r'$. The reciprocal $1/R$ admits the classical Legendre expansion

$$\frac{1}{R} = \sum_{\ell=0}^{\infty} \frac{r'^{\ell}}{r^{\ell+1}} P_{\ell}(\cos \gamma), \quad (4.42)$$

which is well-known and can be found in [55]. Similarly, the Euclidean distance itself can be expanded as

$$R = r - \cos \gamma r' + \sum_{\ell=1}^{\infty} \frac{r'^{\ell+1}}{(2\ell+1)r^{\ell}} (P_{\ell-1}(\cos \gamma) - P_{\ell+1}(\cos \gamma)), \quad (4.43)$$

where $\cos \gamma = -\hat{r} \cdot \hat{r}'$, and $P_\ell(\cdot)$ denotes the Legendre polynomial of degree ℓ , bounded on the interval $[-1, 1]$. The derivation of the R expansion is provided in Appendix B. At hierarchical level l , the spatial orders of interest scale as follows:

$$r' \sim a_{v,l} \sim \sqrt{a_l}, \quad r \sim a_l.$$

We define a characteristic ratio:

$$\Delta = \frac{r'^2}{r} \sim \frac{a_{v,l}^2}{a_l} \sim 1, \quad (4.44)$$

which remains constant across hierarchical levels. Focusing on the dominant exponential term for $\ell = 1$ in (4.41), which is the Gaussian-wave component, we obtain:

$$\exp \left(-\frac{jk\Delta}{3} [P_0(-\hat{r} \cdot \hat{r}') - P_2(-\hat{r} \cdot \hat{r}')] \right).$$

Since $\Delta \sim 1$ and the Legendre polynomials are bounded, the argument of the exponential is $\mathcal{O}(1)$. This implies that the required Taylor expansion degree d for this dominant term remains fixed as a_l increases. To analyze the effect of the higher-order remainder terms, we expand their orders explicitly as follows:

$$\begin{aligned} & \exp \left(-j \sum_{\ell=2}^{\infty} \mathcal{O}(a_l^{-\ell/2+0.5}) \right) \sum_{\ell=0}^{\infty} \mathcal{O}(a_l^{-\ell/2}) \\ &= \exp \left(-j \mathcal{O}(a_l^{-0.5}) \right) [1 + \mathcal{O}(a_l^{-0.5})] \\ &= [1 + \mathcal{O}(a_l^{-0.5})] [1 + \mathcal{O}(a_l^{-0.5})] \\ &= 1 + \mathcal{O}(a_l^{-0.5}). \end{aligned} \quad (4.45)$$

This asymptotic behavior confirms that the contribution of remainder terms diminishes with increasing a_l . Therefore, a Taylor expansion with degree d sufficient for finer levels retains its accuracy, or becomes more accurate, at coarser levels. Consequently, using a uniform Taylor degree d across all hierarchical levels ensures stable and consistent approximation accuracy.

4.4 Implementation of the Proposed Method

After the hierarchical subdivision of the computational domain is completed, each box is further partitioned into virtual boxes. These virtual boxes are organized

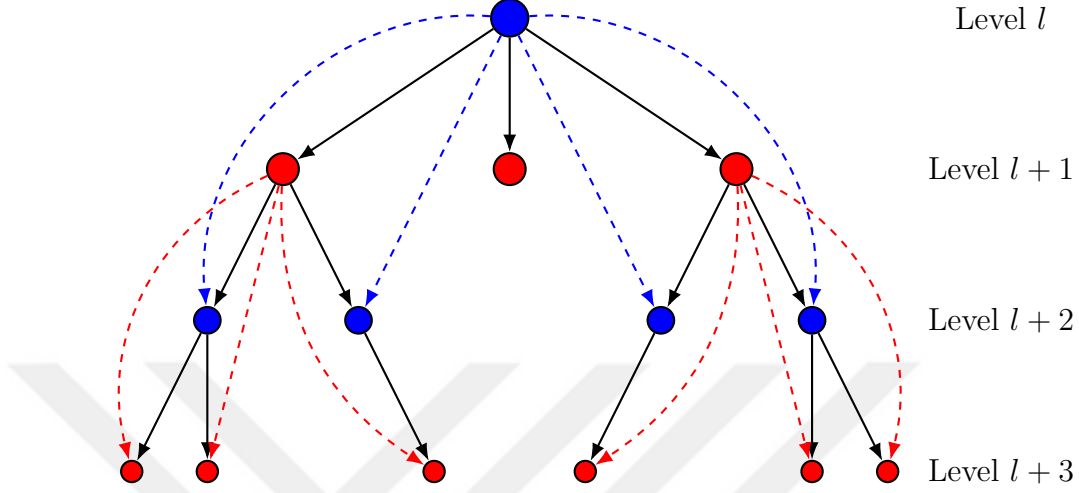


Figure 4.5: Hierarchical tree (in black) and information flow between levels (in red and blue).

such that their relationships are restricted to levels of the same parity; that is, even and odd levels are treated independently. As illustrated in Fig. 4.5, each node represents a box in the hierarchical subdivision of the object. The black arrows indicate the structure of the hierarchical tree, encoding the physical relationships between boxes (i.e., parent-child information). However, for the aggregation and disaggregation stages involving virtual boxes, interactions are organized based on level parity. As a result, the information flow is separated by a two-level difference (highlighted in red for one path and blue for the other). To illustrate this setup, consider the example shown in Fig. 4.6. The j^{th} box at level $l+2$ is subdivided into a $3 \times 3 \times 3$ grid of virtual boxes, ensuring the condition $3a_{v,l+2} = a_{l+2}$. Consequently, the maximum permissible ratio Δ , defined in (4.44), becomes $\Delta = a_{l+2}/9$. To enforce the same Δ in the i^{th} box at level l , which contains the j^{th} box, we require:

$$\Delta = \frac{a_{l+2}}{9} = \frac{a_{v,l}^2}{a_l} = \frac{a_{v,l}^2}{4a_{l+2}} \implies a_{v,l} = \frac{2a_{l+2}}{3} = 2a_{v,l+2} = \frac{a_l}{6}. \quad (4.46)$$

Thus, the i^{th} box at level l must be subdivided into a $6 \times 6 \times 6$ grid of virtual boxes. As can be seen in Fig. 4.6, the virtual boxes at level l (shown in blue) encompass those at level $l+2$. Based on this structure, aggregation and disaggregation operations are defined separately for even and odd levels, since virtual boxes at level l act as parents to those at level $l+2$. Note that this construction cannot be applied at level $l+1$ due to the square-root scaling relationship between physical

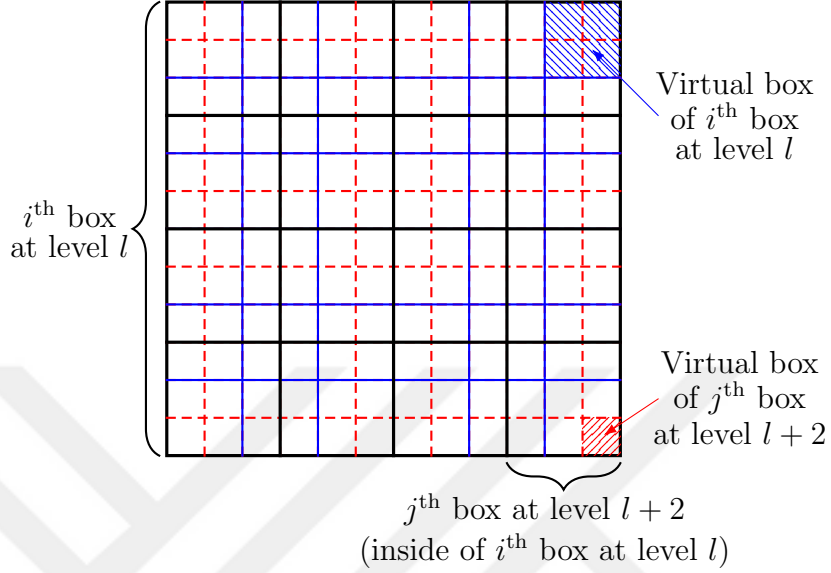


Figure 4.6: An example division of boxes into virtual boxes.

and virtual box sizes. By following this example, as seen in Fig. 4.7, a virtual box at level l serves as a child of a virtual box at level $l - 2$ and a parent of a virtual box at level $l + 2$. Consequently, at consecutive even (or odd) levels, each virtual box at level $l - 2$ may generate up to eight child virtual boxes at level l . This construction adheres to the size definitions of virtual boxes relative to their parent boxes. Specifically, if $a_{v,l}$ denotes the size of a virtual box at level l , then the geometric constraint $a_{v,l} = 2a_{v,l+2}$ holds. This relationship is consistent with the previously defined box sizes, where $a_l = 4a_{l+2}$. As a result, the maximum permissible ratio Δ remains constant across all levels:

$$\Delta = \frac{a_{v,l}^2}{a_l} = \frac{4a_{v,l+2}^2}{4a_{l+2}} = \frac{a_{v,l-2}^2}{a_{l-2}}. \quad (4.47)$$

This invariance ensures that the Taylor expansion degree d can be maintained uniformly throughout the hierarchy. Moreover, due to the separation between even and odd levels, the aggregation and disaggregation stages are performed independently on each parity class. Since these processes are entirely decoupled, they are parallelized in the numerical implementation, as demonstrated in the numerical experiments.

In practice, after the hierarchical division of boxes is completed, the subdivision into virtual boxes begins from the highest levels. At the topmost even and odd

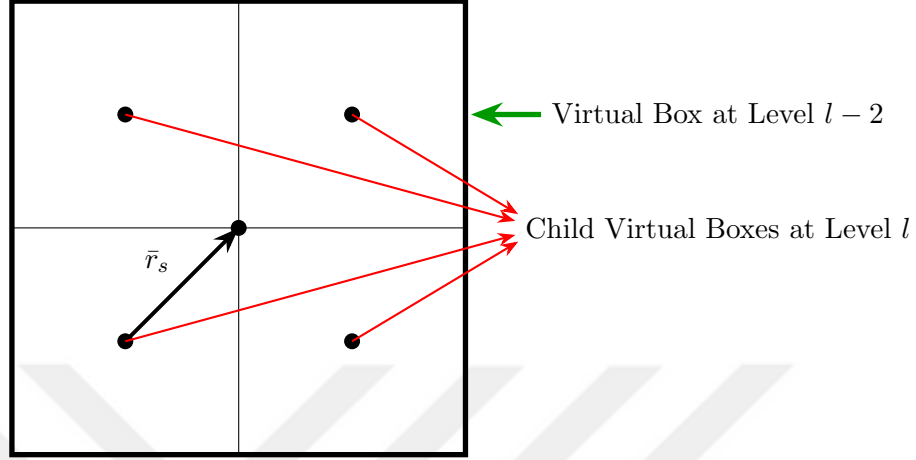


Figure 4.7: The possible child virtual boxes of a virtual box at level l

levels, each box is partitioned into a predefined number of virtual boxes. Then, as the level decreases, the parent virtual boxes corresponding to these children are generated recursively.

The proposed algorithm follows a three-stage hierarchical procedure, *aggregation over virtual boxes*, *translation over virtual boxes*, and *disaggregation over virtual boxes*. It should be noted that the aggregation and disaggregation stages are carried out over virtual boxes. During the translation stage, based on the far-field separation of boxes in the hierarchical tree, each virtual box-to-virtual box translation is computed as a whole to evaluate the corresponding box-to-box interaction.

Let the triple (i, b, l) represent the i^{th} virtual box within the b^{th} basis box at level l . Define $\bar{\mathfrak{B}}_{\text{comb}, i, b, l}(\hat{r}_m)$ as the combined basis monomial vector (radiating coefficients) associated with this virtual box in the direction $\hat{r}_m$. Similarly, let $\bar{\mathfrak{C}}_{\text{comb}, j, t, l}(\hat{r}_m)$ denote the corresponding incoming coefficients for the j^{th} virtual box in the t^{th} test box at level l . These quantities represent the position-dependent forms of the radiating and incoming coefficient vectors introduced in (4.18).

To facilitate aggregation and disaggregation between levels, we define shift matrices $\bar{\bar{S}}(\hat{r}, \bar{r}_s)$ that translate the monomial expansions based on the displacement

vector $\bar{r}_s$ from the center of a child virtual box to the center of its parent. The entries of the shift matrix are given by

$$\bar{S}_{mn} = \begin{cases} \binom{d_1}{d'_1} \binom{d_2}{d'_2} \binom{d_3}{d'_3} z_s'^{d_1-d'_1} y_s'^{d_2-d'_2} x_s'^{d_3-d'_3} & \text{if } d_1 \geq d'_1, d_2 \geq d'_2, d_3 \geq d'_3 \\ 0 & \text{otherwise} \end{cases}, \quad (4.48)$$

where the projections of the displacement vector are defined as

$$z'_s = \hat{r} \cdot \bar{r}_s, \quad y'_s = \hat{\theta} \cdot \bar{r}_s, \quad x'_s = \hat{\phi} \cdot \bar{r}_s.$$

In (4.48), the multi-index m corresponds to the ordering of the (d_1, d_2, d_3) triple in block-major fashion, where d_3 varies fastest, followed by d_2 , and d_1 varies slowest; similarly, n corresponds to the ordered index for (d'_1, d'_2, d'_3) .

Multiplying this shift matrix with a monomial vector evaluated at a shifted origin yields the corresponding expansion at the new location:

$$\bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1 + \bar{r}_s) = \bar{S}(\hat{r}, \bar{r}_s) \bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1), \quad (4.49)$$

which is valid for all types of $\bar{\mathfrak{B}}$ and $\bar{\mathfrak{T}}$ vectors. This identity follows from the binomial expansion:

$$\begin{aligned} & (z'_s + z'_1)^{d_1} (y'_s + y'_1)^{d_2} (x'_s + x'_1)^{d_3} \\ &= \sum_{d'_1=0}^{d_1} \sum_{d'_2=0}^{d_2} \sum_{d'_3=0}^{d_3} \binom{d_1}{d'_1} \binom{d_2}{d'_2} \binom{d_3}{d'_3} z_s'^{d_1-d'_1} y_s'^{d_2-d'_2} x_s'^{d_3-d'_3} z_1'^{d'_1} y_1'^{d'_2} x_1'^{d'_3}. \end{aligned} \quad (4.50)$$

4.4.1 Aggregation over Virtual Boxes

In the aggregation stage, the combined basis monomial vectors $\bar{\mathfrak{B}}_{\text{comb},i,b,l}(\hat{r}_m)$ are computed for all virtual boxes at each level and propagated from finer to coarser levels in the hierarchy, for every sampling direction $\hat{r}_m$.

At the highest even and odd levels, these vectors are initialized directly for each virtual box by summing the local basis monomial vectors $\bar{\mathfrak{B}}(\hat{r}_m, \bar{r}'_n)$ associated with all subdomain basis functions n contained in the virtual box. This initial computation is defined in (4.37).

At lower levels, the combined basis monomial vectors of child virtual boxes are aggregated at the center of their parent virtual box using shift matrices. Specifically, for each virtual box (i, b, l) and sampling direction $\hat{r}_m$, the aggregation is performed as

$$\begin{aligned} \bar{\mathfrak{B}}_{\text{comb},i,b,l}(\hat{r}_m) &= \sum_{(i',b',l+2) \in \mathcal{C}_{i,b,l}} \bar{S}(\hat{r}_m, \bar{r}_s) \mathcal{I}(\bar{\mathfrak{B}}_{\text{comb},i',b',l+2}(\hat{r}_n), \hat{r}_m), \\ \forall m &\in \{1, \dots, N_{s,l}\}, \end{aligned} \quad (4.51)$$

where $\mathcal{C}_{i,b,l}$ denotes the set of child virtual boxes $(i', b', l+2)$ corresponding to the parent virtual box (i, b, l) , and $\bar{r}_s$ is the vector from the center of the child virtual box to the center of the parent. The function $\mathcal{I}(\cdot, \hat{r}_m)$ denotes local Lagrange interpolation of the directional vector $\bar{\mathfrak{B}}_{\text{comb},i',b',l+2}$ evaluated at direction $\hat{r}_m$ using the values at the sampling directions $\{\hat{r}_n\}$ available at level $l+2$. That is,

$$\mathcal{I}(\bar{\mathfrak{B}}_{\text{comb},i',b',l+2}(\hat{r}_n), \hat{r}_m) \approx \bar{\mathfrak{B}}_{\text{comb},i',b',l+2}(\hat{r}_m). \quad (4.52)$$

This interpolation is necessary because the sampling directions $\{\hat{r}_m\}$ at level l may differ from those at level $l+2$, and ensures that the aggregation can be performed coherently in the directional domain across levels.

4.4.2 Translation over Virtual Boxes

In the translation stage, for each far-field pair of basis and test boxes (b, t) at a given level l , the radiating basis monomial vectors of the virtual boxes in the basis box b are translated into incoming coefficient vectors for the virtual boxes in the test box t . This operation is performed directionally for each required translation vector $\hat{r}$ and is expressed as

$$\bar{\mathfrak{C}}_{\text{comb},j,t,l}(\hat{r}_m) = \sum_{(b,l) \in \mathcal{F}_{t,l}} \sum_{i \in \mathcal{V}_{b,l}} \mathcal{A} \left(G\bar{T} \mathcal{I}(\bar{\mathfrak{B}}_{\text{comb},i,b,l}(\hat{r}_n), \hat{r}), \hat{r}_m \right), \quad (4.53)$$

where $\mathcal{F}_{t,l}$ denotes the set of far-field basis boxes associated with test box t at level l , while $\mathcal{V}_{b,l}$ denotes the set of virtual box indices within the basis box, respectively. The vector $\bar{r}$ represents the translation vector from the center of virtual box (i, b, l) to the center of virtual box (j, t, l) , and the translation matrix $G\bar{T}$ is constructed accordingly.

As discussed in Section 4.3, the radiating basis monomial vector $\bar{\mathfrak{B}}_{\text{comb},i,b,l}(\hat{r})$ is evaluated via local Lagrange interpolation using its values at the sampling directions $\{\hat{r}_n\}$. The interpolation operator $\mathcal{I}$ performs this evaluation by interpolating the directional values of the radiating vector on the sample grid to a target direction $\hat{r}$, that is,

$$\mathcal{I}(\bar{\mathfrak{B}}_{\text{comb},i,b,l}(\hat{r}_n), \hat{r}) \approx \bar{\mathfrak{B}}_{\text{comb},i,b,l}(\hat{r}). \quad (4.54)$$

Following interpolation and translation, the antepolation operator $\mathcal{A}$ projects the result onto the incoming coefficient basis corresponding to the direction $\hat{r}_m$. Specifically, if $\bar{v}$ denotes the translated vector obtained from $G\bar{T}\mathcal{I}(\cdot)$, then

$$\mathcal{A}(\bar{v}, \hat{r}_m) \mapsto \bar{\mathfrak{C}}_{\text{comb},j,t,l}(\hat{r}_m), \quad (4.55)$$

where $\bar{\mathfrak{C}}_{\text{comb},j,t,l}(\hat{r}_m)$ is the resulting incoming coefficient vector for the test virtual box (j, t, l) at sampling direction $\hat{r}_m$. The contributions from all interacting basis virtual boxes are summed at each of these sampling directions.

4.4.3 Disaggregation over Virtual Boxes

In the disaggregation stage, the incoming coefficient vectors $\bar{\mathfrak{C}}_{\text{comb},j,t,l}(\hat{r}_m)$ are computed for all virtual boxes at each level and propagated from coarser to finer levels in the hierarchy, for every sampling direction $\hat{r}_m$.

At lower levels, the incoming coefficient vectors of parent virtual boxes are disaggregated to their child virtual boxes using shift matrices. Specifically, for each virtual box (i, b, l) and sampling direction $\hat{r}_m$, the disaggregation is performed as

$$\begin{aligned} \bar{\mathfrak{C}}_{\text{comb},i,b,l}^+(\hat{r}_m) &= \bar{\mathfrak{C}}_{\text{comb},i,b,l}(\hat{r}_m) + \bar{S}^T(\hat{r}_m, \bar{r}_s) \mathcal{A}(\bar{\mathfrak{C}}_{\text{comb},i',b',l+2}^+(\hat{r}_n), \hat{r}_m), \\ &\forall m \in \{1, \dots, N_{s,l}\}, \end{aligned} \quad (4.56)$$

where $(i, b, l) \in \mathcal{C}_{i',b',l+2}$, $\bar{r}_s$ is the vector from the center of the parent virtual box to the center of the child, and $\bar{\mathfrak{C}}_{\text{comb},i,b,l}^+$ denotes the combined incoming coefficient vector in which the contributions from lower levels have been accumulated.

Similar to the aggregation stage, the operator $\mathcal{A}$ denotes the directional anteroposition, which maps the incoming coefficients from the sampling grid at level $l+2$ to those at level l .

It is important to note that in (4.56), the transpose $\bar{\bar{S}}^T$ of the shift matrix appears instead of $\bar{\bar{S}}$. This is because disaggregation performs the inverse operation of aggregation: while aggregation maps basis functions to coefficient vectors (i.e., projection), disaggregation maps coefficients back to test functions (i.e., reconstruction). As a result, the transpose of the shift operator is used to reverse the direction of the projection.

At the highest even and odd levels, the resulting incoming coefficient vectors are finally contracted with the local test monomial vectors $\bar{\mathfrak{T}}(\hat{r}_m, \bar{r}'_n)$ associated with the n^{th} test functions contained in each virtual box. This dot product yields the total far-field interactions for the corresponding test functions.

4.5 Memory and Time Complexity of the Proposed Method

For the SIE problems, the memory and time complexity of the proposed method are as follows. Let us denote the number of boxes at level l as $N_{B,l}$. Since we are dealing with a surface integral equation (SIE) problem, we have $N_{B,l} \sim 4N_{B,l-1}$ due to the two-dimensional nature of surface discretizations. However, if the object is quasi-1D, then $N_{B,l} \sim 2N_{B,l-1}$, as boxes grow linearly along the object's elongated axis. Additionally, for the highest level L , the number of boxes times box size is proportional to the number of unknowns as $N_{B,L} \cdot a_L \sim N$ in quasi-1D and $N_{B,L} \cdot a_L^2 \sim N$ in quasi-2D or 3D cases.

Furthermore, at each level l , the average number of virtual boxes per parent box, denoted $N_{v,l}$, is on the order of a_l , where a_l is the number of unknowns per box at level l . This holds for general surface (3D or 2D) problems. However, for quasi-1D objects, we have $N_{v,l} \sim \sqrt{a_l}$, because only a subset of virtual boxes (in

a linear pattern) are populated due to the object's geometry.

For the translation stage, each box-to-box translation involves all virtual boxes in the source and test boxes, leading to $N_{v,l}^2$ virtual box interactions. Therefore, the time complexity of the translation stage is $\mathcal{O}(N_{B,l} \cdot N_{v,l}^2)$ per level in the general case. In terms of memory complexity, each sample direction at each virtual box must be stored. Thus, the total memory required for translation is proportional to $\mathcal{O}(N_{B,l} \cdot N_{v,l} \cdot N_{s,l})$, where $N_{s,l}$ denotes the number of sample directions at level l .

For the aggregation and disaggregation stages at level l , each sample direction at each virtual box is used either in interpolation (aggregation) or antepolation (disaggregation). Therefore, both stages have the same time and memory complexity as translation memory complexity, which is $\mathcal{O}(N_{B,l} \cdot N_{v,l} \cdot N_{s,l})$.

Depending on the object's geometry, the number of sample directions $N_{s,l}$ behaves differently. As explained in Section 4.3:

- For fully 3D surfaces (e.g., spheres): $N_{s,l} = \mathcal{O}(a_{v,l}^2) = \mathcal{O}(a_l)$
- For quasi-2D surfaces (e.g., flat plates): $N_{s,l} = \mathcal{O}(a_{v,l}) = \mathcal{O}(\sqrt{a_l})$
- For quasi-1D surfaces (e.g., wires): $N_{s,l} = \mathcal{O}(1)$

Then, for each case, general 3D surface, quasi-2D surface, and quasi-1D surface, we have the following complexity results:

General 3D Surface.

- $N_{B,l} \sim 4^l$, $a_l \sim 2^{-l}$, $N_{v,l} \sim a_l$, $N_{s,l} \sim a_l$
- Time complexity: $\mathcal{O}(\sum_l N_{B,l} \cdot N_{v,l}^2 + N_{B,l} \cdot N_{v,l} \cdot N_{s,l}) = \mathcal{O}(\sum_l N_{B,l} \cdot a_l^2)$, and by using equality $N_{B,l} a_l^2 \sim N_{B,l+1} a_{l+1}^2$ and as the number of levels L is approximately $\log N$, we have $= L \cdot \mathcal{O}(N_{B,L} \cdot a_L^2) = N \log N$.

- Memory complexity: $\mathcal{O}(\sum_l N_{B,l} \cdot N_{v,l} \cdot N_{s,l})$ and similar to time complexity, we have $= N \log N$.

Quasi-2D Surface.

- $N_{B,l} \sim N \cdot 4^l$, $a_l \sim 2^{-l}$, $N_{v,l} \sim a_l$, $N_{s,l} \sim a_l^{1/2}$
- Time complexity: $\mathcal{O}(\sum_l N_{B,l} \cdot N_{v,l}^2 + N_{B,l} \cdot N_{v,l} \cdot N_{s,l})$ is asymptotically equal to $\mathcal{O}(\sum_l N_{B,l} \cdot (a_l^2 + a_l^{3/2}))$, and by using the equality $N_{B,l} a_l^2 \sim N_{B,l+1} a_{l+1}^2$ and noting that the number of levels L is approximately $\log N$, we have $L \cdot \mathcal{O}(N_{B,L} \cdot a_L^2)$, which is equal to $\mathcal{O}(N \log N)$, similar to the 3D case.
- Memory complexity: $\mathcal{O}(\sum_l N_{B,l} \cdot N_{v,l} \cdot N_{s,l})$ and similar to time complexity, we have $= N \log N$.

Quasi-1D Surface.

- $N_{B,l} \sim N \cdot 2^l$, $a_l \sim 2^{-l}$, $N_{v,l} \sim a_l^{1/2}$, $N_{s,l} \sim \mathcal{O}(1)$
- Time complexity: $\mathcal{O}(\sum_l N_{B,l} \cdot N_{v,l}^2 + N_{B,l} \cdot N_{v,l} \cdot N_{s,l}) = \mathcal{O}(\sum_l N_{B,l} \cdot a_l)$. By using the equality $N_{B,l} a_l \sim N_{B,l+1} a_{l+1}$, and noting that the number of levels L is approximately $\log N$, we have $= L \cdot \mathcal{O}(N_{B,L} \cdot a_L)$. Since $N_{B,L} \cdot a_L \sim N$, the overall complexity becomes $\mathcal{O}(N \log N)$.
- Memory complexity: $\mathcal{O}(\sum_l N_{B,l} \cdot N_{v,l} \cdot N_{s,l})$ and similar to time complexity, we have $= N \log N$.

It should be emphasized that, although the proposed method retains $\mathcal{O}(N \log N)$ complexity, the constant hidden in the complexity expression differs from that of MLFMA in both 3D and 2D cases. In particular, for the 3D case, the computational cost of multiplying the $\bar{T}$ and $\bar{S}$ matrices scales as $\mathcal{O}(d^6)$ in the worst-case scenario when no further acceleration techniques are applied. Moreover, achieving uniform sampling over the unit sphere necessitates a relatively large number of directions, further increasing the overall computational

burden. As a result, the constant factor associated with the proposed method in 3D is most probably higher than that of MLFMA. In contrast, the 2D case requires significantly fewer sampling directions on the unit circle, leading to a coefficient comparable to that of MLFMA. This observation is also supported by the numerical experiments presented in this work.

4.6 Numerical Results

Numerical results for PEC objects discretized using RWG basis functions are presented to assess the performance of the proposed method. In all examples, the T-EFIE is employed, and the resulting systems are solved using the GMRES iterative solver without preconditioning. The solver parameters are set as follows: a relative error tolerance of 10^{-3} , a restart value of 100, and a maximum of 1000 iterations. A second-order Gaussian quadrature rule is applied to evaluate the integrals over the surfaces of triangles associated with basis functions, whereas a first-order Gaussian quadrature is used for testing function-related triangles. The scattering problems considered in this study involve a PEC dipole scatterer and a PEC plate, both illuminated by a unit-amplitude (1 V/m), x -polarized uniform plane wave propagating in the $+z$ direction. Parallelization is applied to both MLFMA and the proposed method in all numerical experiments. As discussed in Section 4.4, the independent processing of odd and even levels in the proposed algorithm provides an intrinsic parallelization structure that is not present in MLFMA. Furthermore, as described in Section 4.3, the computation of each matrix element Z_{mn} in the proposed method involves four distinct integrals related to the Green's function, while MLFMA requires two such integrals over the angular variables (θ, ϕ) . Accordingly, the proposed method is executed using 8 CPU cores, whereas MLFMA utilizes 2 cores. Furthermore, each solution is evaluated under two configurations in the proposed method. In the first configuration, the reduced Green's function is computed using sparse matrix multiplications involving the matrix $\bar{\bar{T}}$ in (4.12), based on the formulation in (4.6). In the second configuration, the same function is obtained via direct (dense) matrix multiplications with $\bar{\bar{T}}$, following the formulation in (4.7). It should be highlighted that

the acceleration techniques discussed in Section 4.2 are not implemented in the following numerical experiments. All CPU time measurements are performed on an Intel Xeon Gold 6230 processor and reported as the average of 10 independent runs. Memory usage is not reported. Throughout this section, computation times are reported using the abbreviations NF (near-field) and FF (far-field) to denote the corresponding interaction stages.

A PEC dipole scatterer (a quasi-one-dimensional elongated object) with dimensions $(8, 0.04, 0.04) \times \lambda$ at 118.812 MHz, discretized using 38,592 RWG basis functions, is considered as the first example. The separation between the two rods of the dipole scatterer is set to 0.8λ . In the first and second configurations of the proposed method, the hierarchical level is chosen as $L = 6$. The box sizes used in the hierarchy are $\lambda/4$, $\lambda/2$, λ , and 2λ , each of which is further divided into virtual boxes with sizes $\lambda/8$, $\lambda/4$, $\lambda/4$, and $\lambda/2$, respectively. The non-empty virtual boxes are aligned along the direction of the object, resulting in a sparse virtual box grid that exploits the geometric structure of the problem. In the first configuration, the Taylor expansion degree is set to $d = 6$, whereas in the second configuration, it is set to $d = 4$. Moreover, only the points in the $\pm x$ directions are included in the sampling grid, as interactions between virtual boxes are performed exclusively along these directions. The two configurations produced almost identical results; therefore, only the far-zone results from the first configuration are presented. Fig. 4.8 presents the far-zone scattered electric field computed using the proposed method and compares it with the results of a 6-level MLFMA implementation. The agreement between the two methods is excellent, validating the accuracy of the proposed approach. The corresponding CPU time results are provided in Table 4.1.

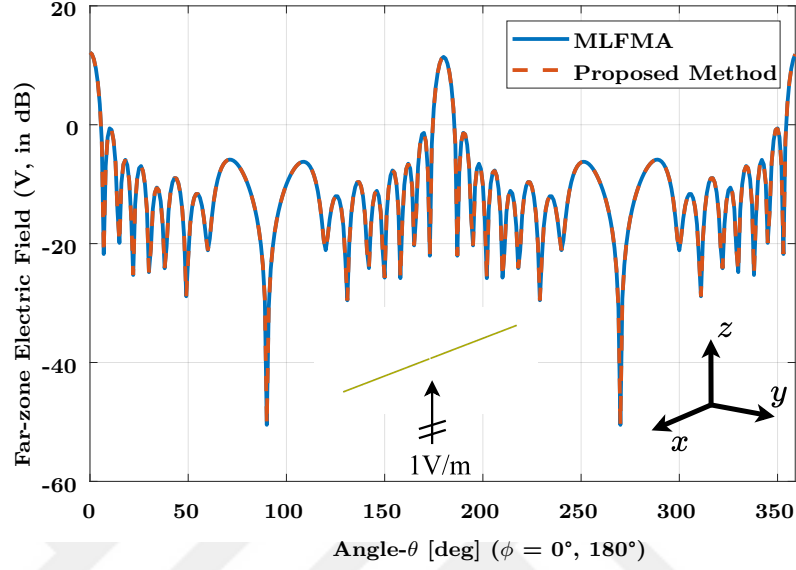


Figure 4.8: Far-zone co-polarized electric fields scattering from a PEC dipole scatterer with dimensions $(8, 0.04, 0.04) \times \lambda$ illuminated with a unit amplitude uniform plane wave at 118.812 MHz. The discretized surface consists of 38,592 basis functions.

Table 4.1: The CPU Time Results for dipole scatterer. The L value is 6 and the same in both MLFMA and the proposed method.

PEC Dipole	6-Level MLFMA	Proposed Method 1st Configuration	Proposed Method 2nd Configuration
One MVM (FF) CPU Time (sec.)	2.6	0.75	0.44
One MVM (NF) CPU Time (sec.)	4.02		
Iteration Count	798	797	798
Total Time (Hour)	1.47	1.06	0.99

Based on the results in Table 4.1, the proposed method achieves faster FF interaction computations compared to MLFMA. Although the typical performance degradation of MLFMA in elongated structures is not strongly observed in these electrical sizes, the proposed method still accelerates FF calculations by

approximately 3.5 times and 5.9 times relative to MLFMA in the first and second configurations, respectively. Furthermore, the second configuration appears to be more effective than the first, as it achieves the same level of accuracy with a lower Taylor expansion degree d . Although the first method for computing the $\bar{\bar{T}}$ matrix (used in the first configuration) is computationally faster, it requires a higher expansion order to match the accuracy of the second method, which increases computation.

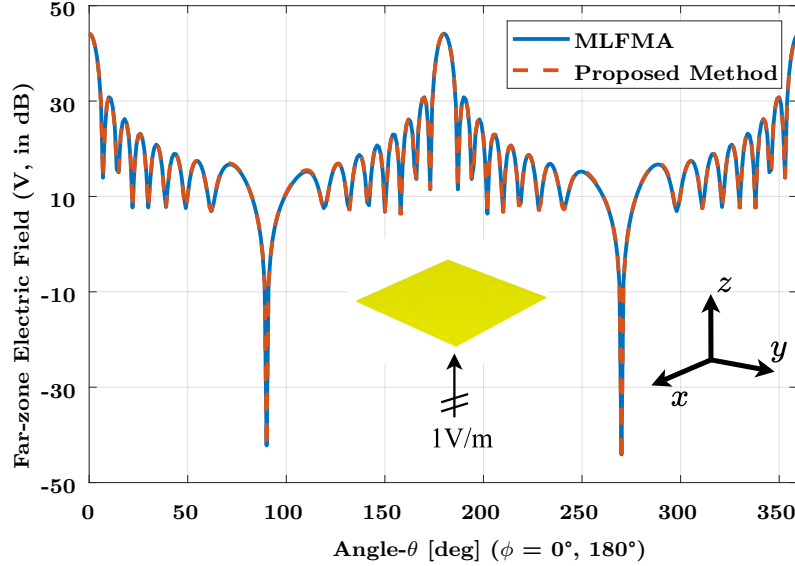


Figure 4.9: Far-zone co-polarized electric fields scattering from a PEC plate with dimensions $(8, 8, 0.08) \times \lambda$ illuminated with a unit amplitude uniform plane wave at 120 MHz. The discretized surface consists of 183,060 basis functions.

The next example is a quasi-2D object: a PEC plate with dimensions $(8, 8, 0.08) \times \lambda$ at 120 MHz, discretized using 183,060 RWG basis functions. In the proposed method, the hierarchical level is set to $L = 6$. The box sizes used in the hierarchy are $\lambda/4$, $\lambda/2$, λ , and 2λ , each of which is further subdivided into virtual boxes under two different configurations. In the first configuration, the virtual box sizes are $\lambda/8$, $\lambda/4$, $\lambda/4$, and $\lambda/2$, respectively. The Taylor expansion degree is set to $d = 7$. In the second configuration, the virtual box sizes are $\lambda/4$, $\lambda/2$, $\lambda/2$, and λ , respectively. Here, the Taylor expansion degree is set to $d = 9$ for odd levels and $d = 8$ for even levels. Moreover, only the points on

the unit circle within the unit sphere are included in the sampling grid, as interactions between virtual boxes are performed solely along these directions. The two configurations produced almost identical results; therefore, only the far-zone and surface current distribution results from the first configuration are presented. Figure 4.9 shows the far-zone scattered electric field obtained using the proposed method and compares it with the results of the 6-level MLFMA implementation. A comparison of the surface current densities is provided in Fig. 4.10. The results demonstrate excellent agreement between the two methods, confirming the accuracy of the proposed approach. The corresponding CPU time results are summarized in Table 4.2.

Table 4.2: CPU Time Results for PEC plate. The L value is the same in MLFMA and the proposed method.

PEC Plate	6-Level MLFMA	Proposed Method 1st Configuration	Proposed Method 2nd Configuration
One MVM (FF) CPU Time (sec.)	12.2	20.46	12.1
One MVM (NF) CPU Time (sec.)	12		
Iteration Count	642		
Total Time (Hour)	4.32	5.79	4.3

Based on the results in Table 4.2, the proposed method achieves nearly the same FF interaction computation times as MLFMA when the second configuration is used. Its performance is superior to that of the first configuration, consistent with the previous numerical example. For approximately the same Taylor order d , the second configuration allows for larger virtual box sizes than the first, resulting in fewer virtual box-to-virtual box interactions and improved overall efficiency. It is worth highlighting that, in the second configuration, the virtual box sizes are equal to the parent box sizes at levels 5 and 6. This is a reasonable choice, as the proposed method tends to perform better with electrically large boxes. In contrast, for mid- or low-level boxes that are electrically small, the contribution of the $1/R$ term becomes dominant. In such cases, computing the

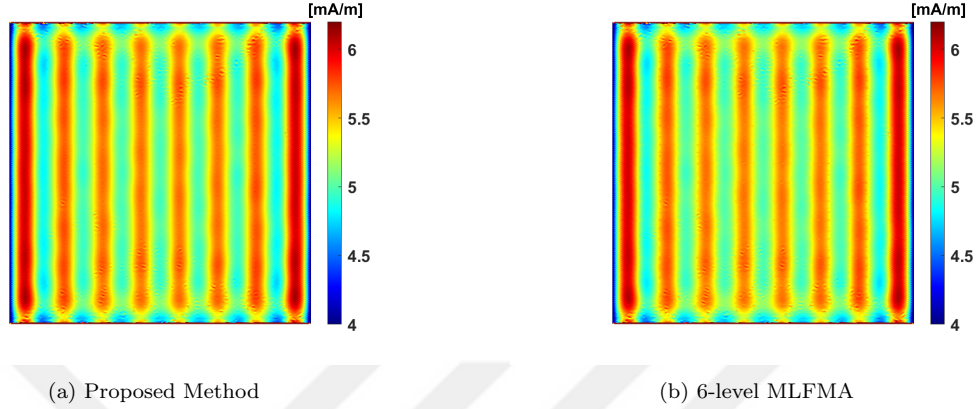


Figure 4.10: Comparison of the induced current density magnitudes for the (a) proposed method and (b) 6-level MLFMA, when the PEC plate is illuminated with a unit amplitude uniform plane wave at 120 MHz.

interaction using a single box, rather than dividing it into virtual boxes, becomes more favorable in terms of both accuracy and efficiency. Furthermore, since the object in this example is quasi-2D, MLFMA does not exhibit significant performance degradation. Consequently, the proposed method reaches the performance level of MLFMA in this case.

As a result, the numerical experiments demonstrate that the proposed method performs significantly better than MLFMA in the case of the quasi-1D elongated object. Theoretically, the proposed method exhibits $\mathcal{O}(N \log N)$ time complexity, and although MLFMA also achieves similar complexity with slight degradation at these box sizes, the constant factor associated with $N \log N$ is substantially smaller in the proposed method. This leads to a clear computational advantage in the elongated case, where faster computation is achieved while maintaining high accuracy in the results. In the quasi-2D case, where MLFMA is known to perform well due to the absence of directional sparsity, the proposed method still reaches comparable performance. Notably, the coefficient of the $\mathcal{O}(N \log N)$ complexity in the proposed method becomes nearly equal to that of MLFMA, confirming the method's robustness in the 2D case as well, again with no compromise in accuracy.

Chapter 5

Conclusion and Future Work

This thesis has presented two novel methods for the fast and accurate solution of electromagnetic IEs, motivated by the increasing computational challenges associated with multiscale and electrically large scattering problems.

In the first part, an ML-accelerated approach has been introduced for the efficient evaluation of far-zone interactions in frequency-domain IEs. The method is based on a group-by-group interaction framework, where far-field MVMs are computed using neural networks trained on the Green's function of Hertzian dipoles. This formulation provides high accuracy, strong scalability, and robust performance across a wide frequency range. Importantly, the method avoids the LFB problem that commonly limits the performance of conventional MLFMA implementations. The use of uniformly sized boxes and fixed uniform basis functions ensures excellent workload balance in parallel environments. Although the current implementation is single-level, a multilevel extension is under active development. Such an extension is expected to require more expressive neural architectures, such as Kolmogorov-Arnold Networks (KAN) [56], to maintain accuracy and efficiency across hierarchical levels. Moreover, adapting the uniform basis layout and preserving parallel efficiency at each level will be crucial. Despite these challenges, the regular and symmetric structure of the method suggests that it can achieve better load balancing than MLFMA and other hierarchical techniques

in the literature.

In the second part, a Taylor expansion and virtual box-based hierarchical method has been developed, particularly tailored for electrically large and elongated scatterers. While adopting the same hierarchical tree structure as MLFMA, the method combines two key ideas: subdividing each box into smaller virtual boxes, and evaluating far-zone interactions via Taylor expansions of the reduced Green’s function (excluding the plane-wave component). Each box of size a is partitioned into virtual boxes of size $\sqrt{a}$, allowing the expansion order to remain constant as the parent box size increases. This ensures the method maintains $\mathcal{O}(N \log N)$ complexity, even in scenarios where MLFMA’s efficiency deteriorates due to directional sparsity loss. The absence of matrix decompositions and the algebraic nature of the Taylor terms also allow for fast implementations, where many operations can be expressed as multidimensional convolutions. These can be further accelerated using FFT-based schemes, which represent a promising direction for future work. Additionally, optimizing the sampling grid for each box (especially in an adaptive or direction-dependent manner) can reduce interpolation cost and memory consumption. As discussed in Section 4.3, the method performs increasingly well as the electrical size grows. Therefore, in low-frequency regimes, alternative formulations that can be combined with this approach beyond a certain threshold may be investigated to ensure broadband applicability.

In summary, the two proposed methods provide robust, scalable, and accurate alternatives to MLFMA for solving frequency-domain IEs. By addressing different aspects of the problem, one through ML and the other through hierarchical and algebraic expansions, the techniques introduced in this thesis provide novel, efficient, and accurate solvers in computational electromagnetics.

Bibliography

- [1] R. F. Harrington and J. L. Harrington, *Field computation by moment methods*. Oxford University Press, Inc., 1996.
- [2] W. C. Chew, J. Jin, E. Michielssen, and J. Song, eds., *Fast and Efficient Algorithms in Computational Electromagnetics*. Artech House Antennas and Propagation Library, Norwood, MA: Artech House, 2001.
- [3] J. Jin, *The Finite Element Method in Electromagnetics*. Hoboken, NJ: John Wiley & Sons, 3rd ed., 2014.
- [4] W. C. Gibson, *The Method of Moments in Electromagnetics*. Boca Raton, FL: Chapman & Hall/CRC, 3rd ed., 2021.
- [5] A. Taflové and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Norwood, MA: Artech House, 3rd ed., 2005.
- [6] Ö. Ergül and L. Gürel, *Multilevel Fast Multipole Algorithm (MLFMA) for Solving Large Scale Computational Electromagnetics Problems*. Hoboken, NJ, USA: Wiley-IEEE, 2014.
- [7] R. Coifman, V. Rokhlin, and S. Wandzura, “The fast multipole method for the wave equation: a pedestrian prescription,” *IEEE Antennas and Propagation Magazine*, 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 transactions on antennas and propagation*, vol. 45, no. 10, pp. 1488–1493, 1997.

- [9] M. Takrimi, Ö. Ergül, and V. B. Ertürk, “Incomplete-leaf multilevel fast multipole algorithm for multiscale penetrable objects formulated with volume integral equations,” *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 9, pp. 4914–4918, 2017.
- [10] B. Khalichi, Ö. Ergül, M. Takrimi, and V. B. Ertürk, “Broadband analysis of multiscale electromagnetic problems: Novel incomplete-leaf MLFMA for potential integral equations,” *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 12, pp. 9032–9037, 2021.
- [11] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, “AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems,” *Radio Science*, vol. 31, no. 5, pp. 1225–1251, 1996.
- [12] M. Yurkin and A. Hoekstra, “The discrete dipole approximation: An overview and recent developments,” *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 106, no. 1, pp. 558–589, 2007. IX Conference on Electromagnetic and Light Scattering by Non-Spherical Particles.
- [13] Y.-N. Liu, X.-M. Pan, and X.-Q. Sheng, “Skeletonization accelerated MLFMA solution of volume integral equation for plasmonic structures,” *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 3, pp. 1590–1594, 2018.
- [14] S. Velamparambil and W. C. Chew, “Analysis and performance of a distributed memory multilevel fast multipole algorithm,” *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 8, pp. 2719–2727, 2005.
- [15] Ö. Ergül and L. Gürel, “A hierarchical partitioning strategy for an efficient parallelization of the multilevel fast multipole algorithm,” *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 6, pp. 1740–1750, 2009.
- [16] L. Greengard, J. Huang, V. Rokhlin, and S. Wandzura, “Accelerating fast multipole methods for the Helmholtz equation at low frequencies,” *IEEE Computational Science and Engineering*, vol. 5, no. 3, pp. 32–38, 1998.

- [17] J. S. Zhao and W. C. Chew, “Applying LF-MLFMA to solve complex PEC structures,” *Microwave and Optical Technology Letters*, vol. 28, no. 3, pp. 155–160, 2001.
- [18] Y.-H. Chu and W. C. Chew, “A multilevel fast multipole algorithm for electrically small composite structures,” *Microwave and Optical Technology Letters*, vol. 43, no. 3, pp. 202–207, 2004.
- [19] V. Melapudi, B. Shanker, S. Seal, and S. Aluru, “A scalable parallel wide-band MLFMA for efficient electromagnetic simulations on large scale clusters,” *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 7, pp. 2565–2577, 2011.
- [20] L. Jiang and W. Chew, “Low-frequency fast inhomogeneous plane-wave algorithm (LF-FIPWA),” *Microwave and Optical Technology Letters*, vol. 40, no. 2, pp. 117–122, 2004.
- [21] I. Bogaert, J. Peeters, and F. Olyslager, “A nondirective plane wave MLFMA stable at low frequencies,” *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 12, pp. 3752–3767, 2008.
- [22] D. T. Schobert and T. F. Eibert, “Low-frequency surface integral equation solution by multilevel Green’s function interpolation with fast Fourier transform acceleration,” *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 3, pp. 1440–1449, 2012.
- [23] M. Kalfa, Ö. Ergül, and V. B. Ertürk, “Multiple-precision arithmetic implementation of the multilevel fast multipole algorithm,” *IEEE Transactions on Antennas and Propagation*, vol. 72, no. 1, pp. 11–21, 2024.
- [24] G.-Y. Zhu, W.-D. Li, W. E. I. Sha, H.-X. Zhou, and W. Hong, “A directional multilevel complex-space fast multipole algorithm for electrically large problems of various dimensions,” *IEEE Transactions on Antennas and Propagation*, vol. 72, no. 1, pp. 22–36, 2024.
- [25] B. Engquist and L. Ying, “Fast directional multilevel algorithms for oscillatory kernels,” *SIAM Journal on Scientific Computing*, vol. 29, no. 4, pp. 1710–1737, 2007.

- [26] M. Messner, M. Schanz, and E. Darve, “Fast directional multilevel summation for oscillatory kernels based on Chebyshev interpolation,” *Journal of Computational Physics*, vol. 231, no. 4, pp. 1175–1196, 2012.
- [27] T. B. Hansen, “Translation operator based on Gaussian beams for the fast multipole method in three dimensions,” *Wave Motion*, vol. 50, no. 5, pp. 940–954, 2013.
- [28] Z. Wei and X. Chen, “Deep-learning schemes for full-wave nonlinear inverse scattering problems,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 1849–1860, 2019.
- [29] H. H. Zhang, H. M. Yao, L. Jiang, and M. Ng, “Solving electromagnetic inverse scattering problems in inhomogeneous media by deep convolutional encoder–decoder structure,” *IEEE Transactions on Antennas and Propagation*, vol. 71, no. 3, pp. 2867–2872, 2023.
- [30] Q. Wu, H. Wang, and W. Hong, “Multistage collaborative machine learning and its application to antenna modeling and optimization,” *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 5, pp. 3397–3409, 2020.
- [31] B. Liu, H. Aliakbarian, Z. Ma, G. A. E. Vandenbosch, G. Gielen, and P. Excell, “An efficient method for antenna design optimization based on evolutionary computation and machine learning techniques,” *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 1, pp. 7–18, 2014.
- [32] D. Xiao, L. Guo, W. Liu, and M. Hou, “Efficient RCS prediction of the conducting target based on physics-inspired machine learning and experimental design,” *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 4, pp. 2274–2289, 2021.
- [33] W.-W. Zhang, D.-H. Kong, X.-Y. He, and M.-Y. Xia, “A machine learning method for 2-D scattered far-field prediction based on wave coefficients,” *IEEE Antennas and Wireless Propagation Letters*, vol. 22, no. 5, pp. 1174–1178, 2023.

- [34] J.-J. Sun, S. Sun, Y. P. Chen, L. Jiang, and J. Hu, "Machine-learning-based hybrid method for the multilevel fast multipole algorithm," *IEEE Antennas and Wireless Propagation Letters*, vol. 19, no. 12, pp. 2177–2181, 2020.
- [35] T. Shan, J. Zeng, X. Song, R. Guo, M. Li, F. Yang, and S. Xu, "Physics-informed supervised residual learning for electromagnetic modeling," *IEEE Transactions on Antennas and Propagation*, vol. 71, no. 4, pp. 3393–3407, 2023.
- [36] T. Shan, M. Li, F. Yang, and S. Xu, "Solving combined field integral equations with physics-informed graph residual learning for em scattering of 3-D PEC targets," *IEEE Transactions on Antennas and Propagation*, vol. 72, no. 1, pp. 733–744, 2024.
- [37] D.-J. Yun, I. I. Jung, H. Jung, H. Kang, W.-Y. Yang, and I. Y. Park, "Improvement in computation time of the finite multipole method by using k-means clustering," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 9, pp. 1814–1817, 2019.
- [38] B. Karaosmanoğlu and Ö. Ergül, "Acceleration of MLFMA simulations using trimmed tree structures," *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 1, pp. 356–365, 2021.
- [39] H. Topözülü, B. Karaosmanoğlu, and V. B. Ertürk, "Trimmed multilevel fast multipole algorithm for D-type volume integral equations," in *2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI)*, pp. 1936–1937, 2022.
- [40] M. Vikram, H. Huang, B. Shanker, and T. Van, "A novel wideband FMM for fast integral equation solution of multiscale problems in electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 7, pp. 2094–2104, 2009.
- [41] M. Vikram, A. Baczewski, B. Shanker, and L. Kempel, "Accelerated Cartesian expansion (ACE) based framework for the rapid evaluation of diffusion, lossy wave, and Klein–Gordon potentials," *Journal of Computational Physics*, vol. 229, no. 24, pp. 9119–9134, 2010.

- [42] B. Shanker and H. Huang, “Accelerated Cartesian expansions – A fast method for computing of potentials of the form $R^{-\nu}$ for all real ν ,” *Journal of Computational Physics*, vol. 226, no. 1, pp. 732–753, 2007.
- [43] M. Kalfa, V. B. Ertürk, and Ö. Ergül, “Error analysis of MLFMA with closed-form expressions,” *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 10, pp. 6618–6623, 2021.
- [44] 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 Journal on Numerical Analysis*, vol. 36, no. 3, pp. 906–921, 1999.
- [45] A. González, “Measurement of areas on a sphere using Fibonacci and latitude–longitude lattices,” *Mathematical Geosciences*, vol. 42, p. 49–64, Nov. 2009.
- [46] Y. He, J. F. Li, X. J. Jing, and M. S. Tong, “Fast solution of volume–surface integral equations for multiscale structures,” *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 12, pp. 7649–7654, 2019.
- [47] C. Lee, H. Hasegawa, and S. Gao, “Complex-valued neural networks: A comprehensive survey,” *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 8, pp. 1406–1426, 2022.
- [48] Seçil Eda Doğan, “A novel hierarchical machine-learning-based method for efficient solutions of electromagnetic scattering problems,” M.Sc. thesis, Bilkent University, Ankara, Turkey, July 2021.
- [49] J. A. Barrachina, “Negu93/cvnn: Complex-valued neural networks,” Nov. 2022.
- [50] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.
- [51] M. M. Botha, “Solving the volume integral equations of electromagnetic scattering,” *Journal of Computational Physics*, vol. 218, no. 1, pp. 141–158, 2006.

- [52] L. Gürel, H. Bağcı, J. C. Castelli, A. Cheraly, and F. Tardivel, “Validation through comparison: Measurement and calculation of the bistatic radar cross section of a stealth target,” *Radio Science*, vol. 38, no. 3, 2003.
- [53] V. Y. Pan, *Structured Matrices and Polynomials: Unified Superfast Algorithms*. Birkhäuser Boston, 2001.
- [54] Z. Tang, *Fast transforms based on structured matrices with applications to the fast multipole method*. Phd thesis, University of Maryland, College Park, College Park, MD, 2003.
- [55] J. Wan and W. Zudilin, “Generating functions of Legendre polynomials: A tribute to Fred Braffman,” *Journal of Approximation Theory*, vol. 164, no. 4, pp. 488–503, 2012.
- [56] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, “KAN: Kolmogorov-arnold networks,” 2025.
- [57] O. Ciftja, “Integrals of Legendre polynomials over half range and their relation to the electrostatic potential in hemispherical geometry,” *Results in Physics*, vol. 40, p. 105838, 2022.

Appendix A

Finding coefficients of Taylor Expansion

As previously discussed, for $r > r'$, the Green's function $R = |\bar{r} + \bar{r}'|$ and its reciprocal admit the following Legendre polynomial expansions [55]:

$$R = r - r' \cos \gamma + \sum_{\ell=1}^{\infty} \frac{r'^{\ell+1}}{(2\ell+1)r^{\ell}} (P_{\ell-1}(\cos \gamma) - P_{\ell+1}(\cos \gamma)), \quad (\text{A.1})$$

$$\frac{1}{R} = \sum_{\ell=0}^{\infty} \frac{r'^{\ell}}{r^{\ell+1}} P_{\ell}(\cos \gamma), \quad (\text{A.2})$$

where $\cos \gamma = -\hat{r} \cdot \hat{r}'$, and $P_{\ell}(\cdot)$ denotes the Legendre polynomial of degree ℓ . Using these expressions, the function p can be expanded as

$$\begin{aligned} p(\bar{r}') &= \sum_{\ell=0}^{\infty} \frac{r'^{\ell}}{r^{\ell}} P_{\ell}(-\hat{r} \cdot \hat{r}') \\ &\times \exp \left(-jk \sum_{\ell=1}^{\infty} \frac{r'^{\ell+1}}{(2\ell+1)r^{\ell}} (P_{\ell-1}(-\hat{r} \cdot \hat{r}') - P_{\ell+1}(-\hat{r} \cdot \hat{r}')) \right). \end{aligned} \quad (\text{A.3})$$

To compute a d th-order Taylor expansion of $p(\bar{r}')$ with respect to the local coordinates

$$z' := \hat{r} \cdot \bar{r}', \quad y' := \hat{\theta} \cdot \bar{r}', \quad x' := \hat{\phi} \cdot \bar{r}',$$

the following procedure is applied. All expressions are treated as multivariate polynomials in the variables (x', y', z') , and all operations (including addition,

multiplication, and truncation) are performed by explicitly manipulating their monomial coefficients in (x', y', z') , with each term represented as a sum of monomials of the form $x'^a y'^b z'^c$. This coefficient-based manipulation ensures precise tracking of polynomial degree at each step and allows systematic truncation of all terms exceeding total degree d .

- **Homogeneous Polynomial Identification:** The Taylor expansion of $p(\bar{r}')$ is carried out with respect to the local spherical coordinates (x', y', z') defined above. To determine which terms contribute to a d th-order approximation, we analyze the polynomial structure of the expansion components. Each Legendre polynomial $P_\ell(-\hat{r} \cdot \hat{r}')$ is evaluated at a linear combination of x', y', z' due to the projection

$$\hat{r} \cdot \hat{r}' = \frac{z'}{r'}.$$

Hence, $P_\ell(-\hat{r} \cdot \hat{r}') = P_\ell\left(-\frac{z'}{r'}\right)$ becomes a degree- ℓ polynomial in $\frac{z'}{r'}$. When multiplied by r'^ℓ , the resulting expression

$$r'^\ell P_\ell\left(-\frac{z'}{r'}\right)$$

becomes a homogeneous polynomial of total degree ℓ in the variables (x', y', z') , since $z' = \hat{r} \cdot \bar{r}'$ is linear in $\bar{r}'$. Similarly, the term

$$r'^{\ell+1} \left(P_{\ell-1}\left(-\frac{z'}{r'}\right) - P_{\ell+1}\left(-\frac{z'}{r'}\right) \right)$$

involves Legendre polynomials of degrees $\ell - 1$ and $\ell + 1$, and upon multiplication by $r'^{\ell+1}$, produces a homogeneous polynomial of degree $\ell + 1$ in (x', y', z') . This homogeneity is essential for Taylor expansion: if we seek a d th-order expansion of $p(\bar{r}')$, we retain only those terms for which the total polynomial degree in (x', y', z') does not exceed d . Any term of a higher total degree is discarded. Thus, the coefficients of the Legendre polynomial terms are explicitly computed and stored, allowing each term to be represented as a homogeneous polynomial in (x', y', z') for use in the Taylor expansion.

- **Series Truncation and Exponential Expansion:** Each inner exponential term of the form

$$\exp\left(-jk \cdot \frac{r'^{\ell+1}}{(2\ell+1)r^\ell} \left(P_{\ell-1}\left(-\frac{z'}{r'}\right) - P_{\ell+1}\left(-\frac{z'}{r'}\right)\right)\right)$$

is expanded using a Taylor series in its argument ν :

$$\exp(\nu) \approx \sum_{n=0}^{d-\ell} \frac{\nu^n}{n!},$$

where the truncation at $d - \ell$ ensures that resulting terms remain within the total degree limit d .

- **Recursive Multiplication and Degree Trimming:** The exponential factors are then multiplied recursively from $\ell = 1$ up to $\ell = d$, and at each step, any terms whose total degree exceeds d are discarded to maintain the correct order of approximation.
- **Final Multiplication with the Polynomial Prefactor:** Finally, the truncated exponential product is multiplied with the truncated series

$$\sum_{\ell=0}^d \frac{r'^\ell}{r^\ell} P_\ell\left(-\frac{z'}{r'}\right),$$

again discarding all terms of total degree greater than d .

Following this procedure yields the complete d th-order Taylor expansion of the scalar function $p(\bar{r}')$ in terms of the spherical projection coordinates (x', y', z') defined above, with all higher-order contributions excluded.

Appendix B

Derivation of R Expansion

As stated earlier, for $r > r'$, the expansion of $1/|\bar{r} - \bar{r}'| = 1/R$ is given by [55] as

$$\frac{1}{R} = \frac{1}{\sqrt{r^2 + r'^2 - 2rr'x}} = \sum_{\ell=0}^{\infty} \frac{r'^{\ell}}{r^{\ell+1}} P_{\ell}(x), \quad x = \hat{r} \cdot \hat{r}', \quad (\text{B.1})$$

where $P_{\ell}(x)$ denotes the Legendre polynomial of degree ℓ . We may express R itself using the integral identity

$$\begin{aligned} R &= \sqrt{r^2 + r'^2 - 2rr'x} \\ &= r + r' - \int_{-1}^x \frac{rr'}{\sqrt{r^2 + r'^2 - 2rr'x'}} dx' = r + r' - rr' \int_{-1}^x \frac{1}{R(x')} dx', \end{aligned} \quad (\text{B.2})$$

and by substituting the expansion in (B.1), we obtain

$$\begin{aligned} R &= r + r' - rr' \int_{-1}^x \sum_{\ell=0}^{\infty} \frac{r'^{\ell}}{r^{\ell+1}} P_{\ell}(x') dx' \\ &= r + r' - \sum_{\ell=0}^{\infty} \frac{r'^{\ell+1}}{r^{\ell}} \int_{-1}^x P_{\ell}(x') dx'. \end{aligned} \quad (\text{B.3})$$

Now, we compute these integrals using the properties of Legendre polynomials.

For $\ell = 0$,

$$\int_{-1}^x P_0(x') dx' = \int_{-1}^x 1 dx' = x + 1.$$

For $n \geq 1$, the antiderivative of $P_n(x)$ is given by [57]

$$\int P_n(x) dx = \frac{1}{2n+1} [P_{n+1}(x) - P_{n-1}(x)], \quad (\text{B.4})$$

so

$$\begin{aligned}
\int_{-1}^x P_n(x') dx' &= \frac{1}{2n+1} [P_{n+1}(x) - P_{n-1}(x) - P_{n+1}(-1) + P_{n-1}(-1)] \\
&= \frac{1}{2n+1} [P_{n+1}(x) - P_{n-1}(x) - (-1)^{n+1} + (-1)^{n-1}] \\
&= \frac{1}{2n+1} [P_{n+1}(x) - P_{n-1}(x)],
\end{aligned}$$

since $(-1)^{n+1} - (-1)^{n-1} = 0$. Now plug back into (B.3):

$$\begin{aligned}
R &= r + r' - \frac{r'^1}{r^0}(x+1) - \sum_{\ell=1}^{\infty} \frac{r'^{\ell+1}}{r^{\ell}} \cdot \frac{1}{2\ell+1} [P_{\ell+1}(x) - P_{\ell-1}(x)] \\
&= r - r'x + \sum_{\ell=1}^{\infty} \frac{r'^{\ell+1}}{(2\ell+1)r^{\ell}} [P_{\ell-1}(x) - P_{\ell+1}(x)].
\end{aligned} \tag{B.5}$$

Thus, we have the expansion

$$R = r - r'x + \sum_{\ell=1}^{\infty} \frac{r'^{\ell+1}}{(2\ell+1)r^{\ell}} [P_{\ell-1}(x) - P_{\ell+1}(x)]. \tag{B.6}$$

Appendix C

An Example of Taylor Expansion

An example of Taylor expansion given in (4.5) with degree $d = 2$ is as follows:

$$p_2(z', x', y') = 1 - \left(\frac{jk}{2r}\right) x^2 - \left(\frac{jk}{2r}\right) y^2 + \left(\frac{1}{r}\right) z - \left(\frac{1}{2r^2}\right) x^2 - \left(\frac{1}{2r^2}\right) y^2 + \left(\frac{1}{r^2}\right) z^2 \quad (\text{C.1})$$

where $z' = \hat{r} \cdot \bar{r}'$, $y' = \hat{\theta} \cdot \bar{r}'$, and $x' = \hat{\phi} \cdot \bar{r}'$. If we separate $\bar{r}' = \bar{r}'_1 + \bar{r}'_2$, we obtain

$$\begin{aligned} p_2(x'_1, y'_1, z'_1, x'_2, y'_2, z'_2) &= 1 \\ &- \left(\frac{jk}{2r} + \frac{1}{2r^2}\right) (x'^2_1 + 2x'_1x'_2 + x'^2_2) \\ &- \left(\frac{jk}{2r} + \frac{1}{2r^2}\right) (y'^2_1 + 2y'_1y'_2 + y'^2_2) \\ &+ \left(\frac{1}{r}\right) (z'_1 + z'_2) \\ &+ \left(\frac{1}{r^2}\right) (z'^2_1 + 2z'_1z'_2 + z'^2_2) \end{aligned} \quad (\text{C.2})$$

The basis monomials and test monomials vectors are as follows:

$$\mathfrak{B}(\hat{r}, \bar{r}'_1) = \exp(jkz'_1) [1, x'_1, x'^2_1, y'_1, x'_1y'_1, y'^2_1, z'_1, x'_1z'_1, y'_1z'_1, z'^2_1], \quad (\text{C.3})$$

$$\mathfrak{T}(\hat{r}, \bar{r}'_2) = \exp(jkz'_2) [1, x'_2, x'^2_2, y'_2, x'_2y'_2, y'^2_2, z'_2, x'_2z'_2, y'_2z'_2, z'^2_2], \quad (\text{C.4})$$

where the entries are ordered in a block fashion: d_3 varies fastest, followed by d_2 , and d_1 varies slowest. That is, the monomials are arranged lexicographically

in increasing order of (d_1, d_2, d_3) , which is as stated above. Furthermore, the $\bar{\bar{T}}$ given in (4.12) for this expansion is given as

$$\bar{\bar{T}} = \begin{bmatrix} 1 & 0 & -\left(\frac{jk}{2r} + \frac{1}{2r^2}\right) & 0 \\ 0 & -2\left(\frac{jk}{2r} + \frac{1}{2r^2}\right) & 0 & 0 \\ -\left(\frac{jk}{2r} + \frac{1}{2r^2}\right) & 0 & 0 & 0 \\ 0 & 0 & 0 & -2\left(\frac{jk}{2r} + \frac{1}{2r^2}\right) \\ 0 & 0 & 0 & 0 \\ -\left(\frac{jk}{2r} + \frac{1}{2r^2}\right) & 0 & 0 & 0 \\ \frac{1}{r} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{r^2} & 0 & 0 & 0 \end{bmatrix} \dots \begin{bmatrix} 0 & -\left(\frac{jk}{2r} + \frac{1}{2r^2}\right) & \frac{1}{r} & 0 & 0 & \frac{1}{r^2} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{r^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (\text{C.5})$$

where the first part is its first 4 columns and the second is its last 6 columns, and we have the equality

$$\exp(jkz')p_2(z', x', y') = \bar{\mathfrak{T}}(\hat{r}, \bar{r}_2)^T \bar{\bar{T}} \bar{\mathfrak{B}}(\hat{r}, \bar{r}_1) \quad (\text{C.6})$$

as stated in (4.12). Furthermore, the expansion in (4.7) with $d = 2$ is as follows:

$$\begin{aligned} p'_2(z'_1, y'_1, x'_1, z'_2, y'_2, x'_2) &= +1 \\ &- \frac{jk}{r} x'_1 - \frac{1}{2r^2} x_1'^2 - \frac{jk}{2r} x_1'^2 - \frac{jk}{r^2} x_1'^2 z'_2 \\ &- \frac{3}{2r^3} x_1'^2 z'_2 - \frac{3jk}{2r^3} x_1'^2 z_2'^2 - \frac{3}{r^4} x_1'^2 z_2'^2 \end{aligned}$$

$$\begin{aligned}
& -\frac{jk}{r} y'_1 - \frac{1}{2r^2} y_1'^2 - \frac{jk}{2r} y_1'^2 - \frac{jk}{r^2} y_1'^2 z'_2 \\
& -\frac{3}{2r^3} y_1'^2 z'_2 - \frac{3jk}{2r^3} y_1'^2 z_2'^2 - \frac{3}{r^4} y_1'^2 z_2'^2 \\
& -\frac{4k^2}{r^2} x'_1 x'_2 y'_1 y'_2 - \frac{3jk}{r^3} x'_1 x'_2 y'_1 y'_2 + \frac{3}{r^4} x'_1 x'_2 y'_1 y'_2 \\
& + \frac{1}{r} z'_1 + \frac{1}{r^2} z_1'^2 \\
& -\frac{2jk}{r^2} x'_1 x'_2 z'_1 - \frac{3}{r^3} x'_1 x'_2 z'_1 \\
& -\frac{2jk}{r^2} y'_1 y'_2 z'_1 - \frac{3}{r^3} y'_1 y'_2 z'_1 \\
& -\frac{3jk}{2r^3} x_2'^2 z_1'^2 - \frac{3}{r^4} x_2'^2 z_1'^2 - \frac{3jk}{2r^3} y_2'^2 z_1'^2 - \frac{3}{r^4} y_2'^2 z_1'^2 \\
& + \frac{6}{r^4} z_1'^2 z_2'^2 \\
& -\frac{jk}{r} x'_2 - \frac{1}{2r^2} x_2'^2 - \frac{jk}{2r} x_2'^2 - \frac{jk}{r^2} x_2'^2 z'_1 - \frac{3}{2r^3} x_2'^2 z'_1 \\
& -\frac{jk}{r} y'_2 - \frac{1}{2r^2} y_2'^2 - \frac{jk}{2r} y_2'^2 - \frac{jk}{r^2} y_2'^2 z'_1 - \frac{3}{2r^3} y_2'^2 z'_1 \\
& -\frac{k^2}{r^2} x_1'^2 y_2'^2 + \frac{3jk}{r^3} x_1'^2 y_2'^2 + \frac{3}{4r^4} x_1'^2 y_2'^2 \\
& -\frac{k^2}{r^2} x_2'^2 y_1'^2 + \frac{3jk}{r^3} x_2'^2 y_1'^2 + \frac{3}{4r^4} x_2'^2 y_1'^2 \\
& + \frac{1}{r} z'_2 + \frac{1}{r^2} z_2'^2 \\
& -\frac{2jk}{r^2} x'_1 x'_2 z'_2 - \frac{3}{r^3} x'_1 x'_2 z'_2 \\
& -\frac{2jk}{r^2} y'_1 y'_2 z'_2 - \frac{3}{r^3} y'_1 y'_2 z'_2 \\
& + \frac{3}{r^3} z_1' z_2'^2 + \frac{3}{r^3} z_1'^2 z_2' \\
& -\frac{3k^2}{r^2} x_1'^2 x_2'^2 + \frac{9jk}{r^3} x_1'^2 x_2'^2 + \frac{9}{4r^4} x_1'^2 x_2'^2 \\
& -\frac{3k^2}{r^2} y_1'^2 y_2'^2 + \frac{9jk}{r^3} y_1'^2 y_2'^2 + \frac{9}{4r^4} y_1'^2 y_2'^2 \\
& -\frac{6jk}{r^3} x'_1 x'_2 z'_1 z'_2 - \frac{12}{r^4} x'_1 x'_2 z'_1 z'_2 \\
& -\frac{6jk}{r^3} y'_1 y'_2 z'_1 z'_2 - \frac{12}{r^4} y'_1 y'_2 z'_1 z'_2
\end{aligned} \tag{C.7}$$

and corresponding $\bar{\bar{T}}$ matrix is as

$$\bar{\bar{T}} = \begin{bmatrix} 1 & 0 & \frac{-1}{2r^2} + \frac{-jk}{2r} & 0 & 0 \\ 0 & \frac{-1}{r^2} + \frac{-jk}{r} & 0 & 0 & 0 \\ \frac{-1}{2r^2} + \frac{-jk}{2r} & 0 & \frac{9}{4r^4} - \frac{3k^2}{4r^2} + \frac{9jk}{4r^3} & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{r^2} + \frac{-jk}{r} & 0 \\ 0 & 0 & 0 & 0 & \frac{3}{r^4} - \frac{k^2}{r^2} + \frac{3jk}{r^3} \dots \\ \frac{-1}{2r^2} + \frac{-jk}{2r} & 0 & \frac{3}{4r^4} - \frac{k^2}{4r^2} + \frac{3jk}{4r^3} & 0 & 0 \\ \frac{1}{r} & 0 & \frac{-3}{2r^3} + \frac{-jk}{r^2} & 0 & 0 \\ 0 & \frac{-3}{r^3} + \frac{-2jk}{r^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-3}{r^3} + \frac{-2jk}{r^2} & 0 \\ \frac{1}{r^2} & 0 & \frac{-3}{r^4} + \frac{-3jk}{2r^3} & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \frac{-1}{2r^2} + \frac{-jk}{2r} & \frac{1}{r} & 0 & 0 & \frac{1}{r^2} \\ 0 & 0 & \frac{-3}{r^3} + \frac{-2jk}{r^2} & 0 & 0 \\ \frac{3}{4r^4} - \frac{k^2}{4r^2} + \frac{3jk}{4r^3} & \frac{-3}{2r^3} + \frac{-jk}{r^2} & 0 & 0 & \frac{-3}{r^4} + \frac{-3jk}{2r^3} \\ 0 & 0 & 0 & \frac{-3}{r^3} + \frac{-2jk}{r^2} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{9}{4r^4} - \frac{3k^2}{4r^2} + \frac{9jk}{4r^3} & \frac{-3}{2r^3} + \frac{-jk}{r^2} & 0 & 0 & \frac{-3}{r^4} + \frac{-3jk}{2r^3} \\ \frac{-3}{2r^3} + \frac{-jk}{r^2} & \frac{2}{r^2} & 0 & 0 & \frac{3}{r^3} \\ 0 & 0 & \frac{-12}{r^4} + \frac{-6jk}{r^3} & 0 & 0 \\ 0 & 0 & 0 & \frac{-12}{r^4} + \frac{-6jk}{r^3} & 0 \\ \frac{-3}{r^4} + \frac{-3jk}{2r^3} & \frac{3}{r^3} & 0 & 0 & \frac{6}{r^4} \end{bmatrix}, \quad (\text{C.8})$$

where the first part is its first 5 columns and the second is its last 5 columns. Similar to the previous, we have the equality

$$\exp(jkz')p'_2(z'_1, y'_1, x'_1, z'_2, y'_2, x'_2) = \bar{\mathfrak{T}}(\hat{r}, \bar{r}'_2)^T \bar{\bar{T}} \bar{\mathfrak{B}}(\hat{r}, \bar{r}'_1). \quad (\text{C.9})$$

Here, for the recurrence relationship, we can only determine the coefficient with order $z'^2_{1,2}$ terms because in $d = 2$, it is the only possibility. Thus, the recurrence relationship matrix $\bar{\bar{R}}$ given in (4.30) is a concatenation of an identity matrix and

a row. For $d = 2$, the matrix $\bar{\bar{R}}$ is as follows:

$$\bar{\bar{R}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -jk \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (\text{C.10})$$