LARGE-SCALE SOLUTIONS OF ELECTROMAGNETICS PROBLEMS USING THE MULTILEVEL FAST MULTIPOLE ALGORITHM AND PHYSICAL OPTICS

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

Mert Hidayetoğlu

April, 2015

LARGE-SCALE SOLUTIONS OF ELECTROMAGNETICS PROBLEMS USING THE MULTILEVEL FAST MULTIPOLE ALGORITHM AND PHYSICAL OPTICS

by Mert Hidayetoğlu

April, 2015

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.

Assoc. Prof. Ömer İlday (Advisor)

Prof. Ayhan Altıntaş

Prof. Adnan Köksal

Approved for the Graduate School of Engineering and Science:

Prof. Levent Onural
Director of the Graduate School

ABSTRACT

LARGE-SCALE SOLUTIONS OF ELECTROMAGNETICS PROBLEMS USING THE MULTILEVEL FAST MULTIPOLE ALGORITHM AND PHYSICAL OPTICS

Mert Hidayetoğlu
Master of Science in Electrical and Electronics Engineering
Advisor: Assoc. Prof. Ömer İlday
April, 2015

Integral equations provide full-wave (accurate) solutions of Helmholtz-type electromagnetics problems. The multilevel fast multipole algorithm (MLFMA) discretizes the equations and solves them numerically with $\mathcal{O}(NLogN)$ complexity, where N is the number of unknowns. For solving large-scale problems, MLFMA is parallelized on distributed-memory architectures. Despite the low complexity and parallelization, the computational requirements of MLFMA solutions grow immensely in terms of CPU time and memory when extremely-large geometries (in wavelengths) are involved. The thesis provides computational and theoretical techniques for solving large-scale electromagnetics problems with lower computational requirements. One technique is the out-of-core implementation for reducing the required memory via employing disk space for storing large data. Additionally, a pre-processing parallelization strategy, which eliminates memory bottlenecks, is presented. Another technique, MPI+OpenMP parallelization, uses distributed-memory and shared-memory schemes together in order to maintain the parallelization efficiency with high number of processes/threads. The thesis also includes the out-of-core implementation in conjunction with the MPI+OpenMP parallelization. With the applied techniques, full-wave solutions involving up to 1.3 billion unknowns are achieved with 2 TB memory. Physical optics is a high-frequency approximation, which provides fast solutions of scattering problems with $\mathcal{O}(N)$ complexity. A parallel physical optics algorithm is presented in order to achieve fast and approximate solutions. Finally, a hybrid integral-equation and physical-optics solution methodology is introduced.

Keywords: Integral equations, multilevel fast multipole algorithm, physical optics, electromagnetic scattering, parallel computing, out-of-core method.

ÖZET

ÇOK SEVİYELİ HIZLI ÇOKKUTUP YÖNTEMİ VE FİZİKSEL OPTİK İLE BÜYÜK ÖLÇEKLİ ELEKTROMANYETİK PROBLEMLERİN ÇÖZÜMLERİ

Mert Hidayetoğlu
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Doç. Dr. Ömer İlday
Nisan, 2015

İntegral denklemleri, Helmholtz tipi elektromanyetik problemlerin tam dalga (doğru) çözümlerini sağlamaktadır. Çok seviyeli hızlı çokkutup yöntemi (ÇSHÇY) bu denklemleri numerik ortamda ayrıklaştırarak N bilinmeyenli bir denklemi $\mathcal{O}(NLogN)$ karmaşıklığında çözer. Büyük problemlerin çözülebilmesi için ÇSHÇY dağınık bellek mimarilerinde paralelleştirilmiştir. Düşük karmaşıklığı ve paralelleştirilmesine rağmen, büyük geometriler içeren (dalga boyu cinsinden) ÇSHÇY çözümlerinin hesaplama gereksinimleri CPU zamanı ve bellek bakımından muazzam bir şekilde artmaktadır. Bu tez büyük ölçekli problemlerin daha az hesaplama gereksinimi ile çözülebilmesi için hesaplamalı ve teorik yöntemler sunmaktadır. Bir yöntem büyük verilerin saklanması için disk alanını kullanarak bellek gereksinimini düşüren dışarı yazma uygulamasıdır. Ek olarak, bellek darboğazlarını aşabilmek için önişlemeyi paralelleştiren bir strateji sunulmuştur. Başka bir teknik olan MPI+OpenMP paralelizasyonu dağınık bellek ve paylaşımlı bellek düzenlerini birlikte kullanarak yüksek işlem/işlemcik sayılarında etkin paralelleştirme sağlar. Ayrıca tez, dışarı yazma önteminin MPI+OpenMP paralelizasyonu ile birlikte kullanılmasını da kapsar. Uygulanan yntemlerle 2 TB bellek kullanılarak 1.3 milyara kadar bilinmeyen içeren problemlerin tam dalga çözümleri sağlanmıştır. Bir tüksek frekans yaklaşımı olan fiziksel optik, saçılım problemlerinin hızlı çözümlerini $\mathcal{O}(N)$ karmaşıklığında yaklaşık olarak sağlar. Yaklaşık ve hızlı çözümler için bir paralel fiziksel optik algoritması sunulmuştur. Son olarak, integral denklemlerini ve fiziksel optiği hibrit olarak kullanan bir çözüm yöntemi tanıtılmıştır.

Anahtar sözcükler: İntegral denklemleri, çok seviyeli hızlı çokkutup yöntemi, fiziksel optik, elektromanyetik saçılım, paralel hesaplama, dışarı yazma yöntemi.

Acknowledgement

I would like to express my deepest gratitude to Assoc. Prof. Ömer İlday for advising me in the last four months of my MS studies due to resignation of my former advisor Prof. Levent Gürel. I learned a lot from him.

I would like to thank Prof. Ayhan Altıntaş and Prof. Adnan Köksal for evaluating and commenting on the thesis. I would also thank my colleagues in Bilkent University, the Computational Electromagnetics Research Center (BiLCEM), and the Ultrafast Optics and Lasers Laboratory (UFOLAB) for their friendship.

I would like to thank my former advisor Prof. Levent Gürel for providing me the research area and his full support for my studies in computational electromagnetics. Without him, I would not be in love with electromagnetics.

My final gratitude is to Duygu Çelik and my family, who gave me their love, support, and encouragement throughout my studies, and I would like to dedicate this thesis to them.

This work was supported by Turkcell Corporation through graduate scholarship, by Scientific and Technical Research Council of Turkey (TÜBİTAK) under the Research Grants 111E203 and 11E268, by Sclumberger-Doll Research (SDR), and by contracts from Military Electronic Industries (ASELSAN), Turkish Aerospace Industries (TAI), and Undersecretariat for Defense Industries (SSM). I would like to thank Intel Corporation and Turkish Academic Network and Information Center (ULAKBİM) for generous allocations of parallel-computer time, and Jamie Wilcox of Intel for his invaluable expertise.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Electric-field and magnetic-field integral equations provide full-wave solutions of Helmholtz-type electromagnetics problems. Starting from the Maxwell's equations, derivations of the integral equations and their employment for solving electromagnetics problems are introduced in Section 2.1 and Section 2.2, respectively. The integral equations cannot be solved analytically when arbitrary geometries are involved. The method of moments (MoM) [2],[3], described in Section 2.3, discretises the integral equations in a numerical environment and turns the electromagnetics problem into a linear system of equations $\overline{\boldsymbol{Z}} \cdot \boldsymbol{x} = \boldsymbol{v}$ with N unknowns, where $\overline{\boldsymbol{Z}}$ is the known impedance (or interaction) matrix, and $\boldsymbol{x}$ is the unknown coefficient-vector. The solutions of the MoM system using conventional techniques, e.g., Gaussian elimination or matrix inversion, require $\mathcal{O}(N^2)$ memory and $\mathcal{O}(N^3)$ time, whereas iterative solutions [4] via matrix-vector multiplications (MVMs) have $\mathcal{O}(IN^2)$ computational complexity, where I is the number of iterations. For the rest of the thesis, the number of iterations is ignored for simplicity when computational complexity is mentioned. There are plenty of choices among the iterative solvers [5], e.g., least square (LSQR) [6], conjugate gradients squared (CGS) [7], or biconjugate gradient stablilized (Bi-CGSTAB) [8], and their convergence performances may vary by formulation and problem type. For a stronger convergence, preconditioners are developed for electromagnetics solutions, e.g. block-diagonal [9],[10] or near-field [11], in order to be used in conjunction with

the iterative solvers.

The diagonalization and factorization of the Green's function of the Helmholtz's equation using multipole expansions [12],[13] provides flexibility to deal with groups of elements, as opposed individual pairs, in an N-body problem, where each element interacts with every elements [14]. The multilevel fast multipole algorithm (MLFMA) [10] uses these properties to reduce the complexity of the iterative solution down to $\mathcal{O}(NLogN)$ [10]. MLFMA partitions MoM system as $\overline{\boldsymbol{Z}}_{NF} \cdot \boldsymbol{x} + \overline{\boldsymbol{Z}}_{FF} \cdot \boldsymbol{x} = \boldsymbol{v}$, where $\overline{\boldsymbol{Z}}_{NF}$ is the near-field and $\overline{\boldsymbol{Z}}_{FF}$ is the far-field interaction matrices. Holding the interactions between spatially close elements, the near-field interaction matrix has $\mathcal{O}(N)$ non-zero elements and is treated in the same way as in the conventional MoM, whereas the far-field interaction matrix is multiplied on-the-fly in each MVM. Each far-field MVM, i.e $\overline{\boldsymbol{Z}}_{FF} \cdot \boldsymbol{x}$, has $\mathcal{O}(NLogN)$ complexity which also defines the complexity of MLFMA [1].

MLFMA's low complexity and its error-controllable nature makes it the most qualified algorithm for electromagnetics solutions up to date [15],[16]. Despite its low complexity, large-scale MLFMA solutions require vast amount of memory and CPU time that is not available in ordinary computers, e.g., PCs. Therefore MLFMA is parallelized on distributed-memory architectures order to partition the data and the task among processes employed in multiple computing nodes [17]. The hierarchical partitioning strategy provides excellent efficiency up to 128 processes and it is the most efficient strategy up to date [18], [19],[20]. The details of MLFMA and the hierarchical partitioning strategy are introduced in Section 2.4.

In spite of MLFMA's efficient parallelization, the memory capacity of the computational platform limits the scale of solvable problems [21],[22],[23]. To solve larger problems within a memory limit, an out-of-core implementation of MLFMA is developed for employing disk space for storing large data-structures. Those data structures are stored once and used from disk when needed. In order to prevent slowdown, solid-state disks (SSD) are used in order to read and write the out-of-core data fast [24]. The details of the out-of-core implementation [25],[26]

and its parallel employment are provided in Section 3.1 and Section 3.3, respectively. Section 3.3 also presents large-scale solutions of electromagnetics problems involving up to 1.3 billion unknowns using 2 TB memory,[27].

The out-of-core implementation reduces the peak memory, occurred in the iterative solution, However, the reduced memory reveals another memory bottleneck in the pre-processing stage of the algorithm. The the pre-processing memory is constituted by large data-structures of the input geometry which grow immensely when large geometries are involved. The hierarchical partitioning strategy does not give a receipt for parallelizing the pre-processing, therefore, a pre-processing parallelization scheme is developed for achieving data-parallelism. The scheme distributes the data structures as the geometry is being read from the input file and performs organized communications at certain steps in order to pass the data portions among processes. The pre-processing parallelization scheme is introduced in Section 3.2.

In a distributed-memory parallelization scheme, e.g. hierarchical partitioning strategy, it is inevitable to duplicate some data structures in order not to make intense communications. As a result, the required memory increases proportional to the number of processes, and unfortunately, number of processes cannot be increased within the memory limits and a portion of processing cores may become idle. As a remedy, a hybrid parallelization scheme, namely MPI+OpenMP parallelization, is developed which uses distributed-memory and shared-memory schemes together. At certain steps of the algorithm, each MPI process are forked off into multiple OpenMP threads, where the threads can share the workload and the memory of the belonging MPI process, thus, all processing cores can be employed without requiring extra memory. The OpenMP+MPI parallelization scheme and its employment in conjunction with the out-of-core implementation are provided in Sections 3.4 and in 3.5, respectively.

Physical optics is a high-frequency approximation for solutions of scattering problems [28]. Although it is approximate, it is fast with its $\mathcal{O}(N)$ complexity. A parallel physical optics solver is developed for fast and approximate solutions of extremely large-scale scattering problems [27]. Physical optics is introduced

in Section 2.5, and the solution algorithm and its parallelization are provided in Sections 4.1 and 4.2, respectively. Section 4.2 includes large-scale scattering solutions for demonstrating the parallel solver's performance.

For combining the rapidity of physical-optics and the accuracy of integral-equation (MLFMA) solutions, a hybrid solution methodology for employing good sides of both methods is developed in [29]. The methodology partitions the problem domain into integral-equation and physical-optics domains and solves the surface current on the domains with corresponding formulation. Typically, large and uncomplicated surfaces are solved with physical optics in $\mathcal{O}(N)$ time and the rest is solved with the integral equations in $\mathcal{O}(NLogN)$ time. The solutions cannot be thought independent from each other and the methodology resolves the interaction among domains in $\mathcal{O}(NLogN)$ time. The formulation of the proposed methodology is provided in Section 5.1. Solutions of scattering and radiation problems are demonstrated in Section 5.2 and Section 5.3, respectively.

Chapter 2

Background

2.1 Field Integrals in Electromagnetics

The time-harmonic Maxwell's equations can be written as

$$\nabla \times \boldsymbol{E}(\boldsymbol{r}) = i\omega\mu\boldsymbol{H}(\boldsymbol{r}) \tag{2.1}$$

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

$$\nabla \cdot \boldsymbol{H}(\boldsymbol{r}) = 0 \tag{2.3}$$

$$\nabla \cdot \boldsymbol{E}(\boldsymbol{r}) = \rho(\boldsymbol{r})/\epsilon \tag{2.4}$$

in $i\omega t$ time convention, i.e., $\boldsymbol{E}(\boldsymbol{r}, t) = Re\{\boldsymbol{E}(\boldsymbol{r})^{i\omega t}\}$, and in homogeneous and isotropic medium, i.e., $\boldsymbol{D}(\boldsymbol{r}) = \epsilon\boldsymbol{E}(\boldsymbol{r})$ and $\boldsymbol{B}(\boldsymbol{r}) = \mu\boldsymbol{H}(\boldsymbol{r})$, where $\boldsymbol{E}(\boldsymbol{r})$ and $\boldsymbol{H}(\boldsymbol{r})$ are the electric-field and magnetic-field intensities in V/m and A/m, $\boldsymbol{B}(\boldsymbol{r})$ and $\boldsymbol{D}(\boldsymbol{r})$ are the magnetic and electric flux densities in Wb/m^2 and C/m^2, $\boldsymbol{J}(\boldsymbol{r})$ and $\rho(\boldsymbol{r})$ are the electric current and charge densities in A/m^2 and C/m^3, ϵ and μ are the electric permittivity and the magnetic permeability in F/m and H/m, respectively, and ω is the angular frequency in rad/s. The electric current and charge densities can be related with the continuity equation $\nabla \cdot \boldsymbol{J} = i\omega\rho$.

The electric-field and the magnetic-field integrals are to be developed in order to find the vectors $\boldsymbol{E}$ and $\boldsymbol{H}$ for any source distribution $\boldsymbol{J}$. To that end, two

auxiliary vector fields, the vector magnetic potential $\boldsymbol{A}$ and the scalar electric potential ϕ, are defined.

The divergence of the curl of any vector field $\boldsymbol{\psi}$ is always zero, i.e $\nabla \cdot (\nabla \times \boldsymbol{\psi}) = 0$, and therefore the curl of the vector magnetic potential $\boldsymbol{A}$ can be defined as

$$\mu \boldsymbol{H} = \nabla \times \boldsymbol{A} \tag{2.5}$$

in accordance with (2.3). If (2.5) is substituted into (2.1),

$$\nabla \times (\boldsymbol{E} - i\omega \boldsymbol{A}) = 0. \tag{2.6}$$

The curl of the gradient of any scalar field ψ is always zero, i.e. $\nabla \times (\nabla \psi) = 0$, and therefore the divergence of the scalar electric potential ϕ can be defined as

$$\boldsymbol{E} - i\omega \boldsymbol{A} = -\nabla \phi. \tag{2.7}$$

The curl of (2.5) gives

$$\mu \nabla \times \boldsymbol{H} = \nabla \times (\nabla \times \boldsymbol{A}) \tag{2.8}$$

and (2.2) can be substituted into (2.8) as

$$\mu(-i\omega\epsilon \boldsymbol{E} + \boldsymbol{J}) = \nabla(\nabla \cdot \boldsymbol{A}) - \nabla^2 \boldsymbol{A}, \tag{2.9}$$

where ∇^2 is the vector Laplacian operator, i.e., $\nabla^2 \boldsymbol{\psi} = \nabla(\nabla \cdot \boldsymbol{\psi}) - \nabla \times (\nabla \times \boldsymbol{\psi})$. If (2.7) is substituted into (2.9),

$$-i\omega\mu\epsilon(i\omega \boldsymbol{A} - \nabla \phi) + \mu \boldsymbol{J} = \nabla(\nabla \cdot \boldsymbol{A}) - \nabla^2 \boldsymbol{A}, \tag{2.10}$$

where the expression can be rearranged as

$$\nabla^2 \boldsymbol{A} + \omega^2 \mu\epsilon \boldsymbol{A} - \nabla(\nabla \cdot \boldsymbol{A} - i\omega\mu\epsilon\phi) = -\mu \boldsymbol{J}. \tag{2.11}$$

The Hemholtz theorem states that a vector field is determined if both its divergence and its curl are specified everywhere. Therefore $\nabla \cdot \boldsymbol{A}$ can be defined in any way. The Lorentz gauge chooses

$$\nabla \cdot \boldsymbol{A} = i\omega\mu\epsilon\phi \tag{2.12}$$

and by substituting (2.12) into (2.11), the non-homogeneous Helmholtz's equation for the vector magnetic potential is derived as

$$\nabla^2 \boldsymbol{A} + \omega^2 \mu\epsilon \boldsymbol{A} = -\mu \boldsymbol{J}. \tag{2.13}$$

Similarly, the divergence of (2.7) gives

$$\nabla \cdot \boldsymbol{E} - i\omega \nabla \cdot \boldsymbol{A} = -\nabla^2 \phi, \tag{2.14}$$

where ∇^2 is the Laplacian operator, i.e., $\nabla^2\psi = \nabla\cdot\nabla\psi$. Both (2.4) and (2.12) can be substituted into (2.14) to derive the non-homogeneous Helmholtz's equation for the scalar electric potential, i.e,

$$\nabla^2 \phi + \omega^2 \mu\epsilon \phi = -\rho/\epsilon. \tag{2.15}$$

The solution of the scalar Helmholtz's equation also solves the Hemlholtz's equation in Cartesian coordinate system since we can divide the equation in three scalar components as

$$\begin{aligned} \nabla^2 A_x + \omega^2 \mu\epsilon A_x &= -\mu J_x \\ \nabla^2 A_y + \omega^2 \mu\epsilon A_y &= -\mu J_y \\ \nabla^2 A_z + \omega^2 \mu\epsilon A_z &= -\mu J_z. \end{aligned} \tag{2.16}$$

For a point source at the origin, a general form of the scalar Helmhotz's equation in the spherical coordinates, i.e., $r = |x^2 + y^2 + z^2|$, can be written as

$$\nabla^2 \psi(r) + k^2 \psi(r) = -\delta(r) \tag{2.17}$$

and its general solution is

$$\psi(r) = \frac{c_1}{r} e^{ikr} + \frac{c_2}{r} e^{-ikr}. \tag{2.18}$$

Physically, $\frac{c_1}{r}e^{ikr}$ and $\frac{c_2}{r}e^{-ikr}$ represents the outgoing and incoming waves, respectively, and therefore c_2 can be chosen as zero for imposing a radiating source. The particular solution involving a point source at the origin yields[1] $c_1 = 1/4\pi$, and therefore the solution of (2.17) becomes

$$\psi(r) = \frac{e^{ikr}}{4\pi r}. \tag{2.19}$$

[1]See Derivation 1 at the end of Section 2.1.

For any point source at $\boldsymbol{r}'$ and observation point at $\boldsymbol{r}$ in the Cartesian coordinates, the solution of the scalar Helmholtz's equation, i.e., Green's function, is

$$\psi(\boldsymbol{r}) = g(\boldsymbol{r}, \boldsymbol{r}') = \frac{e^{ik|\boldsymbol{r}-\boldsymbol{r}'|}}{4\pi|\boldsymbol{r}-\boldsymbol{r}'|}. \tag{2.20}$$

The physical interpretation of the Green's function is depicted in Fig. 2.1.

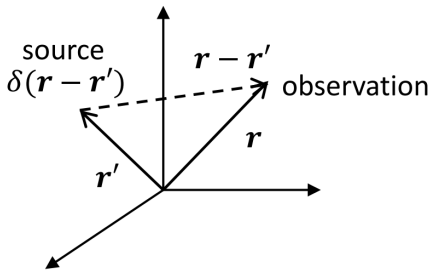

Figure 2.1: For a point source located at $\boldsymbol{r}'$, the Green's function of the scalar Helmholtz's equation observes the field at a point $\boldsymbol{r}$.

For an arbitrary source distribution σ, the solution in (2.20) can be superposed for each infinitesimal point source with an integral and the solution can be written as

$$\psi(\boldsymbol{r}) = \int_V d\boldsymbol{r}'\, g(\boldsymbol{r}, \boldsymbol{r}')\sigma(\boldsymbol{r}'), \tag{2.21}$$

where V is the whole space. The physical interpretation of the integration is depicted in Fig. 2.2.

The solutions of the Helmholtz's equations for the vector magnetic potential and the scalar electric potential can be written as

$$\boldsymbol{A}(\boldsymbol{r}) = \mu \int_V d\boldsymbol{r}'\, g(\boldsymbol{r}, \boldsymbol{r}')\boldsymbol{J}(\boldsymbol{r}') \tag{2.22}$$

and

$$\phi(\boldsymbol{r}) = \frac{1}{\epsilon} \int_V d\boldsymbol{r}'\, g(\boldsymbol{r}, \boldsymbol{r}')\phi(\boldsymbol{r}'), \tag{2.23}$$

respectively.

In order to derive the electric field $\boldsymbol{E}$ directly from the current source $\boldsymbol{J}$, (2.22) and (2.23) can be substituted into (2.14) and the electric-field integral can

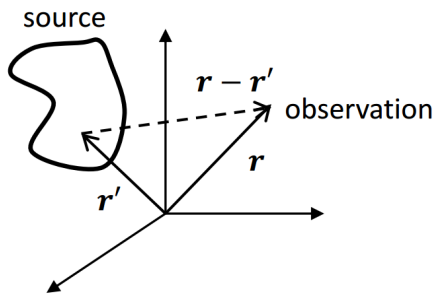

Figure 2.2: For an arbitrary source distribution σ, the superposition of the Green's function and each infinitesimal point source observes the field at $\boldsymbol{r}$.

be derived as

$$\begin{aligned}\boldsymbol{E}(\boldsymbol{r}) =& i\omega\boldsymbol{A}(\boldsymbol{r}) - \nabla\phi(\boldsymbol{r}) \\ =& i\omega\mu\int_V d\boldsymbol{r}'\, g(\boldsymbol{r},\boldsymbol{r}')\boldsymbol{J}(\boldsymbol{r}') - \frac{\nabla}{\epsilon}\int_V d\boldsymbol{r}'\, g(\boldsymbol{r},\boldsymbol{r}')\rho(\boldsymbol{r}') \\ =& i\omega\mu\int_V d\boldsymbol{r}'\, g(\boldsymbol{r},\boldsymbol{r}')\boldsymbol{J}(\boldsymbol{r}') - \frac{\nabla}{i\omega\epsilon}\int_V d\boldsymbol{r}'\, g(\boldsymbol{r},\boldsymbol{r}')\nabla'\cdot\boldsymbol{J}(\boldsymbol{r}').\end{aligned} \tag{2.24}$$

Using the property[2] $g(\boldsymbol{r},\boldsymbol{r}')\nabla'\cdot\boldsymbol{J}(\boldsymbol{r}') = -\nabla g(\boldsymbol{r},\boldsymbol{r}')\cdot\boldsymbol{J}(\boldsymbol{r}')$,

$$\begin{aligned}\boldsymbol{E}(\boldsymbol{r}) &= i\omega\mu\int_V d\boldsymbol{r}'\, g(\boldsymbol{r},\boldsymbol{r}')\boldsymbol{J}(\boldsymbol{r}') + \frac{\nabla}{i\omega\epsilon}\int_V d\boldsymbol{r}'\,\nabla g(\boldsymbol{r},\boldsymbol{r}')\cdot\boldsymbol{J}(\boldsymbol{r}') \\ &= i\omega\mu\left[\overline{\boldsymbol{I}} + \frac{\nabla\nabla}{k^2}\right]\cdot\int_V d\boldsymbol{r}'\, g(\boldsymbol{r},\boldsymbol{r}')\boldsymbol{J}(\boldsymbol{r}'),\end{aligned} \tag{2.25}$$

where $k = \omega\sqrt{\mu\epsilon}$ is the wavenumber,

$$\overline{\boldsymbol{I}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.26}$$

is the unit dyad, and

$$\nabla\nabla = \begin{pmatrix} \frac{\partial}{\partial x}\frac{\partial}{\partial x} & \frac{\partial}{\partial x}\frac{\partial}{\partial y} & \frac{\partial}{\partial x}\frac{\partial}{\partial z} \\ \frac{\partial}{\partial y}\frac{\partial}{\partial x} & \frac{\partial}{\partial y}\frac{\partial}{\partial y} & \frac{\partial}{\partial y}\frac{\partial}{\partial z} \\ \frac{\partial}{\partial z}\frac{\partial}{\partial x} & \frac{\partial}{\partial z}\frac{\partial}{\partial y} & \frac{\partial}{\partial z}\frac{\partial}{\partial z} \end{pmatrix} \tag{2.27}$$

is the outer product of two del operators. As a more compact form, EFIE can be written as

$$\boldsymbol{E}(\boldsymbol{r}) = i\omega\mu\int_V d\boldsymbol{r}'\,\overline{\boldsymbol{G}}(\boldsymbol{r},\boldsymbol{r}')\cdot\boldsymbol{J}(\boldsymbol{r}'), \tag{2.28}$$

[2]See Derivation 2 at the end of Section 2.1.

where

$$\overline{\boldsymbol{G}}(\boldsymbol{r},\boldsymbol{r}') = \left[\overline{\boldsymbol{I}} + \frac{\nabla\nabla}{k^2}\right] g(\boldsymbol{r},\boldsymbol{r}') \tag{2.29}$$

is the dyadic Green's function.

Similarly, in order to derive $\boldsymbol{H}$ directly from the current source $\boldsymbol{J}$, (2.22) can be substituted into (2.5) to derive the magnetic-field integral as

$$\begin{aligned}
\boldsymbol{H}(\boldsymbol{r}) &= \frac{1}{\mu}\nabla \times \boldsymbol{A}(\boldsymbol{r}) \\
&= \nabla \times \int_V d\boldsymbol{r}'\, g(\boldsymbol{r},\boldsymbol{r}')\boldsymbol{J}(\boldsymbol{r}') \\
&= \int_V d\boldsymbol{r}'\, \nabla g(\boldsymbol{r},\boldsymbol{r}') \times \boldsymbol{J}(\boldsymbol{r}') \\
&= \int_V d\boldsymbol{r}'\, \boldsymbol{J}(\boldsymbol{r}') \times \nabla' g(\boldsymbol{r},\boldsymbol{r}').
\end{aligned} \tag{2.30}$$

To summarize, the electric-field and the magnetic-field integrals can be written as,

$$\boldsymbol{E}(\boldsymbol{r}) = i\omega\mu \int_V d\boldsymbol{r}'\, \overline{\boldsymbol{G}}(\boldsymbol{r},\boldsymbol{r}') \cdot \boldsymbol{J}(\boldsymbol{r}') \tag{2.31}$$

and

$$\boldsymbol{H}(\boldsymbol{r}) = \int_V d\boldsymbol{r}'\, \boldsymbol{J}(\boldsymbol{r}') \times \nabla' g(\boldsymbol{r},\boldsymbol{r}'), \tag{2.32}$$

respectively, where

$$\overline{\boldsymbol{G}}(\boldsymbol{r},\boldsymbol{r}') = \left[\overline{\boldsymbol{I}} + \frac{\nabla\nabla}{k^2}\right] g(\boldsymbol{r},\boldsymbol{r}') \tag{2.33}$$

is the dyadic Green's function and

$$g(\boldsymbol{r},\boldsymbol{r}') = \frac{e^{ik|\boldsymbol{r}-\boldsymbol{r}'|}}{4\pi|\boldsymbol{r}-\boldsymbol{r}'|} \tag{2.34}$$

is the Green's function for the scalar Helmholtz's equation in Cartesian coordinates. With the integrals, the electric field $\boldsymbol{E}$ and the magnetic field $\boldsymbol{H}$ resulting from the source $\boldsymbol{J}$ can be determined everywhere directly without using any auxiliary field, i.e. potential functions $\boldsymbol{A}$ and ϕ, as depicted in Fig. 2.3.

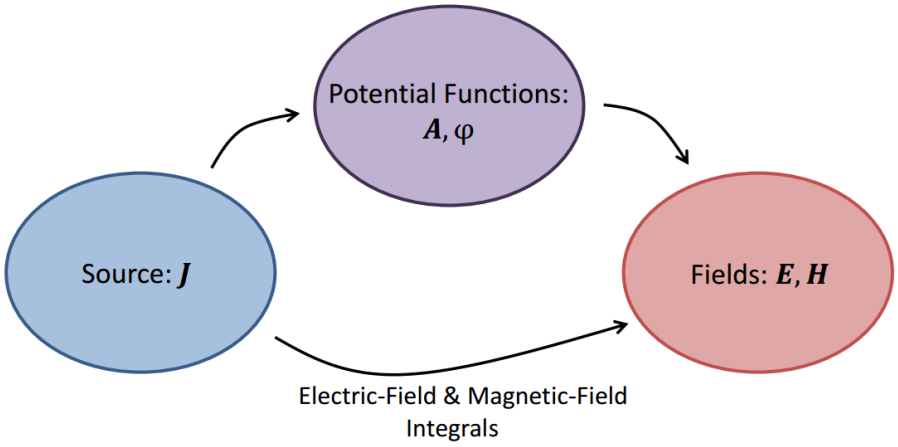

Figure 2.3: Electric and magnetic fields can be obtained directly, i.e., without using any auxiliary fields, with the electric-field and the magnetic-field integrals, respectively.

Derivation 1: Integrating both sides of (2.17) in a volume V with the surface S, where S is a hypothetical sphere with the radius r as depicted in Fig. 2.4, yields

$$\begin{aligned} \int_V d\boldsymbol{r}'\, \nabla'^2 \psi(\boldsymbol{r}') + k^2 \int_V d\boldsymbol{r}'\, \psi(\boldsymbol{r}') &= -\int_V d\boldsymbol{r}' \delta(\boldsymbol{r}') \\ \oint_S d\boldsymbol{r}'\, \nabla' \psi(\boldsymbol{r}') \cdot \hat{\boldsymbol{r}}' + 4\pi k^2 \int_0^r dr'\, \psi(r') &= -1. \end{aligned} \tag{2.35}$$

Substituting $\psi(r) = c_1 e^{ikr}/r$,

$$\begin{aligned} -4\pi r^2 c_1 \left(\frac{e^{ikr}}{r^2} - ik\frac{e^{ikr}}{r} \right) + 4\pi k^2 c_1 \left(\frac{re^{ikr}}{ik} + \frac{e^{ikr}-1}{k^2} \right) &= -1 \\ r \to 0 \Rightarrow c_1 = \frac{1}{4\pi}. & \end{aligned} \tag{2.36}$$

Derivation 2: From the vector identity $\nabla \cdot (\psi \boldsymbol{A}) = \psi \nabla \cdot \boldsymbol{A} + \boldsymbol{A} \cdot \nabla \psi$ in the prime coordinates,

$$\begin{aligned} \nabla' \cdot (g(\boldsymbol{r}, \boldsymbol{r}') \boldsymbol{J}(\boldsymbol{r}')) &= g(\boldsymbol{r}, \boldsymbol{r}') \nabla' \cdot \boldsymbol{J}(\boldsymbol{r}') + \boldsymbol{J}(\boldsymbol{r}') \cdot \nabla' g(\boldsymbol{r}, \boldsymbol{r}') \\ \int_V d\boldsymbol{r}'\, \nabla' \cdot (g(\boldsymbol{r}, \boldsymbol{r}') \boldsymbol{J}(\boldsymbol{r}')) &= \int_V d\boldsymbol{r}'\, g(\boldsymbol{r}, \boldsymbol{r}') \nabla' \cdot \boldsymbol{J}(\boldsymbol{r}') + \boldsymbol{J}(\boldsymbol{r}') \cdot \nabla' g(\boldsymbol{r}, \boldsymbol{r}') \\ \oint_S d\boldsymbol{r}'\, g(\boldsymbol{r}, \boldsymbol{r}') \boldsymbol{J}(\boldsymbol{r}') \cdot \hat{\boldsymbol{n}} &= \int_V d\boldsymbol{r}'\, g(\boldsymbol{r}, \boldsymbol{r}') \nabla' \cdot \boldsymbol{J}(\boldsymbol{r}') + \boldsymbol{J}(\boldsymbol{r}') \cdot \nabla' g(\boldsymbol{r}, \boldsymbol{r}'). \end{aligned} \tag{2.37}$$

Let V be a volume which covers the whole source and the surface S does not intersect with the source, i.e. $\boldsymbol{J}(\boldsymbol{r'}) = 0 \; \forall \boldsymbol{r'} \in S$ as depicted in Fig. 2.4, then

$$\oint_S d\boldsymbol{r}' \, g(\boldsymbol{r}, \boldsymbol{r}') \boldsymbol{J}(\boldsymbol{r}') \cdot \hat{\boldsymbol{n}} = 0, \tag{2.38}$$

and therefore,

$$\begin{aligned} g(\boldsymbol{r}, \boldsymbol{r}') \nabla' \cdot \boldsymbol{J}(\boldsymbol{r}') &= -\boldsymbol{J}(\boldsymbol{r}') \cdot \nabla' g(\boldsymbol{r}, \boldsymbol{r}') \\ &= -\nabla g(\boldsymbol{r}, \boldsymbol{r}') \cdot \boldsymbol{J}(\boldsymbol{r}'). \end{aligned} \tag{2.39}$$

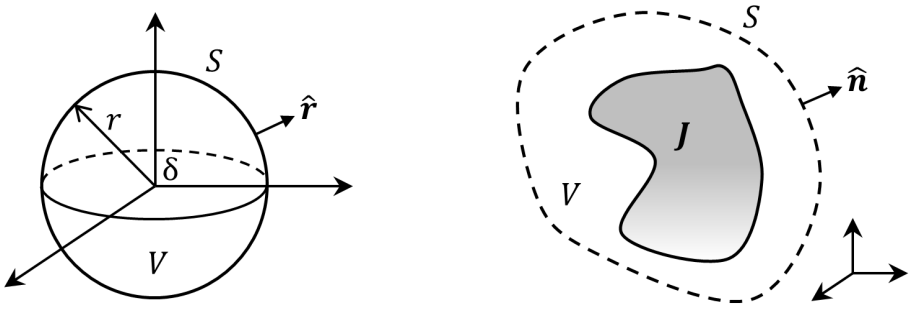

Figure 2.4: Two hypothetical surfaces: the one in the left is a sphere centered at the origin and the one in the right has arbitrary shape covering the whole source.

2.2 Equivalence Principle and Scattering

The extended Maxwell's equation with fictitious magnetic sources can be written as

$$\nabla \times \boldsymbol{E}(\boldsymbol{r}) = i\omega\mu \boldsymbol{H}(\boldsymbol{r}) - \boldsymbol{M}(\boldsymbol{r}) \tag{2.40}$$

$$\nabla \times \boldsymbol{H}(\boldsymbol{r}) = -i\omega\epsilon \boldsymbol{E}(\boldsymbol{r}) + \boldsymbol{J}(\boldsymbol{r}) \tag{2.41}$$

$$\nabla \cdot \boldsymbol{H}(\boldsymbol{r}) = -\rho_m(\boldsymbol{r})/\mu \tag{2.42}$$

$$\nabla \cdot \boldsymbol{E}(\boldsymbol{r}) = \rho(\boldsymbol{r})/\epsilon, \tag{2.43}$$

where $\boldsymbol{M}(\boldsymbol{r})$ is the magnetic current density and $\rho_m(\boldsymbol{r})$ is the magnetic charge density.

The boundary conditions for the configuration in Fig. 2.5 are

$$\begin{aligned} \hat{\boldsymbol{n}} \times (\boldsymbol{E}_1 - \boldsymbol{E}_2) &= -\boldsymbol{M}_S \\ \hat{\boldsymbol{n}} \times (\boldsymbol{H}_1 - \boldsymbol{H}_2) &= \boldsymbol{J}_S, \end{aligned} \tag{2.44}$$

where $\hat{\boldsymbol{n}}$ is the outward unit-normal vector of the boundary S, $\boldsymbol{E}_1$ and $\boldsymbol{H}_1$ are the electric and magnetic fields in Medium 1 and S, $\boldsymbol{E}_2$ and $\boldsymbol{H}_2$ are the electric and magnetic fields in Medium 2, and $\boldsymbol{M}_S$ and $\boldsymbol{J}_S$ are the magnetic and electric current densities on the surface S, respectively.

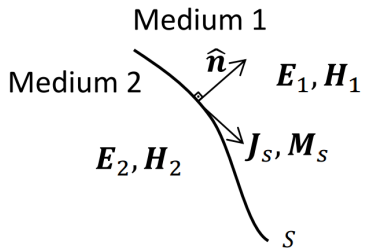

Figure 2.5: Medium 1 and Medium 2 are divided by the surface S.

The equivalence principle states that when two source specifications give the same solution in a limited region of interest, the two problems are equivalent [30],[31]. Two equivalent problems are depicted in Fig. 2.6 as an example. The single arrows shows the electric currents and the double arrows shows the magnetic currents. The original problem on the left-hand side has sources and infinite conductor and an equivalent problem on the right-hand side has the original sources and their images according to the image theory. The problems have different specifications but they are equivalent, i.e., the fields are equal, in the left half space.

The Huygen's principle is depicted in Fig. 2.7 as an example of the surface equivalence formulations. The original configuration on the left-hand side has radiating sources. Equivalent currents on a hypothetical surface S which encloses the sources are used to imitate the original sources in their absence. The uniqueness theorem states that the solution is unique in a volume enclosed by surface S if the boundary conditions are specified to satisfy one the uniqueness conditions

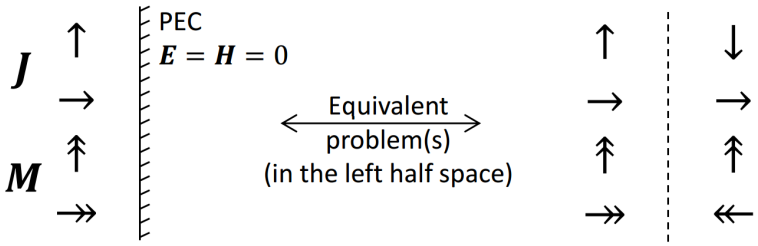

Figure 2.6: The problems in the left and the right hand sides are equivalent in the left half space, i.e., the fields are equal in the left half space.

1. tangential $\boldsymbol{E}$ is specified over the whole surface S, or
2. tangential $\boldsymbol{H}$ is specified over the whole surface S, or
3. tangential $\boldsymbol{E}$ is specified over a part of S, and
   tangential $\boldsymbol{H}$ is specified over the rest of S.

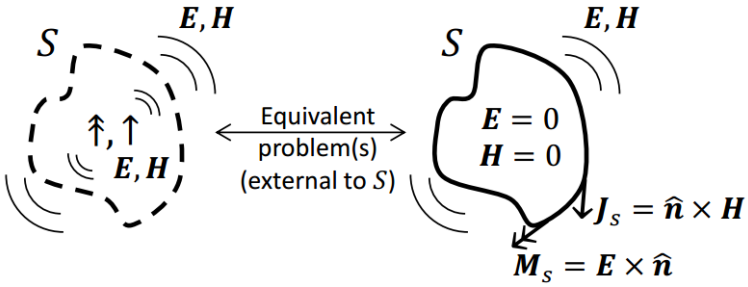

Figure 2.7: The problems in the left and the right hand sides of the figure are equivalent external to S.

A general formulation for the surface equivalence principle is depicted in Fig. 2.8. Problems a) and b) are two original problems and problems c) and d) are equivalent problems of the original problems. All problems have different dielectric properties and source distribution but they have the same boundary surface S. With the equivalence principle, four equivalence relationship can be formed. For example, problem c) is equivalent to the problem a) external to S since the fields are equal outside S in both problems. Similarly, Problem c) is equivalent to the problem b) internal to S since the fields are equal inside S in both problems.

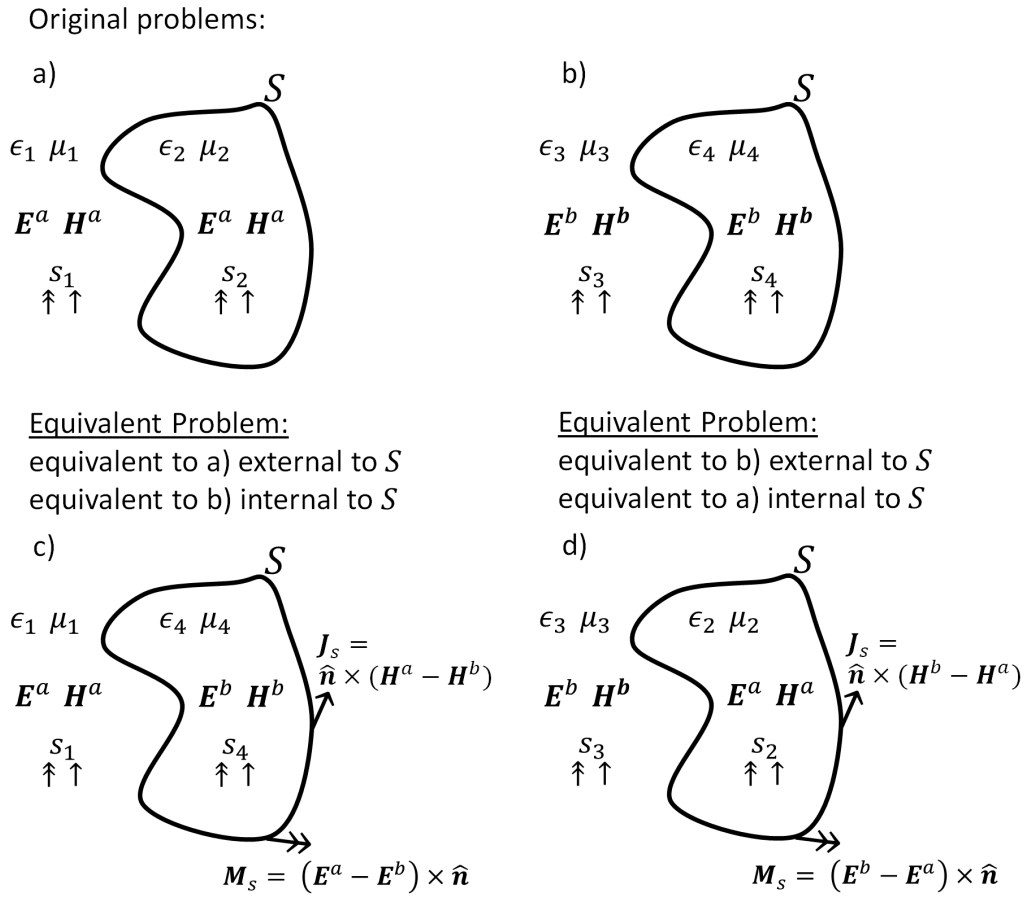

Figure 2.8: Many equivalence relations can be formed with two original problems.

The surface equivalent principle is employed for solving electromagnetic scattering problems. Consider the original problem involving electric and magnetic current sources S_1 and a dielectric object with a boundary surface S in free space on the left-hand side of Fig. 2.9. The object has an electric permittivity of ϵ_1 and a magnetic permeability of μ_1. $\boldsymbol{E}$ and $\boldsymbol{H}$ are the electric and magnetic fields everywhere. An equivalent problem external to S can be defined, where the fields inside S are zero and the electric permittivity and magnetic permeability are ϵ_0 and μ_0, respectively. The fields inside can be chosen arbitrarily, but the fields should be Maxwellian. By using the boundary conditions in (2.5), equivalent currents on the surface S can be written as $\boldsymbol{J}_S = \hat{\boldsymbol{n}} \times \boldsymbol{H}$ and $\boldsymbol{M}_S = \boldsymbol{E} \times \hat{\boldsymbol{n}}$. A similar treatment can be made for the problems involving perfect-electric-conductor (PEC) geometries as in Fig. 2.10.

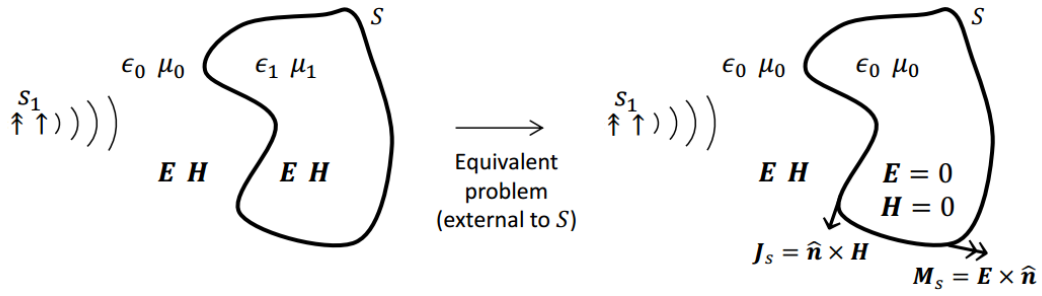

Figure 2.9: An equivalent problem for dielectric object is formulated using the surface equivalence principle.

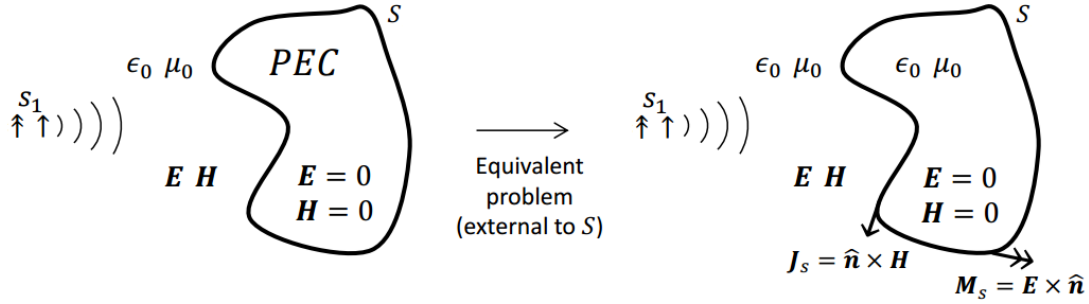

Figure 2.10: An equivalent problem for PEC object is formulated using the surface equivalence principle.

The total fields can be considered as sum of the incident fields and the scattered fields, i.e., $\boldsymbol{E} = \boldsymbol{E}^{inc} + \boldsymbol{E}^{sca}$. The incident field $\boldsymbol{E}^{inc}(S_1)$ is originated by the sources S_1 and the scattered field $\boldsymbol{E}^{sca}(\boldsymbol{J}_S, \boldsymbol{M}_S)$ is originated by the equivalent currents. Similarly, the total magnetic field can be written as. $\boldsymbol{H} = \boldsymbol{H}^{inc} + \boldsymbol{H}^{sca}$. The surface currents are unknown and their solution gives the scattered fields $\boldsymbol{E}^{sca}$ and $\boldsymbol{H}^{sca}$, and hence the total fields $\boldsymbol{E}$ and $\boldsymbol{H}$ outside of S.

For the rest of the thesis, PEC geometries and electric sources are considered, therefore fictitious magnetic sources vanish. The equivalent electric current $\boldsymbol{J}_S$ on the boundary S will be denoted as $\boldsymbol{J}$ for simplicity. The electric-field integral equation (EFIE) and the magnetic-field integral equation (MFIE) are to be derived for solving the unknown current. Choosing $\epsilon_1 = \epsilon_0$ and $\mu_1 = \mu_0$ provides a homogeneous medium everywhere so that the electric-field and magnetic-field

integrals in (2.31) and (2.32) can be employed for solution of the surface currents.

Boundary conditions state that the tangential electric field on a surface is zero, i.e.,

$$\hat{\boldsymbol{t}} \cdot \boldsymbol{E}(\boldsymbol{r}) = 0, \boldsymbol{r} \in S \tag{2.45}$$

where $\hat{\boldsymbol{t}} \cdot \boldsymbol{E}$ is a tangential operator which finds the tangential component of $\boldsymbol{E}$ on the surface S. The total field can be written as a sum of the incident and scattered fields as

$$\begin{aligned} \hat{\boldsymbol{t}} \cdot (\boldsymbol{E}^{inc}(\boldsymbol{r}) + \boldsymbol{E}^{sca}(\boldsymbol{r})) = 0,\ \boldsymbol{r} \in S \\ \hat{\boldsymbol{t}} \cdot \boldsymbol{E}^{sca}(\boldsymbol{r}) = -\hat{\boldsymbol{t}} \cdot \boldsymbol{E}^{inc}(\boldsymbol{r}),\ \boldsymbol{r} \in S. \end{aligned} \tag{2.46}$$

The scattered electric field $\boldsymbol{E}^{sca}$ due to the unknown surface current $\boldsymbol{J}$ can be found using the electric-field integral (2.31), and integral can be substituted as

$$i\omega\mu\hat{\boldsymbol{t}} \cdot \int_V d\boldsymbol{r}'\, \overline{\boldsymbol{G}}(\boldsymbol{r}, \boldsymbol{r}') \cdot \boldsymbol{J}(\boldsymbol{r}') = -\hat{\boldsymbol{t}} \cdot \boldsymbol{E}^{inc}(\boldsymbol{r}),\ \boldsymbol{r} \in S. \tag{2.47}$$

Similarly, the boundary conditions for magnetic field states that

$$\begin{aligned} \hat{\boldsymbol{n}} \times \boldsymbol{H}(\boldsymbol{r}) = \boldsymbol{J}(\boldsymbol{r}), \boldsymbol{r} \in S \\ \hat{\boldsymbol{n}} \times (\boldsymbol{H}^{inc}(\boldsymbol{r}) + \boldsymbol{H}^{sca}(\boldsymbol{r})) = \boldsymbol{J}(\boldsymbol{r}), \boldsymbol{r} \in S \\ \boldsymbol{J}(\boldsymbol{r}) - \hat{\boldsymbol{n}} \times \boldsymbol{H}^{sca}(\boldsymbol{r}) = \hat{\boldsymbol{n}} \times \boldsymbol{H}^{inc}(\boldsymbol{r}), \boldsymbol{r} \in S \end{aligned} \tag{2.48}$$

and the scattered magnetic field $\boldsymbol{H}^{sca}$ due to the unknown surface current $\boldsymbol{J}$ can be found using the magnetic-field integral (2.31), and integral can be substituted as

$$\boldsymbol{J}(\boldsymbol{r}) - \hat{\boldsymbol{n}} \times \int_V d\boldsymbol{r}'\, \boldsymbol{J}(\boldsymbol{r}') \times \nabla' g(\boldsymbol{r}, \boldsymbol{r}') = \hat{\boldsymbol{n}} \times \boldsymbol{H}^{inc}(\boldsymbol{r}), \boldsymbol{r} \in S. \tag{2.49}$$

To summarize, EFIE and MFIE for the surface current $\boldsymbol{J}$ on S can be written as

$$\hat{\boldsymbol{t}} \cdot \int_S d\boldsymbol{r}'\, \overline{\boldsymbol{G}}(\boldsymbol{r}, \boldsymbol{r}') \cdot \boldsymbol{J}(\boldsymbol{r}') = \frac{i}{k\eta} \hat{\boldsymbol{t}} \cdot \boldsymbol{E}^{inc}(\boldsymbol{r}) \tag{2.50}$$

and

$$\boldsymbol{J}(\boldsymbol{r}) - \hat{\boldsymbol{n}} \times \int_S d\boldsymbol{r}'\, \boldsymbol{J}(\boldsymbol{r}') \times \nabla' g(\boldsymbol{r}, \boldsymbol{r}') = \hat{\boldsymbol{n}} \times \boldsymbol{H}^{inc}(\boldsymbol{r}), \tag{2.51}$$

respectively. Everything except $\boldsymbol{J}$ is known and the solution of the unknown current gives the scattered field, hence the total fields $\boldsymbol{E}$ and $\boldsymbol{H}$. EFIE can be used to formulate the scattering problems involving open and closed surfaces while MFIE can only be used for closed surfaces.

2.3 The Method of Moments

The method of moments (MoM) discretizes the surface current $\boldsymbol{J}(\boldsymbol{r})$ using known basis functions as

$$\sum_{n=1}^{N} x_n \boldsymbol{b}_n(\boldsymbol{r}) \approx \boldsymbol{J}(\boldsymbol{r}), \tag{2.52}$$

where N is the number of basis functions, and $\boldsymbol{b}_n(\boldsymbol{r})$ and x_n is the n^{th} basis function and its coefficient, respectively. EFIE and can be discretized by substituting (2.52) into (2.50) as

$$\hat{\boldsymbol{t}} \cdot \int_{S'} d\boldsymbol{r}' \, \overline{\boldsymbol{G}}(\boldsymbol{r}, \boldsymbol{r}') \cdot \sum_{n=1}^{N} x_n \boldsymbol{b}_n(\boldsymbol{r}') = \frac{i}{k\eta} \hat{\boldsymbol{t}} \cdot \boldsymbol{E}^{inc}(\boldsymbol{r}) \tag{2.53}$$

and the summation can be carried outside of the integration as

$$\hat{\boldsymbol{t}} \cdot \sum_{n=1}^{N} x_n \int_{S_n} d\boldsymbol{r}' \, \overline{\boldsymbol{G}}(\boldsymbol{r}, \boldsymbol{r}') \cdot \boldsymbol{b}_n(\boldsymbol{r}') = \frac{i}{k\eta} \hat{\boldsymbol{t}} \cdot \boldsymbol{E}^{inc}(\boldsymbol{r}), \tag{2.54}$$

where $S_n \subset S'$ is the n^{th} basis function domain.

MoM tests (2.54) using M testing functions as

$$\int_{S_m} d\boldsymbol{r} \, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \sum_{n=1}^{N} x_n \int_{S_n} d\boldsymbol{r}' \, \overline{\boldsymbol{G}}(\boldsymbol{r}, \boldsymbol{r}') \cdot \boldsymbol{b}_n(\boldsymbol{r}') = \frac{i}{k\eta} \int_{S_m} d\boldsymbol{r} \, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{E}^{inc}(\boldsymbol{r}), \tag{2.55}$$

and the summation can be carried outside of the testing integration as

$$\sum_{n=1}^{N} x_n \int_{S_m} d\boldsymbol{r} \, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \int_{S_n} d\boldsymbol{r}' \, \overline{\boldsymbol{G}}(\boldsymbol{r}, \boldsymbol{r}') \cdot \boldsymbol{b}_n(\boldsymbol{r}') = \frac{i}{k\eta} \int_{S_m} d\boldsymbol{r} \, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{E}^{inc}(\boldsymbol{r}), \tag{2.56}$$

where $S_m \subset S$ is the m^{th} testing function domain. Note that the tangential operators in (2.54) vanish since the testing functions are already tangential to the descretized surface S. For a more compact form, (2.56) can be written as

$$\sum_{n=1}^{N} Z_{mn}^{EFIE} x_n = v_m^{EFIE}, \tag{2.57}$$

where

$$Z_{mn}^{EFIE} = \int_{S_m} d\boldsymbol{r} \, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \int_{S_n} d\boldsymbol{r}' \, \overline{\boldsymbol{G}}(\boldsymbol{r}, \boldsymbol{r}') \cdot \boldsymbol{b}_n(\boldsymbol{r}') \tag{2.58}$$

is the interaction of the m^{th} testing and n^{th} basis functions, in other words, testing of the incoming field originated by the n^{th} basis function on the m^{th} testing function, and

$$v_m^{EFIE} = \frac{i}{k\eta} \boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{E}^{inc}(\boldsymbol{r}) \tag{2.59}$$

is the testing of the incident field on the m^{th} testing function.

The expression in (2.57) constitutes an $M \times N$ linear system of equations in a general form of

$$\overline{\boldsymbol{Z}} \cdot \boldsymbol{x} = \boldsymbol{v}, \tag{2.60}$$

where $\overline{\boldsymbol{Z}}$ is the known impedance (or interaction) matrix, $\boldsymbol{v}$ is the known testing vector of the incident field, and $\boldsymbol{x}$ is the unknown basis-coefficient vector.

Similarly, MFIE can be discretized by substituting (2.52) into (2.51) and tested on the testing functions, and written in a compact form as

$$\sum_{n=1}^{N} Z_{mn}^{MFIE} x_n = v_m^{MFIE}, \tag{2.61}$$

where

$$Z_{mn}^{MFIE} = \int_{S_m} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{b}_n(\boldsymbol{r}) - \int_{S_m} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \hat{\boldsymbol{n}} \times \int_{S_n} d\boldsymbol{r}'\, \boldsymbol{b}_n(\boldsymbol{r}') \times \nabla' g(\boldsymbol{r}, \boldsymbol{r}') \tag{2.62}$$

and

$$v_m^{MFIE} = \int_{S_m} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \hat{\boldsymbol{n}} \times \boldsymbol{H}^{inc}(\boldsymbol{r}). \tag{2.63}$$

The EFIE and MFIE formulations can be used in a combined form as

$$\sum_{n=1}^{N} Z_{mn}^{CFIE} x_n = v_m^{CFIE}, \tag{2.64}$$

where

$$Z_{mn}^{CFIE} = \alpha Z_{mn}^{EFIE} + (1-\alpha)\frac{i}{k} Z_{mn}^{MFIE} \tag{2.65}$$

and

$$v_m^{CFIE} = \alpha v_m^{EFIE} + (1-\alpha)\frac{i}{k} v_m^{MFIE} \tag{2.66}$$

with the combination factor $\alpha \in [0,\, 1]$. If $\alpha = 1$ or $\alpha = 0$, the system is formulated with EFIE or MFIE, respectively. MFIE elements are multiplied with i/k in order

to weight the equations equally for the linear combination. The systems obtained by using CFIE are generally better conditioned than the systems obtained using EFIE or MFIE [32],[9].

The conventional MoM uses Rao-Wilton-Glisson (RWG) basis and testing functions with Galerkin formulation [2]. The Galerkin formulation chooses the basis and testing functions are the same, i.e., $\boldsymbol{t}_n = \boldsymbol{b}_n$, and therefore $\overline{\boldsymbol{Z}}$ in (2.60) is an $N \times N$ matrix. The EFIE formulation yields a symmetric matrix, i.e., $Z_{mn}^{EFIE} = Z_{nm}^{EFIE}$, where it is sufficient to store half of the elements, but the MFIE (and therefore CFIE) formulation does not have such property. The resulting MoM system can be solved iteratively in $\mathcal{O}(N^2)$ time and memory via MVMs.

An RWG function $\boldsymbol{b}_n(\boldsymbol{r})$ is defined as

$$\boldsymbol{b}_n(\boldsymbol{r}) = \begin{cases} \boldsymbol{\rho}_n^{\pm}(\boldsymbol{r}) & \text{if } \boldsymbol{r} \in T_n^{\pm} \\ 0 & \text{else,} \end{cases} \tag{2.67}$$

where T_n^+ and T_n^- are two adjacent planar triangles with their areas A_n^+ and A_n^-, respectively, l_n is their common side-length, and

$$\boldsymbol{\rho}_n^{\pm}(\boldsymbol{r}) = \pm \frac{l_n}{2A_n^{\pm}} (\boldsymbol{r} - \boldsymbol{r}_n^{\pm}). \tag{2.68}$$

Note that $\boldsymbol{b}_n(\boldsymbol{r}) = 0$ if $\boldsymbol{r} \notin S_n = T_n^+ \cup T_n^-$.

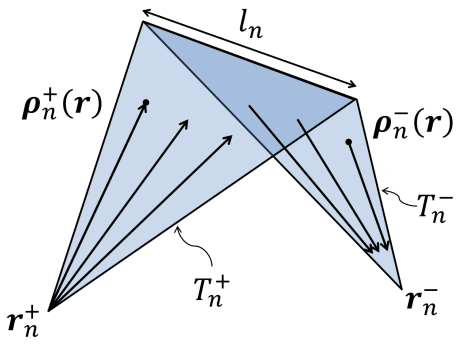

Figure 2.11: RWG basis (and testing) function.

An important property of RWG functions is that their divergence is finite everywhere, i.e.,

$$\nabla \cdot \boldsymbol{b}_n(\boldsymbol{r}) = \begin{cases} \pm l_n / A_n^{\pm} & \text{if } \boldsymbol{r} \in T_n^{\pm} \\ 0 & \text{else,} \end{cases} \tag{2.69}$$

and therefore the total charge is zero, i.e., $A_n^+ l_n / A_n^+ - A_n^- l_n / A_n^- = 0$. This property can be used to extract the higher-order singularity arises from $\nabla g(r, r')$ term in EFIE. We can write Z_{mn}^{EFIE} as

$$\begin{aligned} Z_{mn}^{EFIE} &= \int_{S_m} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \int_{S_n} d\boldsymbol{r}'\, \overline{\boldsymbol{G}}(\boldsymbol{r}, \boldsymbol{r}') \cdot \boldsymbol{b}_n(\boldsymbol{r}') \\ &= \int_{S_m} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \int_{S_n} d\boldsymbol{r}'\, \left[\overline{\boldsymbol{I}} + \frac{\nabla\nabla}{k^2} \right] g(\boldsymbol{r}, \boldsymbol{r}') \cdot \boldsymbol{b}_n(\boldsymbol{r}') \\ &= \int_{S_m} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \int_{S_n} d\boldsymbol{r}'\, g(\boldsymbol{r}, \boldsymbol{r}') \boldsymbol{b}_n(\boldsymbol{r}') \\ &+ \int_{S_m} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \int_{S_n} d\boldsymbol{r}'\, \frac{\nabla\nabla}{k^2} g(\boldsymbol{r}, \boldsymbol{r}') \cdot \boldsymbol{b}_n(\boldsymbol{r}') \end{aligned} \tag{2.70}$$

and two del operators can be carried on to the testing and basis functions as

$$\begin{aligned} Z_{mn}^{EFIE} = &\int_{S_m} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \int_{S_n} d\boldsymbol{r}'\, g(\boldsymbol{r}, \boldsymbol{r}') \boldsymbol{b}_n(\boldsymbol{r}') \\ &- \frac{1}{k^2} \int_{S_m} d\boldsymbol{r}\, \nabla \cdot \boldsymbol{t}_m(\boldsymbol{r}) \int_{S_n} d\boldsymbol{r}'\, g(\boldsymbol{r}, \boldsymbol{r}') \nabla' \cdot \boldsymbol{b}_n(\boldsymbol{r}'). \end{aligned} \tag{2.71}$$

We can substitute 2.69 into 2.71 as

$$\begin{aligned} Z_{ik,jl}^{EFIE} = &\frac{l_{ik} l_{jl}}{4 A_i A_j} \int_{S_i} d\boldsymbol{r}\, (\boldsymbol{r} - \boldsymbol{r}_{ik}) \cdot \int_{S_j} d\boldsymbol{r}'\, g(\boldsymbol{r}, \boldsymbol{r}') (\boldsymbol{r}' - \boldsymbol{r}_{jl}) \\ &- \frac{1}{k^2} \frac{l_{ik} l_{jl}}{A_i A_j} \int_{S_i} d\boldsymbol{r} \int_{S_j} d\boldsymbol{r}'\, g(\boldsymbol{r}, \boldsymbol{r}'). \end{aligned} \tag{2.72}$$

The first order singularity in $g(\boldsymbol{r}, \boldsymbol{r}')$ can be handled using numerical integration techniques [33]. The singularity in Z_{mn}^{MFIE} can be extracted similarly [34].

2.4 The Multilevel Fast Multipole Algorithm

Factorization and diagonalization of Green's function provides flexibility to deal with groups of elements, as opposed to dealing with each pair of individual elements [13]. Each MVM in the iterative solution of a MoM system test the boundary conditions on the testing functions with the incident field coming from the basis functions with $\mathcal{O}(N^2)$ complexity. MLFMA groups the basis and testing functions hierarchically and translates the fields from one group to another through aggregation, translation, and disaggregation steps [10]. Fig. 2.12 depicts the translation of fields from a cluster of basis functions to a cluster of testing functions, where each cluster consists four RWG functions. The radiated fields from the basis functions are aggregated at the center of the basis cluster $\boldsymbol{c'}$, then translated to the center of the testing cluster $\boldsymbol{c}$, and then disaggregated to the testing clusters. The testing functions receives the field and tests the boundary conditions in each iteration.

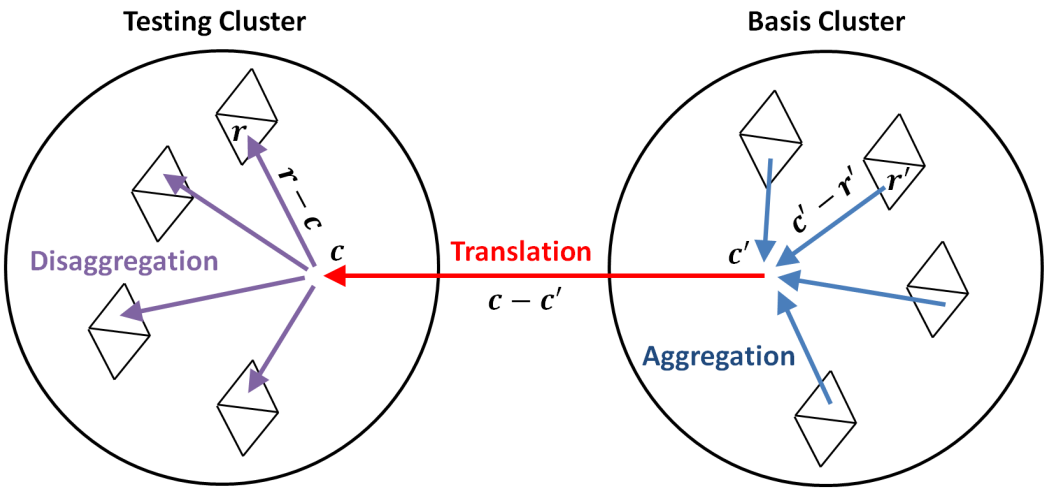

Figure 2.12: Aggregation, translation, and disaggregation scheme of the radiated field from the basis functions.

For aggregating and disaggregating the fields, far-field patterns of the basis (and testing) functions are required, therefore fourier transform of each basis function should be stored [35]. Numerically, patterns are discretized using samples

on a unit sphere and $\mathcal{O}(N)$ samples are stored in order to be used in the iterative solutions.

Parallelization of MLFMA is not trivial due to its sophisticated structure [1]. A basic parallelization of MLFMA is achieved by distributing the basis clusters among processes, but its efficiency is low, since the number of clusters decreases in higher levels. As a remedy, the hybrid parallelization scheme is developed in [17], where the clusters are partitioned in lower levels and field samples in higher levels. In hybrid parallelization, the field samples are accumulated in middle levels and causes inefficiency. To increase the efficiency, a hierarchical parallelization scheme is developed in [18],[19],[35], which partitions both the clusters and the field samples in all levels. The hierarchical partitioning strategy is the most qualified parallelization of MLFMA up to date [21] and a lot of effort is taken for its efficient implementation in the last several years [35], [1]. The developments in this thesis are built upon the efficient parallel MLFMA implementation in courtesy of Prof. Levent Gürel and Dr. Özgür Ergül. The parallel out-of-core implementation (see Chapter 3) is first provided in [24], [25] and [26], and the thesis provides further optimizations and developments on the implementation.

2.5 Physical Optics

Physical optics approximates the surface current $\boldsymbol{J}(\boldsymbol{r})$ as

$$\boldsymbol{J}(\boldsymbol{r}) \approx \boldsymbol{J}^{PO}(\boldsymbol{r}) = \begin{cases} 2\hat{\boldsymbol{n}} \times \boldsymbol{H}^{inc}(\boldsymbol{r}) & \text{if } \boldsymbol{r} \in S_{illum} \\ 0 & \text{else,} \end{cases} \tag{2.73}$$

where $\boldsymbol{J}^{PO}(\boldsymbol{r})$ is the approximate physical optics current and $\boldsymbol{H}^{inc}(\boldsymbol{r})$ is the incident magnetic field at a point $\boldsymbol{r}$ on the surface boundary S, where $\hat{\boldsymbol{n}}$ is the outward surface unit normal vector at $\boldsymbol{r}$. $S_{illum} \subset S$ is the surface which is illuminated directly by the source [28].

Chapter 3

The Multilevel Fast Multipole Algorithm Solutions

With its low complexity and error-controllable nature, MLFMA is the most qualified algorithm for solutions of electromagnetics problems up to date [15],[16],[1]. In order to solve large-scale problems, MLFMA is parallelized on distributed-memory architectures using the hierarchical partitioning strategy [20],[18],[19],[35]. For the rest of the thesis, the parallel implementation of MLFMA is considered. Real-life problems involves hundreds of millions of unknowns to be solved and the computational requirements for solutions grow immensely in terms of memory and CPU time. Therefore a chapter of methods are developed in order to reduce the memory requirements and solve problems faster.

The first method is the out-of-core implementation of parallel MLFMA. The out-of-core method is well known in computer science [36],[37]. The main idea of the out-of-core method is storing large data-structures on disk and using them from disk. In electromagnetics, out-of-core implementation has been applied on well-known solvers based on MoM [25],[38]. A parallel version of MoM using the out-of-core technique is implemented in [39]. A parallel out-of-core implementation of MLFMA is provided in [25],[26],[24]. This thesis provides further

developments on the parallel implementation out-of-core MLFMA [40],[27]. The implementation achieves solutions of extremely-large scattering problems involving up to 1.3 billion unknowns with 64 process and 2 TB memory. The largest solution up to date has 3 billion unknowns with 4096 processes and 25 TB memory [22].

The second method is the MPI+OpenMP parallelization scheme, which provides efficient parallelization with high number of processes [41],[42]. In electromagnetics, MPI+OpenMP parallelization scheme is applied on MLFMA in [43],[44],[45] and its success is proven for solutions of large-scale problems in [21] and [22]. The thesis provides efficient employment of MPI+OpenMP for hierarchical parallelization of MLFMA. Different from other approaches, the thesis uses the pure-MPI parallelization as a basis, i.e., in both inter-node and intra-node communications, and employs MPI+OpenMP when needed in order to prevent using extra memory. The implementation details are provided in Section 3.4 and 3.5.

MLFMA involves pre-processing, setup, and iterative-solution stages. Pre-processing reads the input geometry and constructs the multilevel tree-structure via clustering the basis (and testing) functions hierarchically. The setup calculates the translation operators among the clusters in each level, the far-field patterns of the basis (and testing) functions, and the near-field interaction matrix. Finally, the iterative solution is performed through MVMs.

3.1 The Out-of-Core MLFMA

In order to clarify the memory bottlenecks for solving large-scale problems, the memory requirements of MLFMA solutions are investigated. For this purpose, the allocated memory during MLFMA executions are recorded at various checkpoints in order to keep memory history of solutions. The checkpoints are chosen in a way that they represent the peak memory, i.e., the allocated memory does not exceed the recorded memory at any instant of the execution. Each process keeps its own memory and update a batch file at the checkpoints. As an example, a scattering problem involving a conducting sphere with the radius of 120λ is considered, where λ is the wavelength of the illuminating planewave. The problem has 53 million unknowns to be solved and, according to the formula in Appendix 1.1, the average mesh size is 0.1086λ. Fig. 3.1 shows the per-process memory, i.e., memory allocated by a single process, of the solution with 128 processes.

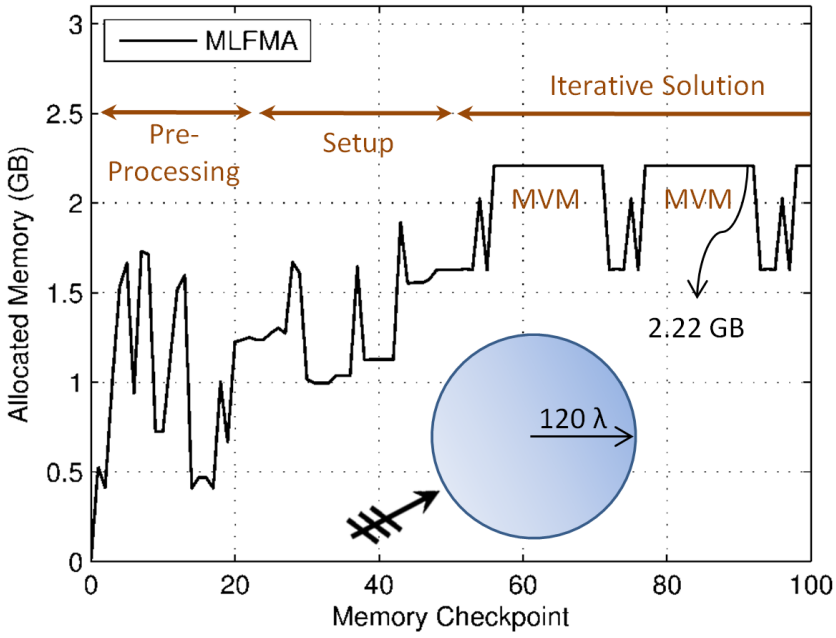

Figure 3.1: Memory history of the solution of a scattering problem.

The pre-processing performs between checkpoints 0 and 22, then the setup continues until the checkpoint 50, finally the iterative solution performs. Note that each memory peak in the iterative solution corresponds to an MVM. The maximum memory is observed at MVMs and therefore only the first 100 checkpoints are considered in the figure for simplicity. The total memory of the solution, i.e., sum of the memories of all processes, is 289 GB and the figure shows that the peak per-process memory is 2.22 GB.

The data structures used in the iterative solution are shown in Fig. 3.2 with their allocated memories. The first six data structures are static, i.e., calculated once and stored, and the latters are updated in each MVM. Requiring $\mathcal{O}(N)$ memory, the near-field interaction matrix and far-field patterns of the basis functions constitutes 61% of the peak memory and they are used out of core. The out-of-core data is stored in disk, hence, a 61% memory saving is achieved.

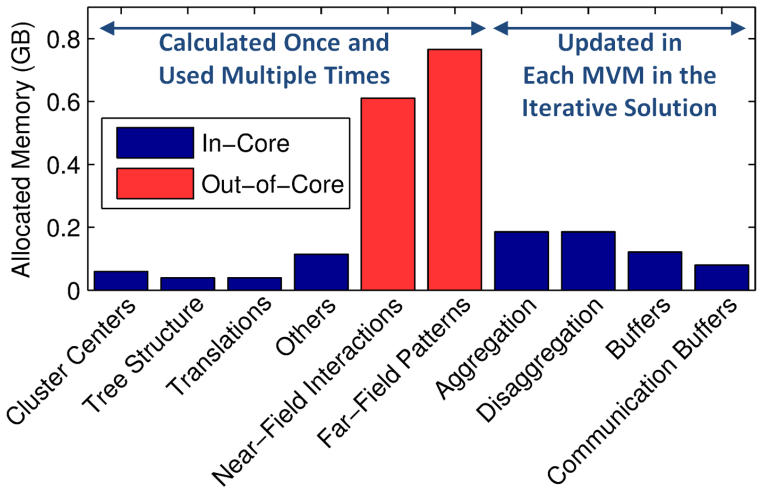

Figure 3.2: The data structures and their memories used in the iterative solution.

Other data than the out-of-core data, i.e. the in-core data, is kept in memory. The aggregation and disaggregation data structures holds the samples of aggregated and translated fields, respectively, of the clusters in every level of the MLFMA tree-structure.

Figure 3.3 shows the implementation scheme of out-of-core MLFMA. The red line divides the memory and disk spaces, referred as RAM and Disk, and the horizontal axis is the CPU time. At a certain point in the setup, the near-field interactions are written on disk, then they are read from disk in each MVM in the iterative solution. Similarly, the far-field patterns are written on disk and they are read from disk in each MVM in the iterative solution. MVMs involve near-field, i.e. $\overline{\boldsymbol{Z}}_{NF} \cdot \boldsymbol{x}_i$, and far-field, i.e., $\overline{\boldsymbol{Z}}_{FF} \cdot \boldsymbol{x}_i$, multiplications, where $\boldsymbol{x}_i$ is the coefficient vector comes from the iterative solver in the i^{th} MVM. In the out-of-core implementation, the near-field multiplication reads the near-field data once and the far-field multiplication reads the far-field patterns twice (one for aggregation and one for disaggregation) in each MVM. The black arrows in the figure represents the read and write (I/O) operations.

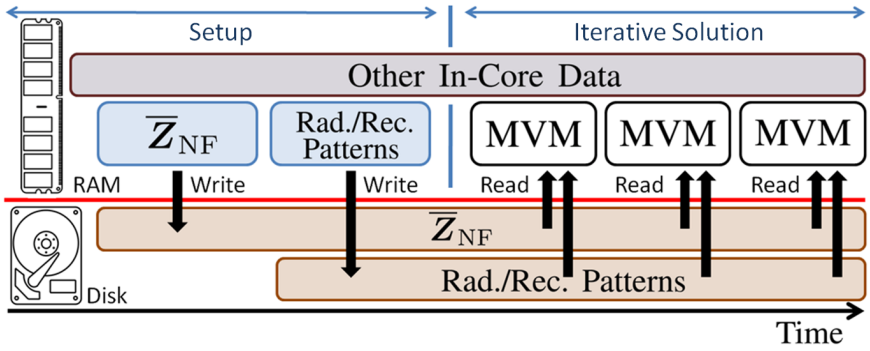

Figure 3.3: Out-of-core implementation scheme of MLFMA.

The I/O operations are performed on-the-fly, i.e., no memory is allocated for storing the whole portion of an out-of-core data structure. Instead, the out-of-core data is written and read in packets through small buffers. The out-of-core data is stored in binary form in order to achieve fast I/O and save disk space.

Figure 3.4 shows the memory history of the solution, described above, with the out-of-core implementation without pre-processing parallelization, i.e. OoC-MLFMA-sp. The red dashed-line shows that the out-of-core implementation reduces the MVM memory from 2.22 GB down to 0.86 GB and saves 1.36 GB of memory. However, the reduced memory in the iterative solution reveals another

memory bottleneck in the pre-processing and the setup. For example, the memories at the checkpoints 8 and 28 are 1.73 GB and 1.67 GB, respectively, and hence, the peak memory is 1.73 GB and the out-of-core implementation is not useful with OoC-MLFMA-sp.

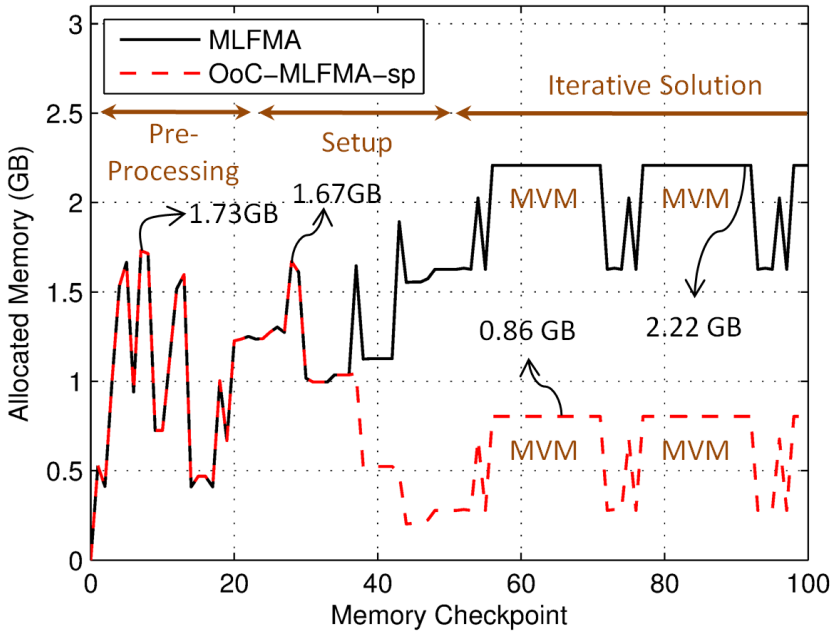

Figure 3.4: Memory history of the solution of a scattering problem with the out-of-core implementation.

The memory in the pre-processing and setup stages are constituted mainly by large data-structures of the input geometry, i.e., the geometry to be solved. MLFMA deallocates those data structures in order to save memory once some necessary structures for the solution stage are constructed, however, the pre-processing must be performed. In pre-processing, each process needs to access the whole geometry data at certain steps and therefore, in a sequential implementation, each process keeps its own copy of data structures. For overcoming the bottleneck the data is partitioned among processes via a data-parallelization strategy for pre-processing as described in Section 3.2.

3.2 Pre-Processing Parallelization

The input-geometry data is read from disk in parts and distributed among processes in portions. One of the processes, e.g., the master process, is employed for reading the data in parts sending the parts to other processes. The memory for keeping the whole portion of data is never allocated in any of the processes, however, each process needs all the data at certain points in the pre-processing, therefore a series of communications are performed.

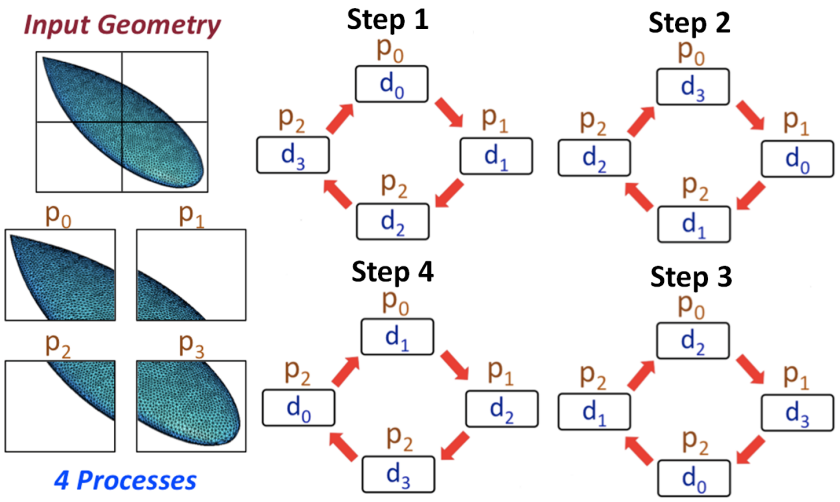

Figure 3.5: Parallelization scheme of MLFMA pre-processing.

Fig. 3.5 shows the partition of the input geometry and communications among processes. There are four processes in the figure and each of them is denoted by p_i where $i \in \{1, 2, 3, 4\}$. The input-geometry data is partitioned equally among processes and each p_i has its initial portion of data d_i. At certain points, a four-step communication is performed for processes to access any portion d_i. In each step, a processes passes its data to another, eventually, each process retrieve its initial portion at the end of the fourth step. Meanwhile a processes can access any portion of data d_i.

The sequential pre-processing takes $\mathcal{O}(N)$ time. The parallelization scheme does not achieve task parallelization, but data-parallelization since each process performs $\mathcal{O}(N/p)$ operations p times, where p is the number of processes. Therefore the complexity of the p-step communication is $\mathcal{O}(N)$ and the communication overhead is $\mathcal{O}(p)$. Fig. 3.6 shows the CPU times of pre-processing for the solution, described in Section 3.1, with various numbers of processes. The figure shows the pre-processing time increases linearly with the number of processes because of the communication overhead.

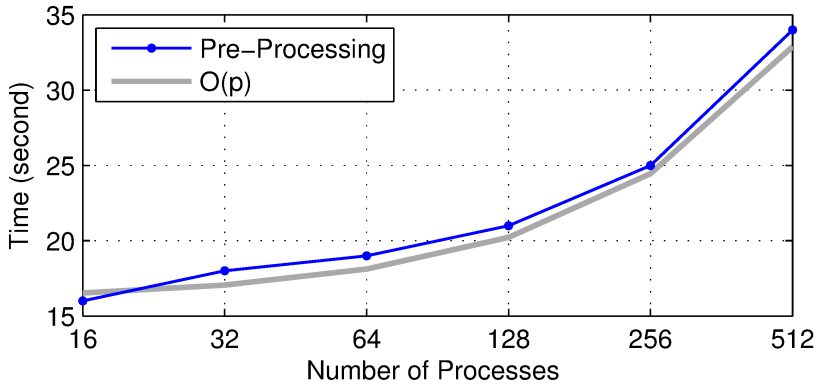

Figure 3.6: CPU times of pre-processing with various numbers of processes.

With the pre-processing parallelization the data structures related with the input geometry is partitioned, i.e., the data is not duplicated, among processes and memory reduction is achieved. Fig. 3.7 shows that the memory of the pre-processing and setup is decreased and the peak memory is constituted by MVMs in the iterative solution as desired. The OoC-MLFMA is the out-of-core implementation of MLFMA with the pre-processing. With the OoC-MLFMA, the peak memory is decreased from 2.22 GB down to 0.86 GB.

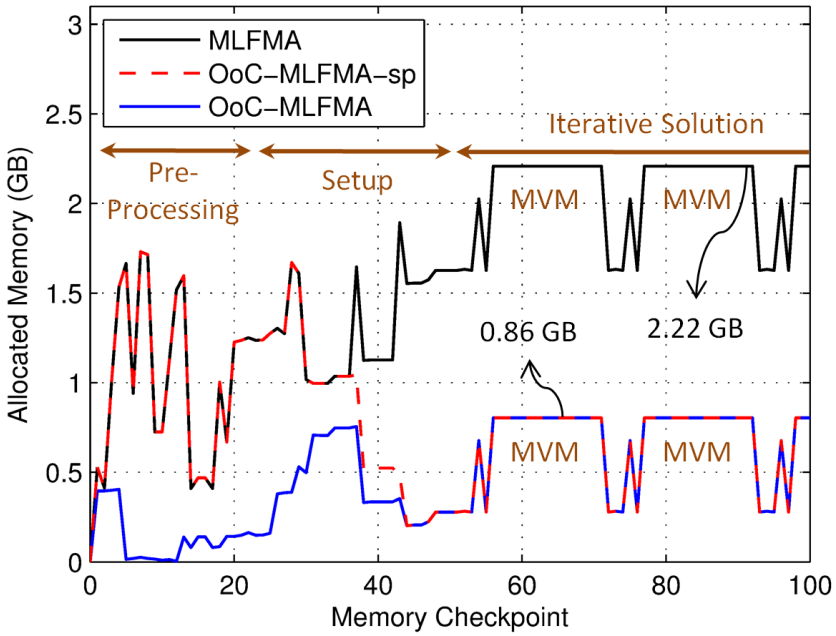

Figure 3.7: Memory history of the solution with the out-of-core implementation and the pre-processing parallelization.

3.3 Parallel implementation of OoC-MLFMA

In the parallel implementation of the out-of-core method, each process reads (and writes) the out-of-core data on disk simultaneously [40]. Each process opens a file with a distinct name, i.e., with their MPI rank, and writes their own out-of-core data packets on the file. When a single node has multiple processes, the the disk driver on the node shares the I/O jobs of those processes. An implementation example involving four-node computer cluster and 16 processes is shown in Fig. 3.8. Assuming there is a single disk drive in each node, I/O jobs of four processes are handled by a single disk drive. A process is shown with P_i, where $i \in \{0, 1, 2, ..., 15\}$ is the process ID and a file is shown with F_i, and each process owns a distinct file, in other words, P_i owns F_i.

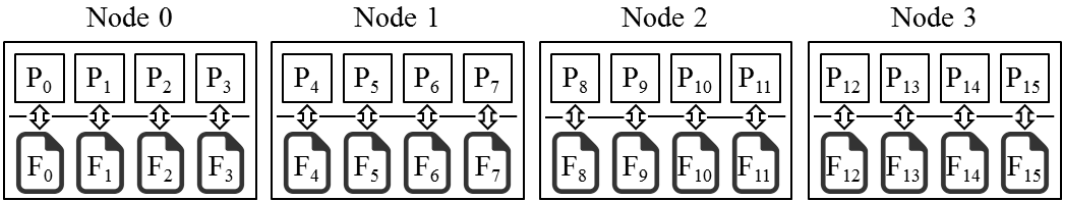

Figure 3.8: The parallel and out-of-core MLFMA with 16 process is employed on a four-node computer cluster.

For a demonstration of the proposed method, a set of scattering problems involving various numbers of unknowns is solved with MLFMA and OoC-MLFMA. The problems involve conducting spheres whose radius vary from 120λ to 500λ. Specifically, a sphere with the radius of 0.3 m is meshed with $\lambda/10$ mesh size at various frequencies. Table 3.1 shows the problem specifications. The first and the second columns show the sphere radius and number of unknowns, respectively. The third, the fourth, and the sixth columns show number of processes, CPU times, and required memories for MLFMA solutions, whereas the fifth, the sixth, and the seventh columns show number of processes, CPU times, and required memories for OoC-MLFMA solutions, respectively. The last column shows the size of out-of-core data of the OoC-MLFMA solutions. The CPU times are given in hours and the memory and disk requirements are given in GB.

The problems are solved in a 16-node high-performance computing cluster with total memory of 2 TB (See Appendix A.2). Each node has 16 processing cores and is equipped with 400 GB SSD for out-of-core storage. The problems are formulated with CFIE formulation with the combination factor of 0.5. A Bi-CGSTAB solver [8] is employed with block-diagonal preconditioner [9] in order iterative solutions to satisfy 1% residual error norm.

Table 3.1: The Problem set Involving Various Numbers of Unknowns

Rad.	**Unk.**	**MLFMA**			**OoC-MLFMA**			
(λ)	**(mil.)**	**Proc.**	**Time**	**Mem.**	**Proc.**	**Time**	**Mem.**	**Disk**
120	53	128	1.3	289	128	1.3	108	185
137.1	69	128	1.8	443	128	1.8	207	241
160	93	128	2.7	563	128	2.7	244	325
180	135	128	3.5	685	128	3.6	426	263
210	167	128	4.7	812	128	4.7	503	313
230	212	128	5.6	968	128	5.7	566	405
260	277	128	7.4	1249	128	7.9	666	533
280	374	128	10.1	1642	128	11.3	779	771
340	540	64	24.1	1906	128	17.8	1123	1102
380	670	-	-	-	128	26.6	1814	1361
440	850	-	-	-	128	38.6	2156	1681
500	1109	-	-	-	64	71.3	1915	2103

The smallest problem has 53 million unknowns and the largest problem that OoC-MLFMA can solve has 1.1 billion unknowns. Without employing the out-of-core method, MLFMA can solve at most 540 million unknowns within 2 TB memory. Increasing number of processes also increases memory slightly, therefore the largest problems are solved with 64 processes in order to fit the solutions into the system within the memory limit (see Section 3.4).

Figure 3.9 shows the required memories for the solutions. The memory limit is shown with a brown line. The blue line shows the MLFMA memory without the employment of the out-of-core method. The largest MLFMA solution requires 1.82 TB memory. The black line shows the OoC-MLFMA memory. The figure shows that OoC-MLFMA reduces the required memory by 38%-64%, depending on the problem. The green dashed-line shows the sum of disk and memory usage of OoC-MLFMA. The sum can be interpreted as a projection of the MLFMA memory and, hypothetically, MLFMA would need 3.93 TB for solving 1.1 billion unknowns. The available memory of the system is depicted with the brown line. The 540-million-unknown MLFMA solution would require 2.12 TB memory (which is not available) with 128 processes.

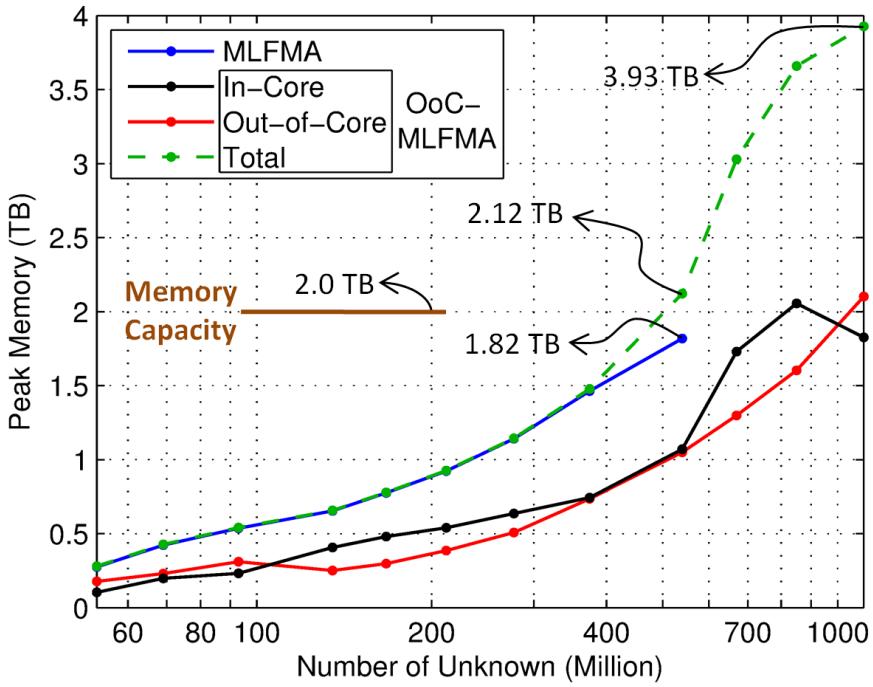

Figure 3.9: Required memory (and disk) spaces for solutions in the problem set in Table 3.1.

Employing the out-of-core method increases the CPU time slightly because of the slow disk I/O operations. The implementation is optimised for fast SSDs for reducing the slowdown, however, it cannot be prevented completely [24]. Consider the solution involving the sphere with the radius of 280λ in the problem set in Table 3.1. The problem involves 374,490,624 unknowns and is solved with MLFMA and OoC-MLFMA using 128 processes. Fig. 3.10 shows the memory history of the solutions. The blue and the black lines show the memory of the MLFMA and the OoC-MLFMA solutions, respectively. The peak memory of the MLFMA solution is 1.46 TB, whereas the peak memory of the OoC-MLFMA solution is 0.74 TB, hence, a 49% memory reduction is achieved. The red dashed-line shows the allocated disk space for storing the out-of-core data. The near-field interactions, i.e., $\overline{\boldsymbol{Z}}_{\boldsymbol{NF}}$, has 46,961,186,432 elements and requires 350 GB memory (eight bytes for each element) to be stored. The whole memory is never allocated and the data it is written out in packets in packets at checkpoint 37. Similarly, the far-field patterns of the basis (and testing) functions are is written

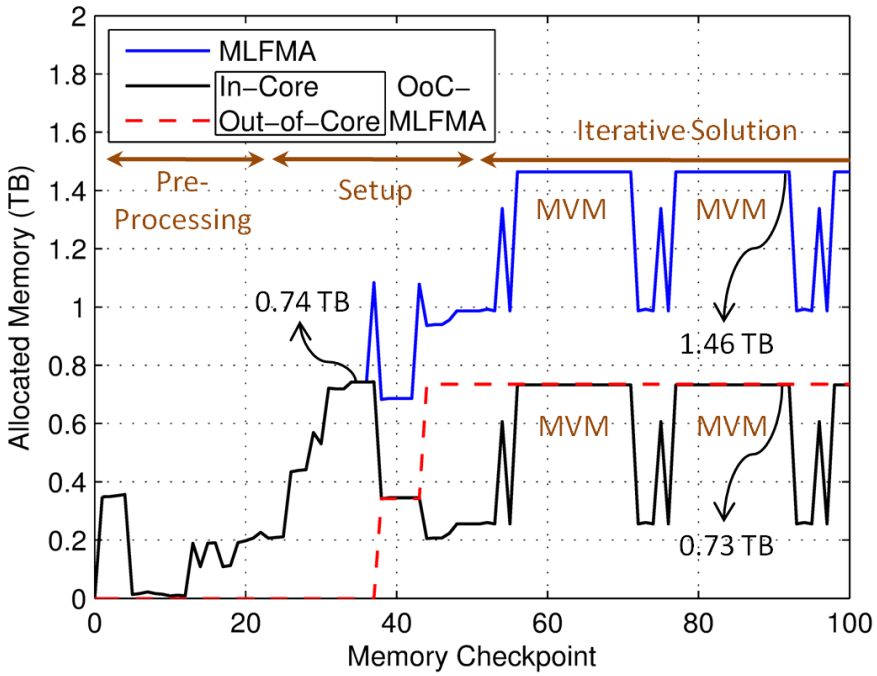

Figure 3.10: Memory history of the solution involving 374 million unknowns with 128 processes.

out at checkpoint 44. Requiring 402 GB, each basis (and testing) function holds 36 samples for electric-field and magnetic-field patterns and each sample has two components. Therefore a total amount of 752 GB is stored out of core, i.e., in disk. The solution of the problem requires 29 iterations to converge and each Bi-CGSTAB iteration requires two MVMs [8], and therefore 58 MVMs are performed. Hence, the out-of-core data is read 58 times in the iterative solution. MLFMA and OoC-MLFMA solves the problem in 10.06 and 11.25 hours, hence, a 12% slowdown is experienced.

Fig. 3.11 shows the CPU times of the solutions in the problem set. The Blue line shows the MLFMA and the red line shows the OoC-MLFMA solutions. The grey line in the figure shows that the solution times are in good agreement with $\mathcal{O}(NLogN)$ curve. The CPU times of the 540-million-MLFMA solution and the 1.1-billion-OoC-MLFMA solution are out of trend since the solutions are performed with 64 processes, while others are performed with 128 processes.

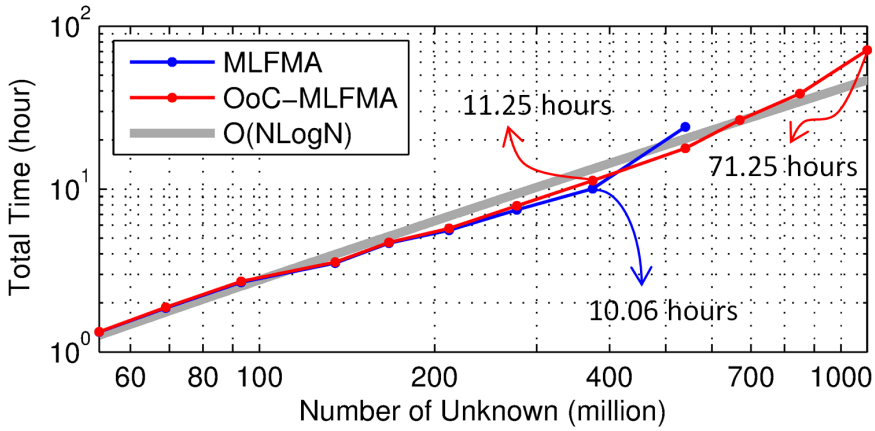

Figure 3.11: CPU times of solutions in the problem set in Table 3.1.

The largest sphere has a diameter of 1000λ and OoC-MLFMA solves the 1,109,280,768 unknowns in 71.25 hours. The detailed CPU times of the largest solution is shown in Table 3.2 in minutes. The first column shows the pre-processing time, the latter four columns show detailed setup time, the sixth column shows the iterative solution time, and the last column shows the total time. The solution performs 37 Bi-CGSTAB iterations, and hence 74 MVMs are performed, where each MVM takes approximately 35 minutes. The measurements show that reading the out-of-core data takes 9 minutes in each MVM.

Table 3.2: CPU Times for Solving 1.1 Billion Unknowns

Pre-Proc.	**Setup**				**Iterative Solution**	**Total**
	Trans.	**Nearfield**	**Fourier**	**Total**		
28	301	1178	10	1652	2596	4275

Figure 3.12 shows the radar-cross section (RCS) of the sphere with 1000λ diameter [40]. The backscattering and forward-scattering angles are 0° and 180° angles, respectively. The figure shows that the OoC-MLFMA results agree well with the analytical Mie-series solution. Specifically, the computational RCS errors are 3.75%, 2.80%, and 0.94% in the 0°–30°, 0°–90°, 0°–180° angle sectors, respectively.

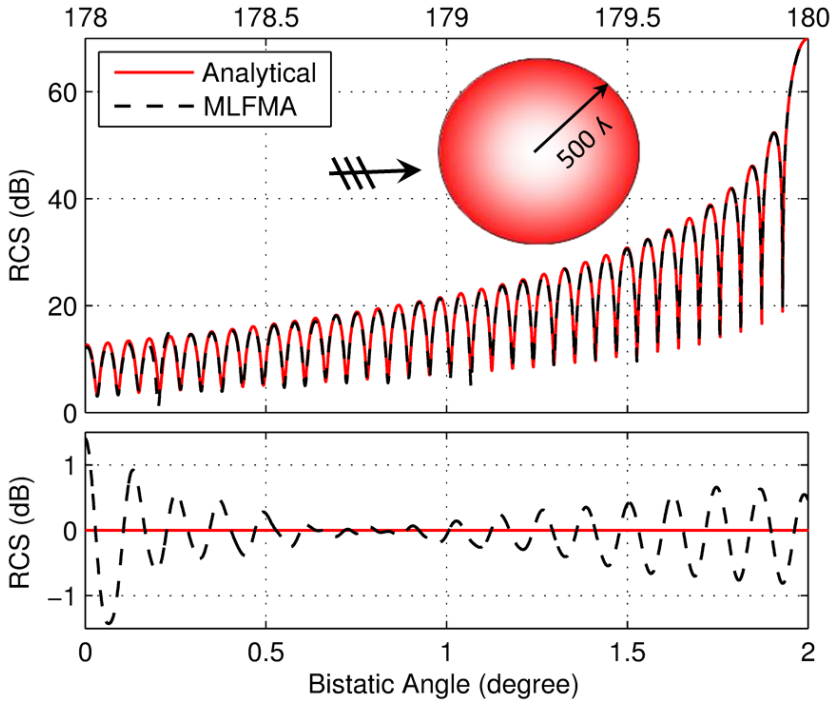

Figure 3.12: RCS of the sphere with 1000λ diameter.

In addition to the solutions in the problem set in Table 3.1, a conducting NASA Almond with a length of 2104λ and a conducting Flamme [46] with a length of 2401λ are solved with 64 processes [27]. Both geometries are illuminated 30° from their sharp ends on the azimuth plane with horizontal polarization. 30° and 210° angles are the backscattering and the forward-scattering angles, respectively. The NASA Almond problem has 1,126,503,936 and the Flamme problem has 1,338,909,696 unknowns and, as in the sphere solutions, Bi-CGSTAB solver and block-diagonal preconditioner are employed for the iterative solutions in order to satisfy 1% residual error norm.

Figure 3.3 shows the RCSs of the targets in dBms. The black and the red lines show the co-polar and cross-polar RCSs, respectively. Backscattering and forward-scattering angle-sectors are zoomed in the figure for the reader's convenience. Table 3.3 shows the detailed CPU times of the solutions in minutes. OoC-MLFMA solves the NASA Almond and flamme problems in 61 and 75 hours, respectively, with 64 processes.

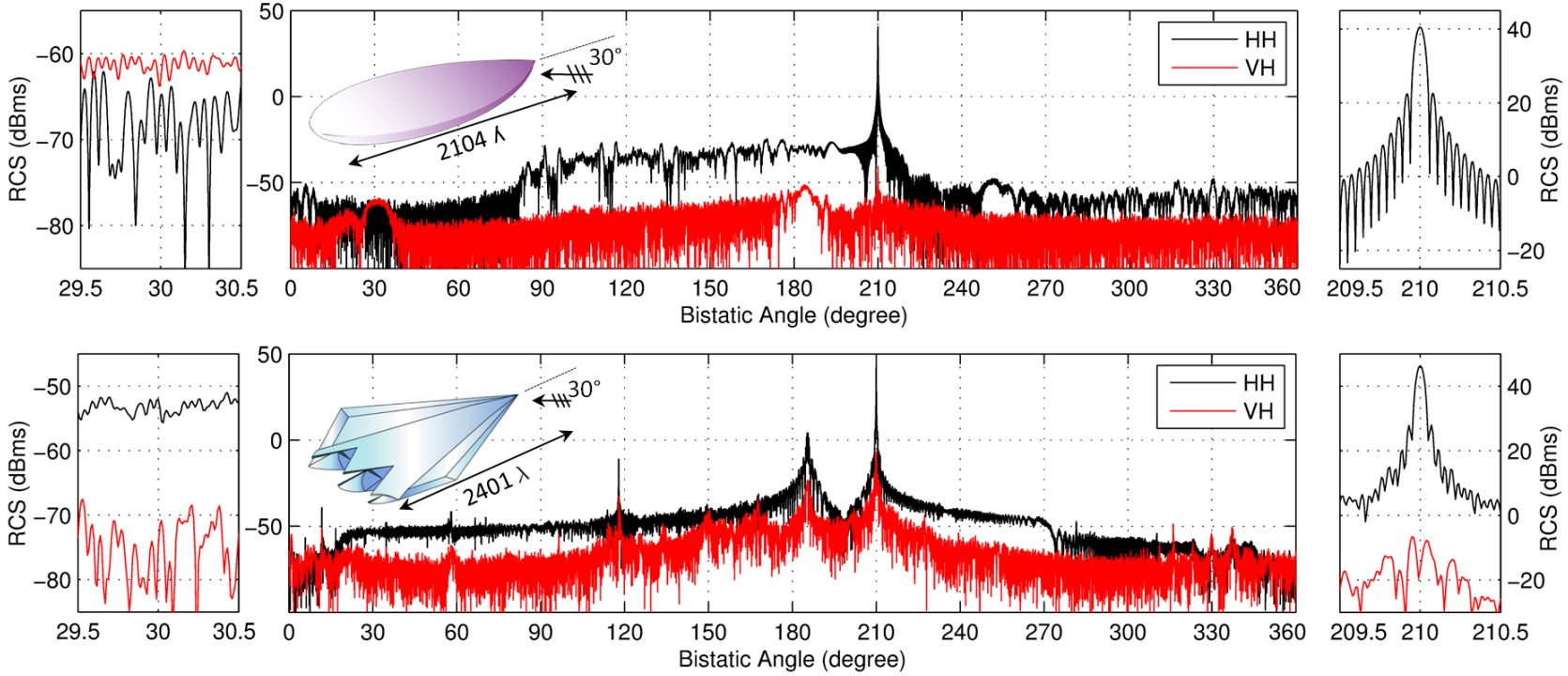

Figure 3.13: Bistatic RCS of the NASA Almond and Flamme.

Table 3.3: CPU Times of the NASA Almond and Flamme Solutions

	Pre-Proc.	Setup				Iterative Solution	Total
		Trans.	Nearf.	Fourier	Total		
N. Alm.	24	205	1163	11	1388	2232	3645
Flamme	31	272	1411	13	1735	2755	4520

Solutions of billions of unknowns takes vast amount of time with 64 processes and it is utterly desirable to solve the problem with higher numbers of processes. The system has 512 processing cores (see Appendix A.2) but the available memory does not permit increasing the number of processes as total memory increases with the number of processes. As a remedy, an MPI+OpenMP parallelization scheme is developed in the rest of the chapter, which can employ all processing cores in the system without any memory increase.

3.4 MPI+OpenMP Parallelization

Hierarchical partitioning strategy of MLFMA uses MPI parallelization [1],[20], which uses distributed-memory scheme, i.e., MPI processes cannot access each other's memory, and they send the data via communications even they use the same physical memory space. MPI parallelization can use multiple computers interconnected with high-speed network, but the communication overhead is its drawback because sending data from one process to another takes time. In order not to make intense communications, some small data structures are duplicated among processes, i.e., each process has a copy of the same data. Therefore the total memory of those data structures grows with the number of processes and prevents employing high number of processes in a limited memory.

Consider the problem with 53 million unknown in the problem set 3.1. The problem is solved with various numbers of processes and the solution memories are shown with black lines in Fig. 3.14. The solid lines represent the MLFMA solutions and the dashed lines represent the OoC-MLFMA solutions. Requiring less memory than that of MLFMA's, OoC-MLFMA memory increases with the number of processes and constitutes a limitation on the number of processes. For

example, hypothetically, OoC-MLFMA could not solve the problem with higher number of processes than 64 in a system which has 100 GB total memory because it would require 118 GB with 128 processes, 167 GB for 256 processes and so on. 512 processes requires almost triple memory of 64 processes'.

The memory efficiency with p processes is defined as $E_p^M = M_{16}/pM_p$, where M_p is the total memory with p processes. For example, if data could be partitioned perfectly among processes, i.e., without duplication, the efficiency would be 100%. The blue lines show that the efficiency of OoC-MLFMA drops faster with increasing number of processes. The main reason is that the well-parallelized data structures, specifically, the near-field interactions and far-field patterns, are used out of core and duplicated data constitute high portion in the remaining in-core memory.

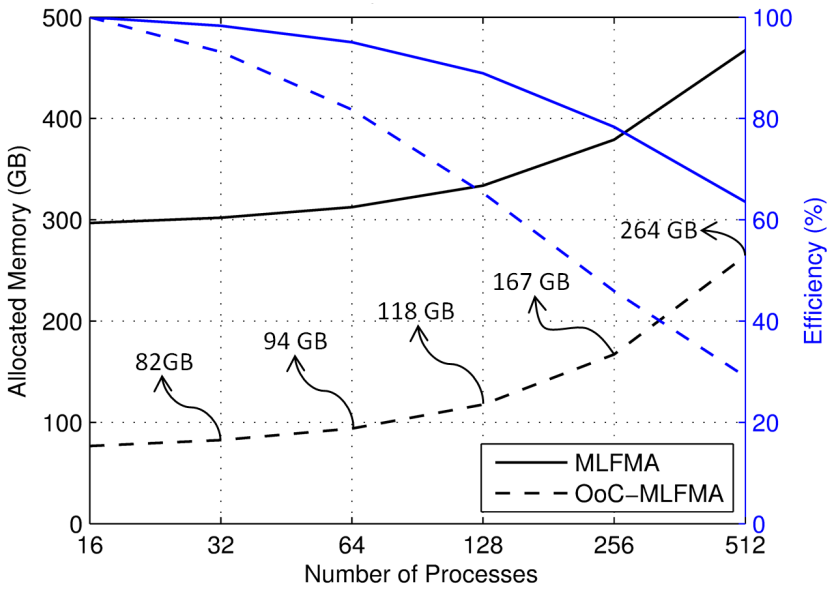

Figure 3.14: MLFMA Memory Increases with the Number of Processes.

As a contrast to MPI parallelization, OpenMP parallelization uses shared-memory scheme, where OpenMP threads can adress the same physical memory space, therefore they can share a data structure without any communication or duplication. One drawback of OpenMP parallelization is that it cannot cannot

be used in distributed-memory architectures and another is that the efficiency may drop because of the race conditions when a memory address is updated by multiple OpenMP threads.

For taking the advantages of MPI and OpenMP parallelization schemes a hybrid MPI+OpenMP parallelization scheme is developed for MLFMA [21],[22]. In the scheme, each MPI process forks off a number of OpenMP threads and the threads divides the task of an MPI process within the same memory space. Fig. 3.15 depicts the MPI+OpenMP parallelization scheme. There are two MPI processes and each process summons two OpenMP threads in order to partition its task. The summoned threads are employed on idle processing cores by the operating system, hence, more processing cores are be employed without requiring extra memory. An OpenMP thread can share a data structure with its sibling, however, it cannot address a memory of its cousin thread. Therefore communications must be handled by MPI processes and threads must be terminated when a communication is to be initiated.

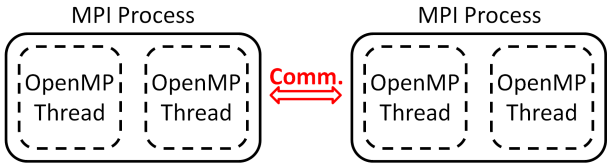

Figure 3.15: OpenMP threads can share MPI Processes' memory while the processes must make communications.

MLFMA is divided into MPI and MPI+OpenMP regions. The regions which requires communications, MPI regions, are handled by MPI and the regions with no communication, MPI+OpenMP regions, are parallelized with OpenMP. Fig. 3.16 depicts the MPI and MPI+OpenMP regions. Requiring the same memory, MPI regions employ p processing cores and MPI+OpenMP regions employ $p \times t$ processing cores, where t is the number of threads per process.

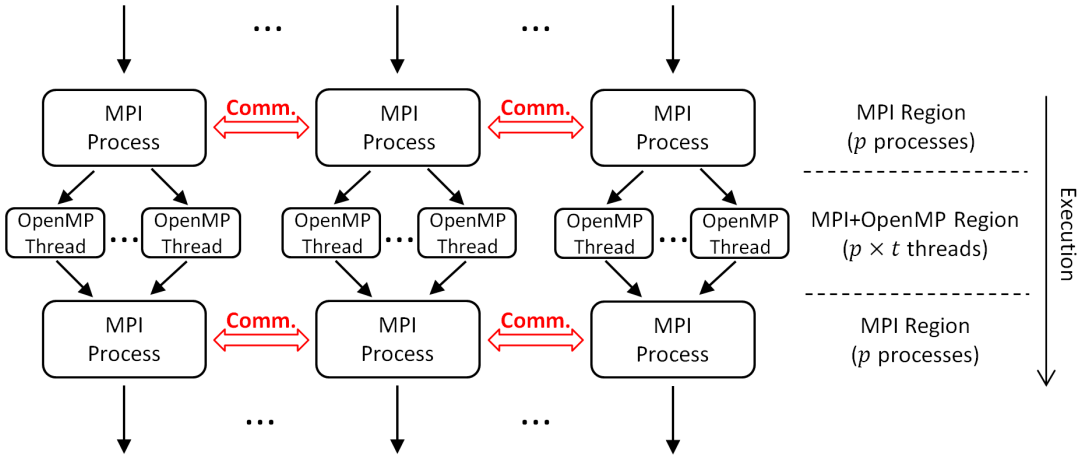

Figure 3.16: Each MPI process forks off a number of OpenMP threads in MPI+OpenMP regions.

3.5 MPI+OpenMP Implementation of OoC-MLFMA

MLFMA setup constitutes approximately 38% of the CPU times of the large-scale solutions, as seen in Tables 3.2 and 3.3. Requiring no communication, the setup stage is parallelized with MPI+OpenMP scheme. Fig. 3.17 shows the MPI+OpenMP regions of the algorithm. Calculation of translation operators, near-field interaction matrix, far-field patterns of the basis functions, and pre-conditioner matrix are parallelized with MPI+OpenMP scheme.

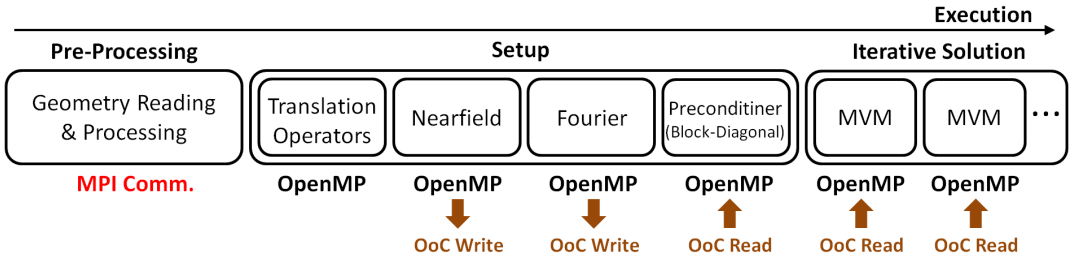

Figure 3.17: Setup and certain parts of the iterative solution are parallelized with MPI+OpenMP scheme.

Additional to the setup stage, certain parts of MVMs, which does not require communication, are parallelized with MPI+OpenMP scheme. Fig. 3.18 shows the MPI+OpenMP regions in an MVM. The near-field multiplication, lowest-level aggregations and disaggregations, and intra-process translations does not require communication and parallelized with MPI+OpenMP. The MPI communications are depicted in figures in order to highlight the regions that cannot be parallelized with MPI+OpenMP scheme. The brown arrows in the figures show the out-of-core employment and the direction of the I/O operations, i.e., downward arrows depict writing on disk and upward arrows depict reading from disk.

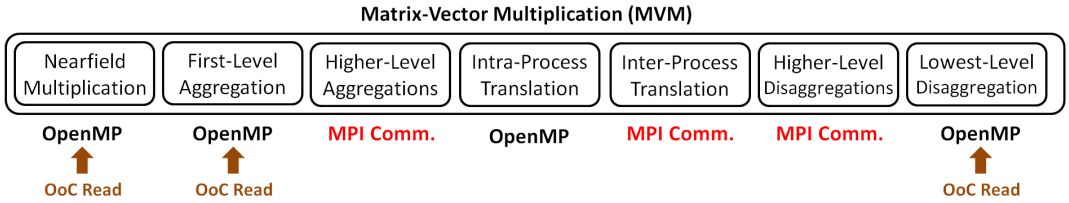

Figure 3.18: MPI communications prevent MPI+OpenMP parallelization at certain parts of MVMs.

The out-of-core method writes the near-field interaction matrix, which is used in preconditioner in setup and near-field multiplication in MVMs, and the far-field patterns of basis functions, which are used in lowest-level aggregation and disaggregation. In the MPI+OpenMP regions, OpenMP threads are employed for reading and writing the out-of-core data. Fig. 3.19 shows the employment of 8x2 threads for I/O operations. There are eight processes and each process employs two threads, yielding total number of 16 threads, where each threads names its own out-of-core file with a unique name, i.e., t_i names its file with f_i. Specifically, each thread names its file with its thread ID and belonging MPI rank.

As a demonstration, the 53-million-unknown-problem in Table 3.1 is solved with MPI+OpenMP implementation. Table 3.4 shows the detailed CPU times and memories of the solutions. The CPU times are given in second and the memories are given in GB. The first row shows the solver type, i.e., MLFMA or OoC-MLFMA, the second row shows the parallelization scheme, the third row shows the number of MPI processes, and the fourth row shows the total number

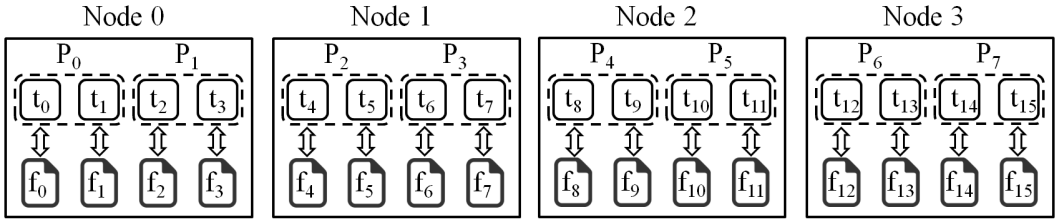

Figure 3.19: OpenMP threads are employed for reading and writing the out-of-core data.

of threads for MPI+OpenMP parallelization. In MPI parallelization, no thread is employed. The last row shows that the total memory increases with the number of process, but it remains the same when OpenMP threads are employed. The total CPU times show that MPI+OpenMP parallelization decreases the CPU time by 80% in setup and 53% in total by employing 512 threads, using the same memory with 64 processes. The table also shows that MPI+OpenMP parallelization beats the MPI parallelization not only in terms of memory, but also in terms of CPU time, i.e. 16x8 threads are faster than 512 processes. This is mainly due to MPI communications which decreases the parallelization efficency with high number of processes.

Additionally, the sphere with 1.1 billion unknowns, the NASA Almond with 1.1 billion unknowns, and the flamme with 1.3 billion unknowns are solved with MPI+OpenMP implementation. The implementation employs 64 MPI processes in the MPI regions and 512 OpenMP threads in the MPI+OpenMP regions. Table 3.5 shows detailed CPU times of the solutions in minutes. MPI parallelization can use at most 64 processes but MPI+OpenMP can use 512 threads with the same memory. The table shows that the MPI+OpenMP parallelization is very effective in setup stage of the algorithm, decreasing the setup times by 81% for all of the solutions. The parallelization decreases the iterative solution times by 11%-13% and total times by 38%-39%. The employment saves 23 hours, for the sphere and NASA Almond solutions and 28 hours for the flamme solution without requiring extra memory.

Table 3.4: CPU Times and Memories for Solving 53 Million Unknowns with MPI and MPI+OpenMP Schemes

			MLFMA	**OoC-MLFMA**							
Parallelization Scheme			**MPI**					**MPI+OpenMP**			
Number of MPI Processes			**64**	**64**	**128**	**256**	**512**	**64**			
Number of OpenMP Threads			N/A	N/A	N/A	N/A	N/A	64x1	64x2	64x4	64x8
CPU Time (sec)	**Pre-Processing**		161	92	120	181	322	87	89	86	93
	Setup	**Translation**	233	234	124	134	148	241	132	75	50
		Nearfield	4699	4680	2451	1377	955	4517	1910	1763	1010
		Fourier	85	76	37	23	11	79	48	26	13
		Total	5715	5664	2900	1735	1566	5468	3173	1866	1141
	Iterative Solution		4294	5713	3930	2989	3838	5006	4581	4390	4176
	Total I/O		N/A	122	69	28	59	131	72	36	24
	Total		10170	11469	6949	4904	5726	10560	7842	6342	5410
Peak Memory (GB)			313	93	115	163	283	93			

Table 3.5: CPU Times of the Sphere, NASA Almond, and Flamme Solutions

Parallelization Scheme			**MPI**	**MPI+OpenMP**
Number of MPI Processes			**64**	
Number of OpenMP Threads			**N/A**	**64x8**
Sphere	**Pre-Processing**		28	40
	Setup	**Translation**	301	56
		Nearfield	1178	230
		Fourier	10	2
		Total	1652	325
	Iterative Solution		2596	2258
	Total		4275	2623
NASA Almond	**Pre-Processing**		24	35
	Setup	**Translation**	205	38
		Nearfield	1163	196
		Fourier	11	1
		Total	1388	266
	Iterative Solution		2232	1976
	Total		3645	2277
Flamme	**Pre-Processing**		31	43
	Setup	**Translation**	272	51
		Nearfield	1411	255
		Fourier	13	2
		Total	1735	327
	Iterative Solution		2755	2453
	Total		4520	2823

Chapter 4

Physical Optics Solutions

For physical optics (PO) solutions of scattering problems involving arbitrary geometries, the surface current is discretized using RWG basis functions and the current coefficients are determined by solving a linear system of equations iteratively. For solving large-scale scattering problems, the PO algorithm is parallelized via distributing the data and computation among processes in a load-balanced manner. In this way, each process calculates a distinct portion of the linear system and performs the MVMs in the iterative solution simultaneously.

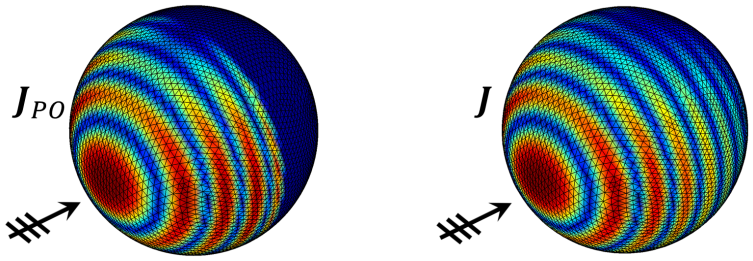

Figure 4.1: PO current (left) and exact current (right) on a sphere illuminated by a planewave.

Figure 4.1 shows approximate PO and exact currents on a sphere illuminated by a planewave. Exact currents are achieved with MLFMA. The figure shows that PO approximation omits the surface current on the dark side of the sphere.

PO has been implemented on non-uniform meshes in [47],[48], and [49], however, it is not efficient when uniform meshes are involved and its efficient applications are limited. Moreover, the solution's computational requirements increases immensely when large geometries are involved, i.e. billions of triangle patches are involved. As a remedy, a parallel PO solver is developed in this thesis, which uses uniform RWG basis functions. For determining the basis coefficients, a sparse linear system with $\mathcal{O}(N)$ non-zero elements is solved in linear time.

4.1 Discretization of Surface Current

The PO current can be discretized using RWG basis functions as

$$\sum_{n=1}^{N} x_n^{PO} \boldsymbol{b}_n(\boldsymbol{r}) \approx \boldsymbol{J}^{PO}(\boldsymbol{r}), \tag{4.1}$$

where $\boldsymbol{J}^{PO}(\boldsymbol{r})$ is the PO current, N is the number of basis functions, and $\boldsymbol{b}_n(\boldsymbol{r})$ and x_n^{PO} are the n^{th} basis function and its coefficient, respectively.

To find the unknown coefficient vector $\boldsymbol{x}^{PO}$, (4.1) is tested with N testing functions as

$$\int_{S_t} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \sum_{n=1}^{N} x_n^{PO} \boldsymbol{b}_n(\boldsymbol{r}) = \int_{S_t} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{J}^{PO}(\boldsymbol{r}), \tag{4.2}$$

where $\boldsymbol{t}_m(\boldsymbol{r})$ and S_t are the m^{th} testing function and its domain, respectively. The summation can be taken outside as

$$\sum_{n=1}^{N} x_n^{PO} \int_{S_t} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{b}_n(\boldsymbol{r}) = \int_{S_t} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{J}^{PO}(\boldsymbol{r}), \tag{4.3}$$

and the equation can be rewritten as

$$\sum_{n=1}^{N} Z_{mn}^{PO} x_n^{PO} = v_m \tag{4.4}$$

or equivalently as

$$\overline{\boldsymbol{Z}}^{PO} \cdot \boldsymbol{x}^{PO} = \boldsymbol{v}^{PO}, \tag{4.5}$$

where

$$Z_{mn} = \int_{S_t} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{b}_n(\boldsymbol{r}) \tag{4.6}$$

is the interaction of the m^{th} testing and the n^{th} basis functions and

$$v_m = \int_{S_t} d\boldsymbol{r}\, \boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{J}^{PO}(\boldsymbol{r}) \tag{4.7}$$

is the testing of the PO current on the m^{th} testing function. One can substitute (2.73) into (4.7) for calculating v_m. Solution of (4.5) for $\boldsymbol{x}^{PO}$ gives the desired current coefficients in order to perform post processing.

When the m^{th} testing and the n^{th} basis function domains are not overlapped, Z_{mn} vanishes, and therefore the interaction matrix $\overline{\boldsymbol{Z}}^{PO}$ in (4.5) is extremely sparse. Specifically, with RWG basis and testing functions, there are at most $5N$ non-zero elements (exactly $5N$ non-zero elements in closed geometries) in $\overline{\boldsymbol{Z}}^{PO}$. Fig. 4.1 depicts a testing function overlapping with five basis functions on a discretized NASA Almond geometry. As the interaction matrix $\overline{\boldsymbol{Z}}^{PO}$ has $\mathcal{O}(N)$ non-zero elements, the vanishing elements are never stored and the system can be solved iteratively in linear time.

4.2 Parallelization of the Physical Optics Solver

For parallelization of the proposed PO algorithm, the basis (and testing) functions are sorted with respect to their positions in the three-dimensional Cartesian space, then clustered using hypothetical boxes with pre-determined dimensions. The basis clusters are distributed among processes in a load-balanced manner. Each box corresponds to a matrix block in the system (4.5) and each process calculates a matrix partition. The sorting, clustering, and distribution of basis functions are performed in parallel using the pre-processing parallelization proposed in Section 3.2. Fig. 4.2 shows the basis functions on a NASA Almond and hypothetical boxes used for clustering the basis functions.

For the demonstration of the proposed parallel PO algorithm, large-scale scattering problems involving a conducting sphere and a conducting NASA Almond

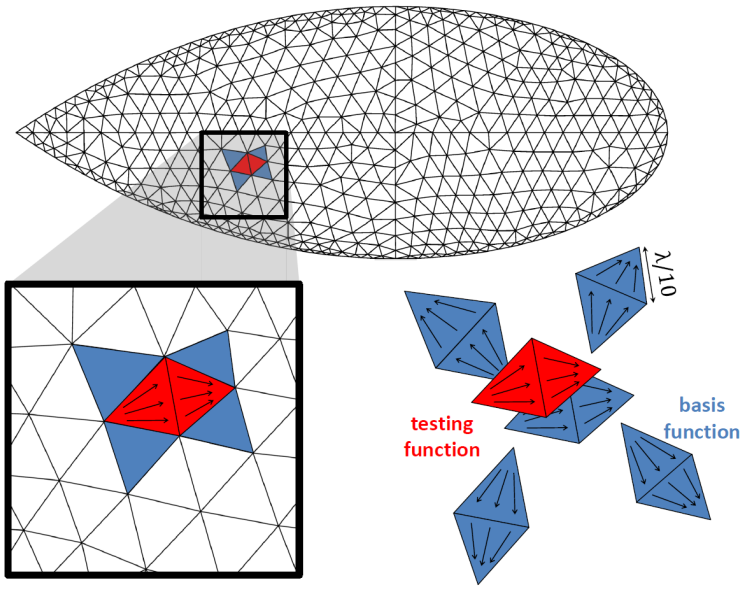

Figure 4.2: A descretized NASA Almond geometry with RWG testing and basis functions. A testing function (shown with red color) overlaps five basis functions (shown with blue color).

geometries are solved. The sphere has a radius of 500λ and the NASA Almond has a length of 2104λ. The geometries are discretized with 0.1λ mesh size and the cluster size is chosen as 0.25λ The solutions are performed using 64 processes on a 16-node high performance computing system with 2 TB of total memory.

The RCS of the sphere, shown in Fig. 4.4, shows that the PO result is in a good agreement with the analytical Mie series result in forward-scattering (180°) and backscattering (0°) angles. But the PO result is approximate in other bistatic-angle sectors, as expected. The system in (4.5) has 1,109,290,768 unknowns and the solution requires 11 CGS iterations in order to satisfy 10^{-6} residual error norm [7]. Table 4.1 shows the CPU times of the sphere solutions in seconds. The first column shows the pre-processing time, the second column shows the setup time, and the third column shows the iterative-solution time. The setup involves filling of the sparse matrix $\overline{\boldsymbol{Z}}^{PO}$ and the right-hand-side $\boldsymbol{v}^{PO}$.

The bistatic RCS the NASA Almond is shown in Fig. 4.5. As a reference,

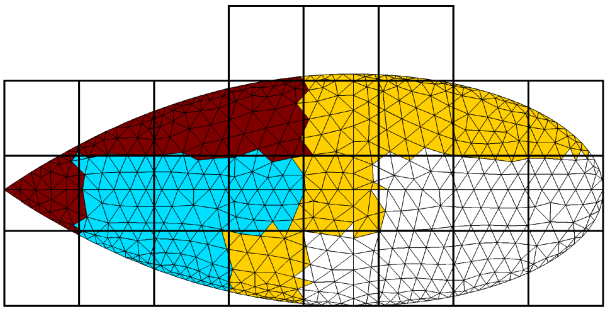

Figure 4.3: A descretized NASA Almond geometry with RWG testing and basis functions. A testing function overlaps with at most five basis functions.

Table 4.1: CPU times of the Sphere Solution with PO Solver

Pre-Processing	**Setup**	**Solution**	**Total**
1658	27	1692	3393

the same problem is solved with MLFMA. Figure shows that the PO result is approximate in general, but it is in a good agreement with the MLFMA result around forward-scattering (210°) and specular-reflection (120°−180°) angles. The system in (4.5) has 1,126,503,936 unknowns and the solution requires 13 CGS iterations in order to satisfy 10^{-6} residual error norm. MLFMA solves the same problems in 30 Bi-CGSTAB iterations along with block-diagonal preconditioner.

Table 4.2 shows the CPU times and memory requirements of the NASA Almond solutions with MLFMA and PO with 64 processes. The CPU times are given in seconds and the memory requirements are given in TB. Table shows that the MLFMA solution takes approximately 60 hours while the PO solution takes less than one hour. The pre-processing times of the solutions are similar since they use the same pre-processing algorithm described in Section 3.2. The setup of MLFMA involves the filling of the near-field matrix $\overline{\boldsymbol{Z}}_{NF}$, calculation the translations and far-field patterns. $\overline{\boldsymbol{Z}}_{NF}$ has approximately 137 billion elements and their calculation takes vast amount of time, i.e., 23 hours. As opposed to the large computations of MLFMA, the PO system $\overline{\boldsymbol{Z}}^{PO}$ has approximately 5.6 billion non-zero elements and it is filled in 27 seconds. As a result, the parallel PO solver solves the NASA Almond 71 times faster than the parallel MLFMA solver.

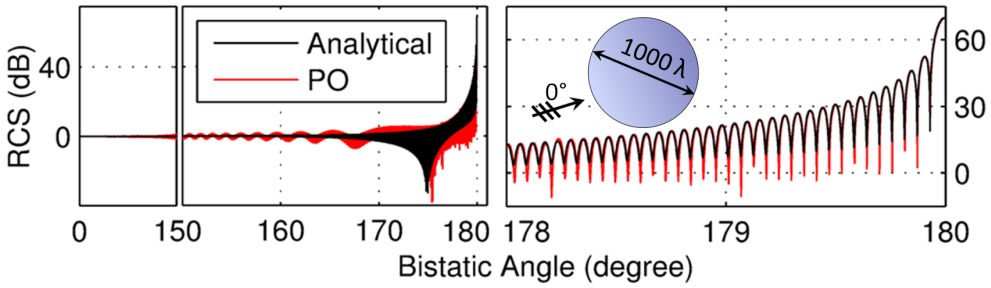

Figure 4.4: RCS of the sphere. The analytical Mie series and the PO solutions are plotted with the black and the red lines, respectively.

For the solutions, PO requires 1.9 TB while MLFMA requires 4.1 TB. MLFMA requires more memory than PO in order to keep the near-field interactions and far-field patterns.

Table 4.2: CPU times and Memory Requirements of NASA Almond Solutions

	Pre-Processing	**Setup**	**Solution**	**Total**	**Memory**
PO	1388	27	1647	3072	1.9
MLFMA	1457	83307	133947	218712	4.1

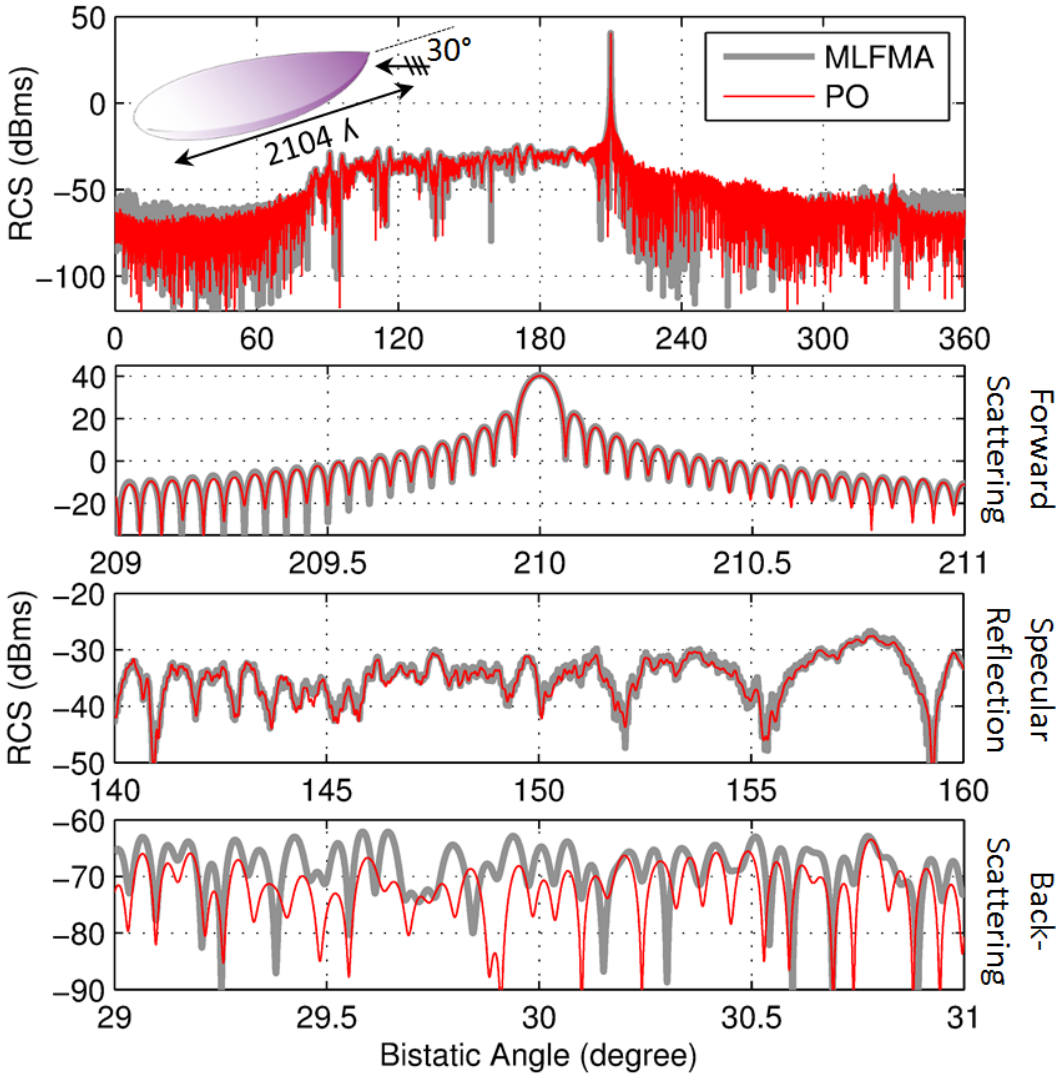

Figure 4.5: Bistatic RCS of the NASA Almond. The PO solution is plotted with the red line, whereas MLFMA solution is plotted with the grey line as the reference.

Chapter 5

Hybrid Integral-Equations and Physical-Optics Solutions

For combining the accuracy of integral-equation (IE) and the rapidity of PO solutions, a hybrid IE and PO solution methodology is introduced. The method divides the problem into IE and PO domains and solves the surface currents with the corresponding formulation, as depicted in Fig. 5.1. Typically, large and basic surfaces are solved with PO and complex surfaces are solved with IE. The solutions cannot be thought independent of each other because of the interaction among domains. The PO solution takes $\mathcal{O}(N)$ time, whereas the IE solution takes $\mathcal{O}(NLogN)$ time, therefore it is desirable to solve more unknowns with PO.

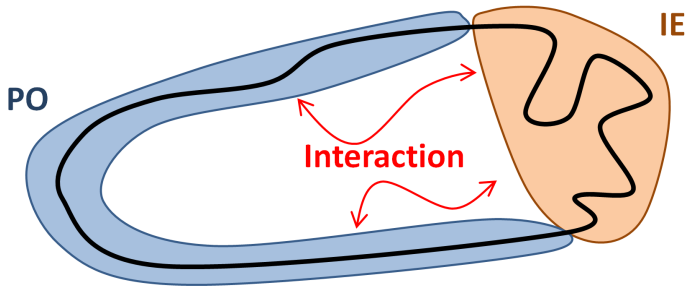

Figure 5.1: The problem domain is divided into non-overlapping PO and IE domains and each domain is formulized by corresponding method.

MoM-based hybrid PO-IE method is formulized in [50]. Various approaches for Hybrid PO-MoM MLFMA technique are presented in [51],[52],[53]. Employment for hybrid PO-IE method is first provided in [54] and [55], then in [56]. This thesis provides a general formulation for MoM-based hybrid PO-IE technique and demonstrates an efficient employment of MLFMA with solutions of scattering and radiation problems [29].

5.1 Formulation

A general MoM system of equations $\overline{\boldsymbol{Z}} \cdot \boldsymbol{x} = \boldsymbol{v}$ can be partitioned as

$$\left[\begin{array}{c|c} \overline{\boldsymbol{Z}}^{IE} & \overline{\boldsymbol{Z}}^{IP} \\ \hline \overline{\boldsymbol{Z}}^{PI} & \overline{\boldsymbol{Z}}^{PO} \end{array}\right] \cdot \left[\begin{array}{c} \boldsymbol{x}^{IE} \\ \hline \boldsymbol{x}^{PO} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{v}^{IE} \\ \hline \boldsymbol{v}^{PO} \end{array}\right], \tag{5.1}$$

where $\boldsymbol{x}^{IE}$ and $\boldsymbol{x}^{PO}$ are the coefficients of the currents in the IE and PO domains, respectively. The system in (5.1) can be written as two separate systems as

$$\overline{\boldsymbol{Z}}^{IE} \cdot \boldsymbol{x}^{IE} + \overline{\boldsymbol{Z}}^{IP} \cdot \boldsymbol{x}^{PO} = \boldsymbol{v}^{IE} \tag{5.2}$$

and

$$\overline{\boldsymbol{Z}}^{PI} \cdot \boldsymbol{x}^{IE} + \overline{\boldsymbol{Z}}^{PO} \cdot \boldsymbol{x}^{PO} = \boldsymbol{v}^{PO}. \tag{5.3}$$

The equations can be interpreted as the following; $\overline{\boldsymbol{Z}}^{IE}$ is the interaction of basis and testing functions in the IE domain, i.e., it tests the incoming field on the IE domain originated by the current on the IE domain. Similarly, $\overline{\boldsymbol{Z}}^{IP}$ tests the incoming field on the IE domain originated by the current on the PO domain and $\overline{\boldsymbol{Z}}^{PI}$ tests the incoming field on the PO domain originated by the current on the IE domain. Therefore $\overline{\boldsymbol{Z}}^{IE}$ and $\overline{\boldsymbol{Z}}^{PO}$ have $N^{IE} \times N^{IE}$ and $N^{PO} \times N^{PO}$ elements, respectively, where N^{IE} and N^{PO} are the number of unknowns in the IE and PO domains, respectively. Similarly, $\overline{\boldsymbol{Z}}^{IP}$ and $\overline{\boldsymbol{Z}}^{PI}$ have $N^{IE} \times N^{PO}$ and $N^{PO} \times N^{IE}$ elements, respectively, i.e. $N = N^{IE} + N^{PO}$. For the hybrid IE and PO solutions, $\overline{\boldsymbol{Z}}^{PO}$ and $\boldsymbol{v}^{PO}$ is formulated as in (4.6) and (4.7), respectively, where $\boldsymbol{x}^{PO}$ gives the PO current coefficients. When RWG basis and testing functions are used, $\overline{\boldsymbol{Z}}^{PO}$ is extremely sparse with $\mathcal{O}(N^{PO})$ elements. In order to enhance the hybrid solutions, N^{PO} is kept large.

The three matrices other than $\overline{\boldsymbol{Z}}^{PO}$ are full matrices. To be able to solve large systems, only the near-field elements are calculated and stored, and the far-field elements are multiplied on-the-fly via aggregation, translation, and disaggregation of the fields in a multilevel scheme. In this regard, the MVMs are performed similar to the MVMs in MLFMA.

5.2 Solutions of Scattering Problems

The selection of IE and PO regions is a matter of choice, however, it is desirable to solve more unknowns with PO. Consider a disk-like geometry in Fig. 5.2. The geometry with the radius of 10λ is lied on the x-y plane and is illuminated with a ϕ-polarized planewave coming from $(\theta, \phi) = (45°, 0°)$ [54]. The full-IE and the full-PO solutions are given in Fig. 5.3. The error of the PO solution is found by subtracting the PO current from the IE current, i.e., $|\boldsymbol{J}^{IE} - \boldsymbol{J}^{PO}|$, which is shown in Fig. 5.4. The figure shows that the error is mainly around the shadow boundary, i.e. the boundary which divides the illuminated and shadowed regions. The error is mainly due to the discontinuity of the PO current. Therefore, to provide the smooth current transition from the illuminated region and the shadowed region, the hybrid solution chooses the region around shadow boundary is as the IE region and the rest of the surface as the PO region.

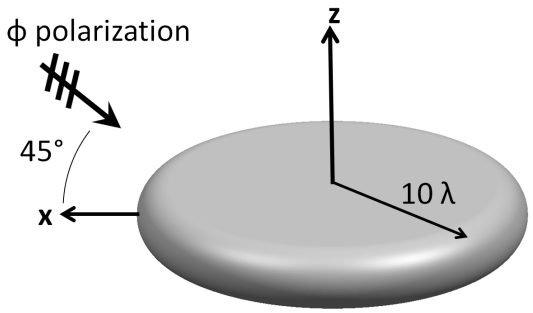

Figure 5.2: A disk-like geometry is illuminated with planewave.

Figure 5.5 shows the IE and PO regions on the disk. The problem involving

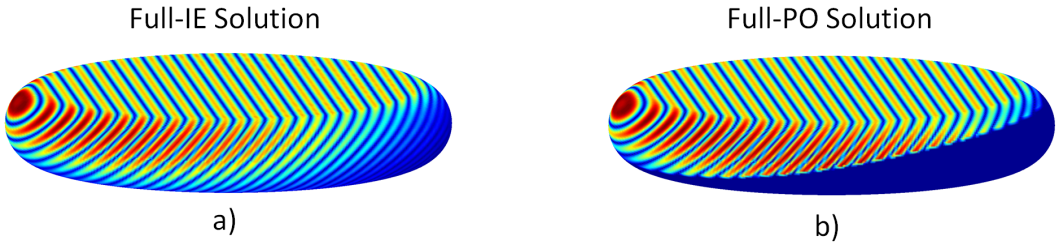

Figure 5.3: a) Surface current of the full-IE solution. b) Surface current of the full-PO solution.

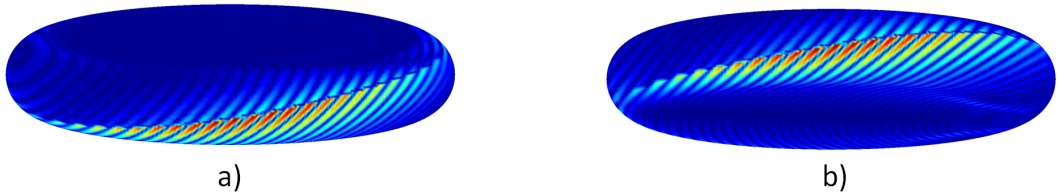

Figure 5.4: The error of the PO solution is found by subtracting the PO current from the IE current.

the disk geometry is discretized with 261,786 unknowns and the partitioning of the surface into the IE and PO regions yields 68,374 unknowns in the IE region and 193,412 unknowns in the PO region.

Figure 5.5: The region around the shadow boundary is chosen as the IE domain and the rest is chosen as the PO domain.

The PO current is chosen as the starting point to solve the equations (5.2) and (5.3). As an approximation, the algorithm omits the effect of the IE current $\boldsymbol{x}^{IE}$ on the PO domain, in other words, it ignores the coupling between the IE and PO currents. Therefore the elements of $\overline{\boldsymbol{Z}}^{PI}$ in (5.3) is selected as $Z_{mn}^{PI} = 0$ and the PO current is determined by solving $\overline{\boldsymbol{Z}}^{PO} \cdot \boldsymbol{x}^{PO} = \boldsymbol{v}^{PO}$. Then the IE current

is solved in the presence if the PO current by substituting $\boldsymbol{x}^{PO}$ into Eq. (5.2) and solving $\boldsymbol{x}^{IE}$ in

$$\overline{\boldsymbol{Z}}^{IE} \cdot \boldsymbol{x}^{IE} = \boldsymbol{v}^{IE} - \overline{\boldsymbol{Z}}^{IP} \cdot \boldsymbol{x}^{PO} \tag{5.4}$$

iteratively. $\overline{\boldsymbol{Z}}^{IP}$,$\boldsymbol{v}^{IE}$, and $\overline{\boldsymbol{Z}}^{IP}$ is formulated with EFIE and the MVMs $\overline{\boldsymbol{Z}}^{IE} \cdot \boldsymbol{x}^{IE}$ and $\overline{\boldsymbol{Z}}^{IP} \cdot \boldsymbol{x}^{PO}$ are performed similar to those in MLFMA. Note that $\overline{\boldsymbol{Z}}^{IP}$ is a rectangular matrix with $N^{PO} \times N^{IE}$ elements.

Figure 5.6 depicts the described algorithm. Analytically, the problem domain S is divided into into non-overlapping S^{IE} and S^{PO} domains. The PO current $\boldsymbol{J}^{PO}$ on S^{PO} is found using the PO formula in Eq. (2.73). Then the IE current on S^{IE} is found by solving EFIE in Eq. (2.50) in the presence of the incident field form the PO current. The scattered field can be found using the Eq. (2.31) by integrating the currents on both domains, i.e., $S = S^{IE} \cup S^{PO}$.

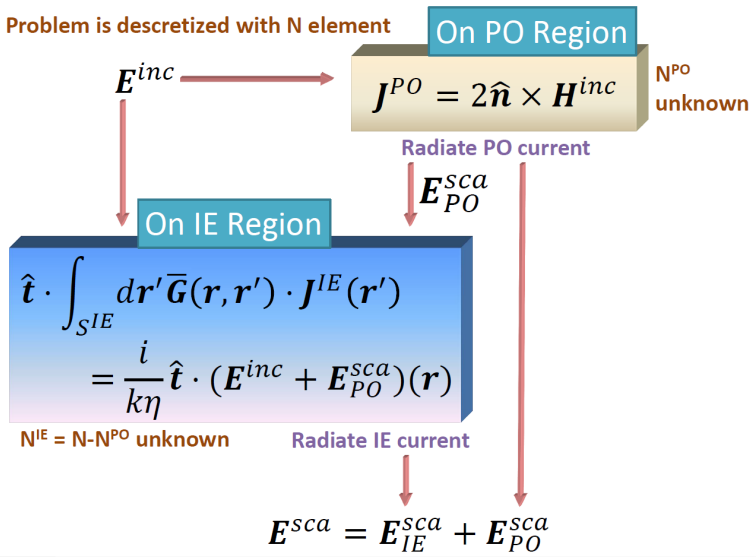

Figure 5.6: The hybrid IE and PO algorithm for solving scattering problems. The currents on IE and PO regions are found seperately using the PO equation and EFIE, respectively.

The problem involving the disk geometry is solved using the hybrid IE and PO formulation. RCS of the disk is showed in Fig. 5.7. 45° is the backscattering and 135° is the forward-scattering angles, respectively. Figure shows that the

hybrid solution is in a good agreement with the IE (reference) solution, whereas the Full-PO solution is approximate.

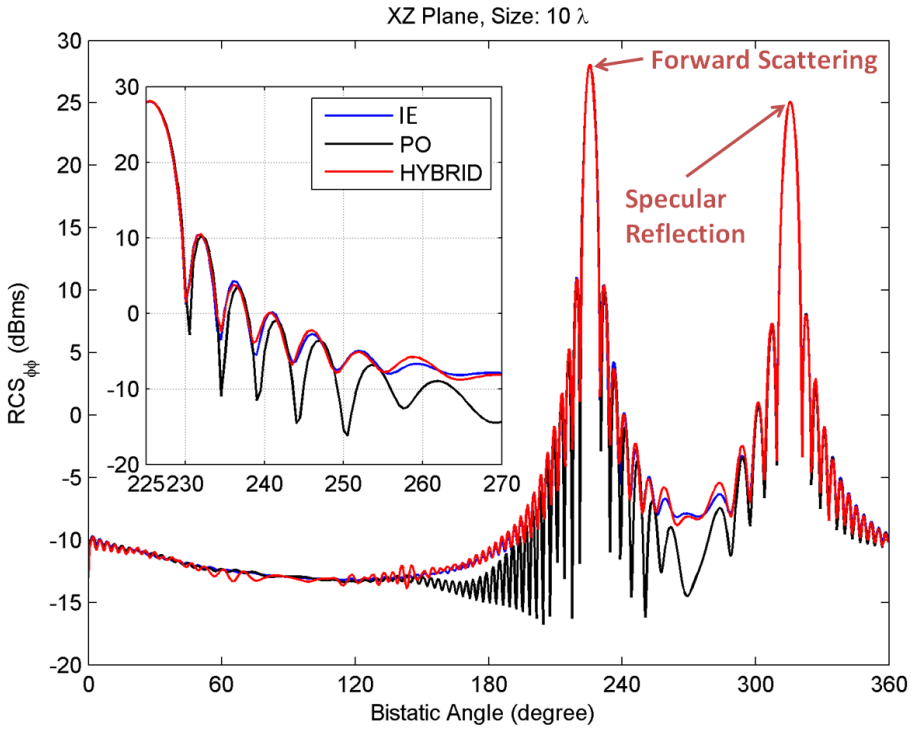

Figure 5.7: Bistatic co-polar RCS of the disk geometry. The IE, PO, and hybrid solutions are shown with the blue, black, and red lines, respectively.

Figure 5.8 shows the surface current error of the hybrid solution. Comparing with the Full-PO error in Fig. 5.4, lower error values are observed for the hybrid IE and PO current. Table 5.1 shows the required CPU times of the solutions. The first and the second rows show the CPU times of the pre-processing and the setup times, and the iterative solution times, respectively. The pure PO solution is quite fast in comparison with the pure IE (MLFMA) solution. The setup of the MLFMA solution involves calculations of the near-field interactions $\overline{\boldsymbol{Z}}_{NF}$ and the setup of the hybrid solution involves the near-field interactions between the PO and IE domains, i.e., $\overline{\boldsymbol{Z}}_{NF}^{IP}$, and the among the IE domain, i.e. $\overline{\boldsymbol{Z}}_{NF}^{IE}$. The number of non-zero elements in $\overline{\boldsymbol{Z}}_{NF}^{IE}$ and $\overline{\boldsymbol{Z}}_{NF}^{IP}$ are quite less than that of $\overline{\boldsymbol{Z}}_{NF}$ and therefore the setup in the hybrid solution is faster than that of MLFMA.

The last row shows the total CPU times of the solutions. The MLFMA solution and the hybrid solution require 26 and 39 CGS iterations, respectively, in order to satisfy 10^{-7} residual error norm, where per-iteration times of the MLFMA and the hybrid solutions are 0.49 and 0.30 minutes. The MLFMA solution is formulated using CFIE with the combination factor of 0.5, whereas the hybrid solution is formulated using pure EFIE since the IE domain is an open surface. As a result, the hybrid solution takes quite less time than the pure IE (MLFMA) solution as seen in the last column of the table.

Figure 5.8: The error of the hybrid PO and IE solution is found by subtracting the resulting current from the IE current.

Table 5.1: CPU Times of the Solutions in Minutes

	Pure IE	**Pure PO**	**Hybrid**
Pre-Proc. & Setup	10.6	0.0	1.4
Solution	12.8	0.1	11.8
Total	23.4	0.1	13.2

5.3 Solutions of Radiation Problems

Hybrid IE and PO formulation solves not only scattering, but also radiation and scattering problems together. As a demonstration of the proposed method, an Archimedian-type spiral antenna with the radius of 10.5 cm is mounted 10 cm over a planar disk with the radius of 40 cm, as seen in Fig. 5.9. The antenna and platform are concentric and the configuration is solved at 2 GHz and 18 GHz frequencies. For the 2-GHz solution, the antenna is meshed with a mesh size gradually decreasing from outer region to the feed direction, specifically, the feed region, the middle region, and the outer region are meshed with the mesh sizes 3

mm, 6 mm, and 9 mm, respectively, as seen in Fig. 5.10. For the 18-GHz solution, the antenna is meshed with $\lambda/10$ mesh size. The ground-planes are meshed with $\lambda/10$ mesh size at the corresponding frequency for both of the problems. The number of unknowns in the antenna and the ground-plane meshes is given in Table 5.2. The antenna and the platform has 24,943 and 7,751 unknowns in the 2-GHz solution and has 79,417 and 675,823 unknowns in the 18-GHz solution, respectively.

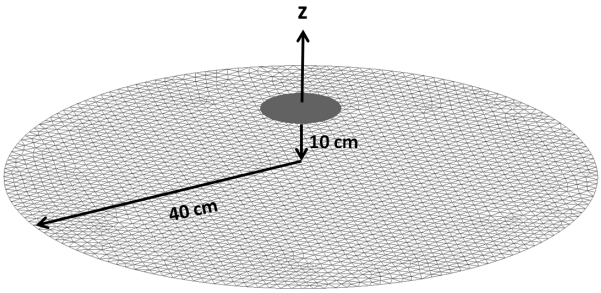

Figure 5.9: The spiral antenna is mounted over a large platform.

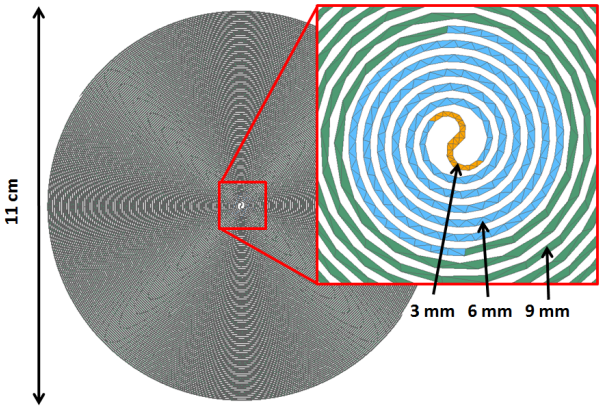

Figure 5.10: The spiral antenna is meshed with various mesh sizes, finer mesh in the feed region and coarser mesh in outer regions.

For each of both of the problems, two alternative solutions are provided. The first one chooses the entire platform as the PO domain, however, the second one adds the region inside the radius $r_{IE} = 20$ cm to the IE domain, whereas the

entire antenna is always in the IE domain. The IE domains are formulated with EFIE since the surfaces are open.

Table 5.2: Number of Unknowns in the Meshes

	2 GHz	**18 GHz**
Antenna	24,943	79,417
Platform	7,751	675,823

For solutions of the problems, the free-space solutions of the antennas, i.e. without platforms, are taken as a starting points. Let a free-space current coefficients of the antenna be $\boldsymbol{x}^{IE}_{init}$. The coefficients are substituted in (5.3) in order to find $\boldsymbol{x}^{PO}$. Note that $\overline{\boldsymbol{Z}}^{IP}$ is formulated with MFIE in order to find the incident magnetic field on the PO region. Then (5.2) is solved for $\boldsymbol{x}^{IE}$ by substituting known $\boldsymbol{x}^{PO}$ into the equation. The solution algorithm is summarized below:

1. Solve the antenna in free-space using IE. Let the solution be $\boldsymbol{x}^{IE}_{init}$
2. Substitute $\boldsymbol{x}^{IE}_{init}$ into eq. (5.3) and solve the equation for $\boldsymbol{x}^{PO}$
3. Substitute $\boldsymbol{x}^{PO}$ into eq. (5.2) and solve the equation for $\boldsymbol{x}^{IE}$

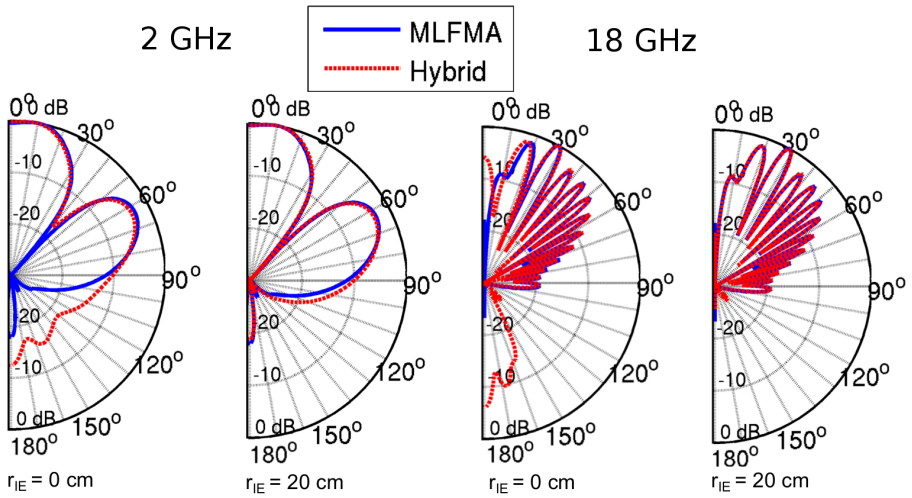

Figure 5.11: Radiation patterns of the antenna mounted on platform at 2 GHz and 18 GHz frequencies.

For the iterative solutions involving the antenna, i.e. in Steps 1 and 4, the LSQR solver is used along with the near-field preconditioner. For solving the $\boldsymbol{x}^{PO}$ in step 2, the CGS solver is used. Figure 5.11 shows the radiation patterns of the antenna mounted on platform at 2 GHz and 18 GHz frequencies. For all of the problems, full-IE (MLFMA) solutions are taken as reference. The figure shows the hybrid results are approximate when r_{IE} = cm, i.e. when the entire platform is chosen as PO region, whereas they are in a good agreement with the MLFMA solutions when $r_{IE} = 20$ cm, i.e. when the center of the platform is in the IE region.

Table 5.3 shows the CPU times and the required memories for the MLFMA and hybrid solutions. The IE (MLFMA) and the hybrid solutions requires the same memory amounts for the solutions, whereas the hybrid solutions take less time than IE solutions. The problem is solved in 1.3 hours with MLFMA and in 1.1 hours with the hybrid algorithm at 2 GHz. At 18 GHz, the problem is solved in 7.4 hours with MLFMA and 3.3 hours with the hybrid algorithm. Results show that the hybrid algorithm is more efficient in higher frequencies because a higher portion of the unknowns are in the PO region which are solved faster in $\mathcal{O}(N)$ time.

Table 5.3: CPU Times and Memory Requirements for the Solutions

	2 Ghz		**18 GHz**	
Solution Method	Full-IE	Hybrid	Full-IE	Hybrid
CPU Time (Hours)	1.3	1.1	7.4	3.3
Memory (GB)	5.1		20.8	

For feeding the antenna in a numerical environment, the right hand side elements in the eq. (5.2) is set as $v_n^{IE} = 0$ except the one in the excitation function. The n_{feed}^{th} function at the center of the antenna is set as $v_{n_{feed}}^{IE} = i/k\eta$.

Chapter 6

Conclusions

A chapter of methods for decreasing the computational resources are developed for MLFMA. The out-of-core implementation of parallel MLFMA employs disk space for storing large data structures and decreases required memory by approximately 50%, but extends the solution time by 10%. Decreasing the peak memory reveals another memory bottleneck in the pre-processing which prevents the benefit of the out-of-core implementation. Therefore a pre-processing parallelization scheme is developed in order to provide data-parallelization, which eliminates memory bottlenecks. With the out-of-core implementation (and the pre-processing parallelization), solutions of scattering problems involving up to 1.3 billion unknowns are achieved with 2 TB memory.

Solutions of billions of unknowns take vast amount of time and therefore it is utterly desirable to employ higher numbers of processing cores if available. But the memory increases with the number of processes and therefore the available memory constitutes a limitation on the number of processes. As a remedy, an MPI+OpenMP parallelization scheme is developed, which can employ all available processing cores in the system without requiring extra memory. The MPI+OpenMP parallelization reduces the CPU time by up to 50% and beats the pure-MPI parallelization in terms of time and memory efficiencies when high number of processing cores are employed.

PO approximation can provide fast solutions of scattering problems, and therefore a parallel PO solver is developed for fast and approximate solutions of large-scale problems. Results show that the PO solver is 60 times faster than MLFMA solver with a good accuracy in forward-scattering and specular-reflection angles, however the results are not necessarily accurate at a backscattering angle.

In order to hybridize the accuracy of MLFMA and the rapidity of PO, a hybrid IE and PO solution methodology is developed, which solves large and basic surfaces with PO and complex surfaces with IE. Scattering and radiation problems are solved for demonstrating the proposed methodology, and results show that hybrid solutions can solve problems 50% faster than pure-IE solutions without much loss of accuracy. However, the methodology does not provide considerable memory saving. The provided techniques in this thesis can be combined in order to solve large-scale problems with less computational resources.

Bibliography

[1] Ö. Ergül and L. Gürel, *The multilevel fast multipole algorithm (MLFMA) for solving large-scale computational electromagnetics problems*. Piscataway, NJ: Wiley-IEEE Press, 2014.

[2] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. 30, pp. 409–418, 1982.

[3] R. F. Harrington, *Field Computation by Method of Moments*. Piscataway, NJ: Wiley-IEEE Press, 1993. Reprinting of 1968 edition.

[4] O. Axelsson, *Iterative Solution Methods*. Cambridge, England: Cambridge University Press, 1996.

[5] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. M. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Philadelphia, PA: SIAM, 2 ed., 1994.

[6] C. C. Paige and S. M. A., "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Trans. Math. Software*, vol. 8, no. 1, pp. 43–71, 1982.

[7] P. Sonneveld, "CGS: A fast Lanczos-type solver for nonsymmetric linear systems," *SIAM J. Sci. and Stat. Comput.*, vol. 10, no. 1, pp. 36–52, 1989.

[8] H. A. Van der Vorst, "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems," *SIAM J. Sci. and Stat. Comput.*, vol. 13, no. 2, pp. 631–644, 1992.

[9] Ö. Ergül and L. Gürel, "Efficient solution of the electric and magnetic current combined-field integral equation with the multilevel fast multipole algorithm and block-diagonal preconditioning," *Radio Sci.*, vol. 44, no. 6, Dec. 2009.

[10] J. Song, C.-C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propag.*, vol. 45, pp. 1488–1493, Oct. 1997.

[11] M. Carr, M. Bleszynski, and J. L. Volakis, "A near-field preconditioner and its performance in conjunction with the BiCGstab(*ell*) solver," *Antennas Propagat. Mag.*, vol. 46, no. 2, pp. 23–30, Apr. 2004.

[12] R. Coiffman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas Propagat. Mag.*, vol. 35, no. 3, pp. 7–12, Jun. 1993.

[13] V. Rokhlin, "Diagonal forms of translation operators for the Helmholtz equation in three dimensions," *Appl. Comput. Harmon. Anal.*, vol. 1, pp. 82–93, 1993.

[14] L. Greengard, *The rapid evaluation of potential fields in particle systems*. PhD thesis, Dept. Comput. Sci., Yale Univ., New Haven, CT, 1987.

[15] Ö. Ergül and L. Gürel, "Accurate solutions of extremely large integral-equation problems in computational electromagnetics," *IEEE Proc.*, vol. 101, no. 2, pp. 342–349, Feb. 2013.

[16] W. C. Chew, J.-M. Jin, E. Michielssen, and J. Song, *Fast and efficient algorithms in computational electromagnetics*. Boston, MA: Artech House, 2001.

[17] S. Velamparambil and W. C. Chew, "Analysis and performance of a distributed memory multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 53, pp. 2719–2727, Aug. 2005.

[18] Ö. Ergül and L. Gürel, "Efficient parallelization of the multilevel fast multipole algorithm for the solution of large-scale scattering problems," *IEEE Trans. Antennas Propag.*, vol. 56, pp. 2335–2345, Aug. 2008.

[19] Ö. Ergül and L. Gürel, "A hierarchical partitioning strategy for an efficient parallelization of the multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 57, pp. 1740–1750, Jun. 2009.

[20] Ö. Ergül and L. Gürel, "Hierarchical parallelization of the multilevel fast multipole algorithm (MLFMA)," *IEEE Proc.*, vol. 101, no. 2, pp. 332–341, Feb. 2013.

[21] X.-M. Pan, W.-C. Pi, M.-L. Yang, Z. Peng, and X.-Q. Sheng, "Solving problems with over one billion unknowns by the MLFMA," *IEEE Trans. Antenas Propag.*, vol. 60, pp. 2571–2574, May. 2012.

[22] B. Michiels, J. Fostier, I. Bogaert, and D. De Zutter, "Full-wave simulations of electromagnetic scattering problems with billions of unknowns," *IEEE Trans. Antenas Propag.*, vol. 63, pp. 796–799, Feb. 2015.

[23] S. Velamparambil, W. C. Chew, and J. Song, "10 million unknowns: is it *that* big?," *IEEE Antenas Propagat. Mag.*, vol. 45, no. 2, pp. 43–58, Apr. 2003.

[24] B. Karaosmanoğlu, "Out-of-core implementation of the parallel multilevel fast multipole algorithm," Master's thesis, Dept. Elec. Electron. Eng., Bilkent Univ., Ankara, Turkey, 2013.

[25] M. Hidayetoğlu, B. Karaosmanoğlu, and L. Gürel, "Reducing MLFMA memory with out-of-core implementation and data-structure parallelization," in *Computational Electromagnetics Int. Workshop (CEM'13)*, İzmir, Turkey, 2013, pp. 34–37.

[26] M. Hidayetoğlu and L. Gürel, "MLFMA memory reduction techniques for solving large-scale problems," in *IEEE Int. Symp. Antennas Propagation USNC-URSI Radio Science Meeting (APS/URSI 2014)*, Memphis, TN, 2014, pp. 749–750.

[27] M. Hidayetoğlu and L. Gürel, "Full-wave and approximate solutions of large electromagnetic scattering problems," in *IEEE Int. Symp. Antennas Propagation North American Radio Science Meeting (APS/URSI 2015)*, Vancouver, Canada, 2015. In print.

[28] C. A. Balanis, *Advanced Engineering Electromagnetics*. Piscataway, NJ: Wiley, 1989.

[29] M. Hidayetoğlu and L. Gürel, "Accelerating hybrid integral-equation and physical-optics solutions with MLFMA," in *1st URSI Atlantic Radio Science Conf. (URSI AT-RASC)*, Gran Canaria, Spain, 2015. In print.

[30] R. F. Harrington, "Time-harmonic electromagnetic fields," 2001. Reprinting of 1961 edition.

[31] S. R. Rengarajan and Y. Lu, Rahmat-Samii, "The field equivalence principle: illustration of the establishment of the non-intutive null fields," *IEEE Antennas Propagat. Mag.*, vol. 42, no. 4, pp. 122–128, Aug. 2000.

[32] X. Q. Sheng, J.-M. Jin, J. Song, and W. C. Chew, "Solution of combined-field integral equation using multilevel fast multipole algorithm for scattering by homogeneous bodies," *IEEE Trans. Antennas Propag.*, vol. 46, pp. 1718–1726, Nov. 1998.

[33] R. D. Graglia, "On the numerical integration of the linear shape functions times the 3-D Green's function or its gradient on a plane triangle," *IEEE Trans. Antennas Propag.*, vol. 41, pp. 1448–1455, Oct. 1993.

[34] Ö. Ergül and L. Gürel, "Singularity of the magnetic-field integral equation and its extraction," *IEEE Antennas Wireless Propag. Lett.*, vol. 4, pp. 229–232, Aug. 2005.

[35] Ö. Ergül, *Accurate and efficient solutions of electromagnetic problems with the multilevel fast multipole algorithm*. PhD thesis, Dept. Elec. Electron., Bilkent Univ., Ankara, Turkey, 2009.

[36] J. S. Vitter, "External memory algorithms and data structures: dealing with massive data," *ACM Computing Surveys*, vol. 33, no. 2, pp. 209–271, Jun. 2001.

[37] M. Jung, E. H. Wilson III, W. Choi, J. Shalf, H. M. Aktulga, C. Yang, E. Saule, U. V. Catalyurek, and M. Kandemir, "Exploring the future of out-of-core computing with compute-local non-volatile memory," *Scienticic Programming*, vol. 22, no. 2, pp. 125–139, 2014.

[38] M. Yuan, , T. K. Sarkar, and B. Kolundzija, "Solution of large complex problems in computational electromagnetics using higher-order basis in MoM with out-of-core solvers," *IEEE Antennas Propagat. Mag.*, vol. 48, no. 2, pp. 55–62, Apr. 2006.

[39] X.-W. Zhao, Y. Zhang, H.-W. Zhang, D. Garcia-Doñoro, S.-W. Ting, T. H. Sarkar, and C.-H. Liang, "Parallel MoM-PO method with out-of-core technique for analysis of complex arrays on electrically large platforms," *Progress in Electromagnetics Research.*, vol. 108, pp. 1–21, 2010.

[40] M. Hidayetoğlu and L. Gürel, "Parallel implementation of the out-of-core MLFMA solver," in *Computational Electromagnetics Int. Workshop (CEM'15)*, İzmir, Turkey, 2015. In print.

[41] N. Drosinos and N. Koziris, "Performance comparison of pure MPI vs OpenMP parallelization models on SMP clusters," in *Int. Parallel Distributed Processing Symp. (IPDPS 2004)*, Santa Fe, NM, 2004.

[42] F. Capello and D. Etiemble, "MPI versus MPI+OpenMP on the IBM SP for the NAS benchmarks," in *Supercomputing, ACM/IEEE 2000 Conference*, Nov. 2000.

[43] H. Xuan, W. Tong, Z. Gong, and Y. Lan, "Implementation and performance analysis of hybrid MPI+OpenMP programming for parallel MLFMA on SMP cluster," in *3rd Int. Conf. Intelligent Control and Information Processing (ICICIP 2012)*, Dalian, China, 2012, pp. 744–748.

[44] H. Xuan, W. Tong, Y. Hu, Z. Gong, and Y. Hou, "Hybrid programming implementation of BiCG method on SMP cluster architecture," in *11th Int. Conf. on Computer and Information Science (IEEE/ACIS 2012)*, Shangai, China, 2012, pp. 470–475.

[45] J. M. Taboada, L. Landesa, M. G. Araújo, J. M. Bértolo, F. Obelleiro, J. L. Rodríguez, J. Rivero, and G. Gajardo-Silva, "Supercomputer solutions of extremely large problems in electromagnetics: from ten million to one billion unknowns," in *5th European Conf. Antennas and Propagation (EuCAP 2011)*, Rome, Italy, 2011, pp. 3221–3225.

[46] L. Gürel, H. Bağcı, J. C. Castelli, A. Cheraly, and F. Tardivel, "Validation through comparison: measurement and calculation of the bistatic radar cross section (BRCS) of a sthealth target," *Radio Sci.*, vol. 38, no. 3, pp. 12–1–12–10, Jun. 2003.

[47] A. Manyas and L. Gürel, "Memory-efficient multilevel physical optics algorithm for fast computation of scattering from three-dimensional complex targets," in *Computational Electromagnetics Int. Workshop (CEM'07)*, pp. 26–30, İzmir, Turkey, 2007, pp. 26–30.

[48] A. Manyas and L. Gürel, "Multilevel physical optics algorithm for fast solution of scattering problems involving nonuniform triangulations," in *IEEE Int. Symp. Antennas and Propagation (AP-S 2007)*, Honululu, HI, 2007, pp. 3277–3280.

[49] A. Manyas, "Memory-efficient multilevel physical optics algorithm for the solution of electromagnetic scattering problems," Master's thesis, Dept. Elec. Electron. Eng., Bilkent Univ., Ankara, Turkey, 2007.

[50] U. Jacobus and F. J. C. Meyer, "A hybrid physical optics/method of moments numerical technique: theory, investigation and ppplication," in *4th AFRICON Conf. Africa (IEEE AFRICON'96)*, Stellenbosch, South Africa, 1996, pp. 282–287.

[51] M. A. A. Moneum, Z. Shen, J. L. Volakis, and O. Graham, "Hybrid PO-MoM analysis of large axi-symmetric radomes," *IEEE Trans. Antennas Propag.*, vol. 49, pp. 1657–1666, Dec. 2001.

[52] Z.-L. Liu and C.-F. Wang, "An efficient iterative MoM-PO hybrid method for analysis of an onboard wire antenna array on a large-scale platform above an infinite ground," *IEEE Antennas Propagat. Mag.*, vol. 55, no. 6, pp. 69–78, Dec. 2013.

[53] R. E. Hodges and Y. Rahmat-Samii, "An iterative current-based hybrid method for complex structures," *IEEE Trans. Antennas Propag.*, vol. 45, pp. 265–276, Feb. 1997.

[54] L. Gürel, A. Manyas, and Ö. Ergül, "PO-MLFMA hybrid technique for the solution of electromagnetic scattering problems involving complex targets," in *2nd European Conf. Antennas Propagation (EuCAP 2007)*, Edinburgh, UK, 2007, pp. 1–5.

[55] A. Manyas, Ö. Ergül, and L. Gürel, "Hybridizing physical optics with MLFMA for efficient scattering computations of three-dimensional complex targets," in *Computational Electromagnetics Int. Workshop (CEM'09)*, İzmir, Turkey, 2009, pp. 69–72.

[56] Y. Zhang and H. Lin, "MLFMA-PO hybrid technique for efficient analysis of electrically large structures," *IEEE Antennas Wireless Propag. Lett.*, vol. 13, pp. 1676–1679, Aug. 2014.

Appendix A

Numerical Environment

A.1 Geometry Modelling and Mesh Generation

The geometries are designed using computer-aided-design (CAD) tools, e.g., I-DEAS, Siemens NX, or CATIA, and meshed with planar triangle patches which have equal side-lengths. Employment of RWG basis functions yields the number of unknowns N is equal to the number of edges. In closed geometries $N = 3T/2$, where T is the number of triangles. Fig. A.1 shows a Flamme [46] mesh.

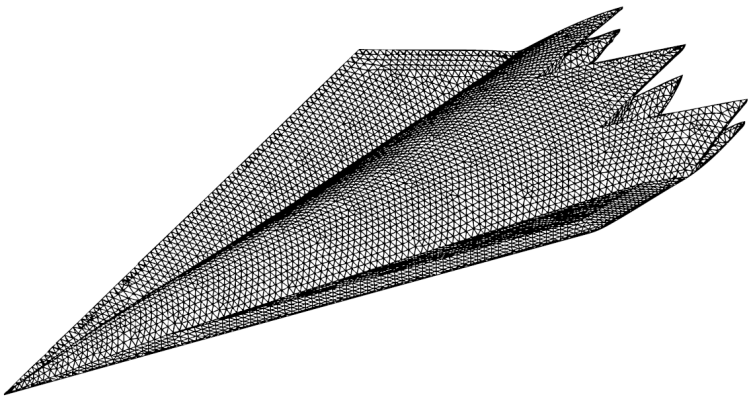

Figure A.1: A Flamme mesh (in courtesy of Ergül and Gürel [1]).

Available CAD programs cannot generate meshes when hundreds of million patches are involved. Therefore a mesh-refinement program is developed in order to generate large meshes. The program reads the input geometry and divides each triangle patch into multiple triangle patches. Fig. A.2 depicts two mesh-refinement schemes. One of them divides triangles into four, i.e., $T_{new} = 4T$, and the other divides them into six, i.e. $T_{new} = 9T$. The resulting number of unknowns are $N_{new} = 4N$ for the first scheme and $N_{new} = 9N$ for the second scheme. If constant mesh size is used, e.g., $\lambda/10$, the solution frequencies are $f_{new} = 2f$ and $f_{new} = 3f$, respectively.

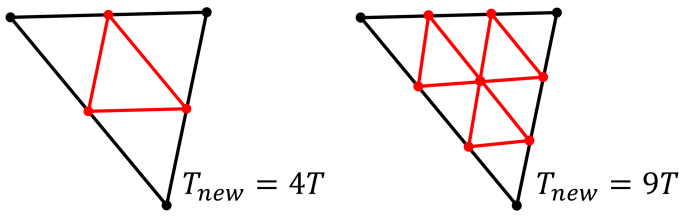

Figure A.2: Two mesh-refinement schemes.

When sphere geometries are involved, the newly defined vertices can be carried onto the sphere surface since the surface has analytical expression. For example, the new vertex $\boldsymbol{v}_{temp} = \boldsymbol{v}_1 + (\boldsymbol{v}_2 - \boldsymbol{v}_1)/2$ is carried at $\boldsymbol{v}_{new} = r\hat{\boldsymbol{v}}_{temp}$ on the sphere surface S, as depicted in Fig. A.3. The resulting number of unknowns for a sphere can be found using the formula $N = 8\sqrt{3}\pi r^2/m^2$, where m is the mesh size.

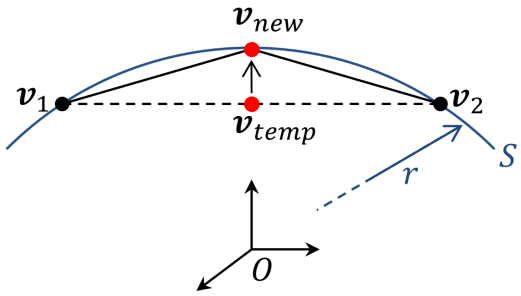

Figure A.3: Mesh refinement for sphere geometries.

A.2 System Specifications

The numerical results in the thesis are achieved using a 16-node (plus one master node) parallel computer cluster where each node has two 2.7 GHz Intel Xeon E5-4650 CPUs with eight processing cores. The CPUs have 20 MB cache and Intel's hyper-threading technology allows employing 32 processes/threads in each node. Fig. A.4 represents the parallel computer used in this work.

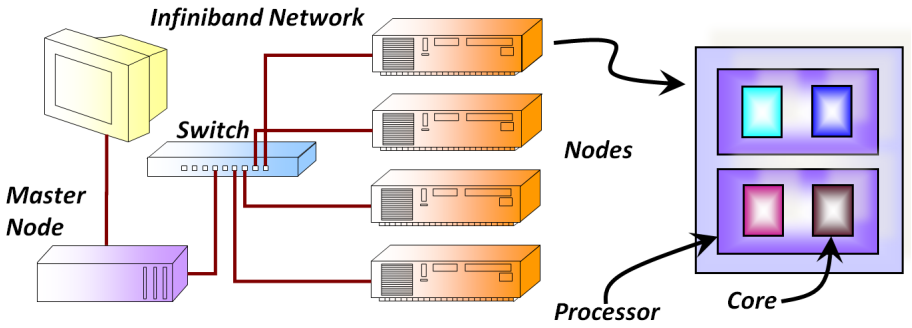

Figure A.4: A parallel computer cluster (in courtesy of Ergül and Gürel [1]).

Each node has 16x8 GB DDR3 random-access memory and is equipped with 400 GB Intel DC S3700 Series SSD for employing the out-of-core method with fast I/O operations. The SSDs are connected to 6 Gb/s SATA bus. The computing nodes are interconnected with a high-speed infiniband network. The QLogic IBA7322 QDR infiniband cards are connected to PCI Express 3.0.

Intel FORTRAN compiler 13.1.0.146 is used for compilations. For iterative solutions, PETSc version 3.3.5 is used along with MVAPICH2 1.7 MPI implementation.

Appendix B

Post Processing

B.1 Near-Field Calculations

Scattered fields, i.e., $\boldsymbol{E}^{sca}$ and $\boldsymbol{H}^{sca}$, can be found using the electric-field integral

$$\boldsymbol{E}^{sca}(\boldsymbol{r}) = i\omega\mu \int_S d\boldsymbol{r}' \, \overline{\boldsymbol{G}}(\boldsymbol{r}, \boldsymbol{r}') \cdot \boldsymbol{J}_{eq}(\boldsymbol{r}') \tag{B.1}$$

and the magnetic-field integral

$$\boldsymbol{H}^{sca}(\boldsymbol{r}) = \int_S d\boldsymbol{r}' \, \boldsymbol{J}_{eq}(\boldsymbol{r}') \times \nabla' g(\boldsymbol{r}, \boldsymbol{r}'), \tag{B.2}$$

respectively, where $\boldsymbol{J}_{eq}(\boldsymbol{r})$ is the equivalent surface current,

$$\overline{\boldsymbol{G}}(\boldsymbol{r}, \boldsymbol{r}') = \left[\overline{\boldsymbol{I}} - \frac{\nabla\nabla}{k^2}\right] g(\boldsymbol{r}, \boldsymbol{r}') \tag{B.3}$$

is the dyadic Green's function and

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

is the Green's function for the scalar Helmholtz's equation in Cartesian coordinates.

The equivalent current is discretized using N known basis functions and their coefficients in a numerical environment as $\boldsymbol{J}_{eq}(\boldsymbol{r}) \approx \sum_{n=1}^{N} x_n \boldsymbol{b}_n(\boldsymbol{r})$, hence the

electric-field and magnetic-field integrals become

$$\boldsymbol{E}^{sca}(\boldsymbol{r}) = i\omega\mu \sum_{n=1}^{N} x_n \int_{S_n} d\boldsymbol{r}' \, \overline{\boldsymbol{G}}(\boldsymbol{r}, \boldsymbol{r}') \cdot \boldsymbol{b}_n(\boldsymbol{r}'), \tag{B.5}$$

and

$$\boldsymbol{H}^{sca}(\boldsymbol{r}) = \sum_{n=1}^{N} x_n \int_{S_n} d\boldsymbol{r}' \, \boldsymbol{b}_n(\boldsymbol{r}') \times \nabla' g(\boldsymbol{r}, \boldsymbol{r}'), \tag{B.6}$$

respectively, where $\boldsymbol{b}_n(\boldsymbol{r})$ is the n^{th} basis function with its coefficient x_n and its domain S_n. The computational complexity of calculating the fields at S samples is $\mathcal{O}(SN)$. As a demonstration, a conducting sphere with 0.3 m radius centered at the origin is illuminated with a planewave with ϕ polarization and propagating in $-\hat{\boldsymbol{x}}$ direction at 2 GHz, i.e., $\boldsymbol{E}^{inc}(\boldsymbol{r}) = \hat{\boldsymbol{y}} e^{i\boldsymbol{r}\cdot\boldsymbol{k}}$, where $\boldsymbol{k} = -\hat{\boldsymbol{x}}\omega\sqrt{\mu_0\epsilon_0}$ is the wavevector, $\omega = 2\pi f$ is the angular frequency, f=2 GHz, and μ_0 and ϵ_0 are the magnetic permeability and the electric permittivity of free space, respectively. The scattered electric field $\boldsymbol{E}^{sca}$ in the vicinity of the sphere is shown in Fig. B.1 and the total electric field $\boldsymbol{E} = \boldsymbol{E}^{inc} + \boldsymbol{E}^{sca}$ is shown in Fig. B.2. Note that the total field is zero inside the conducting sphere, as expected.

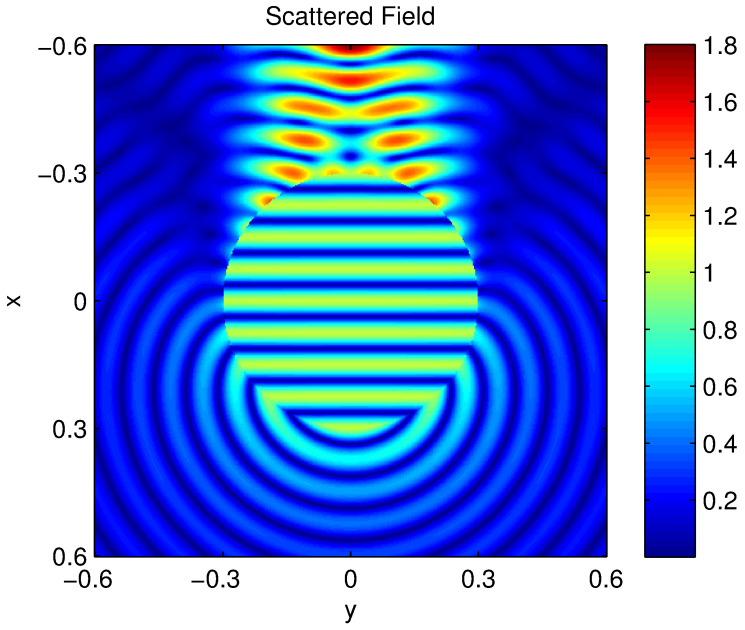

Figure B.1: Real magnitude of the scattered electric-field intensity.

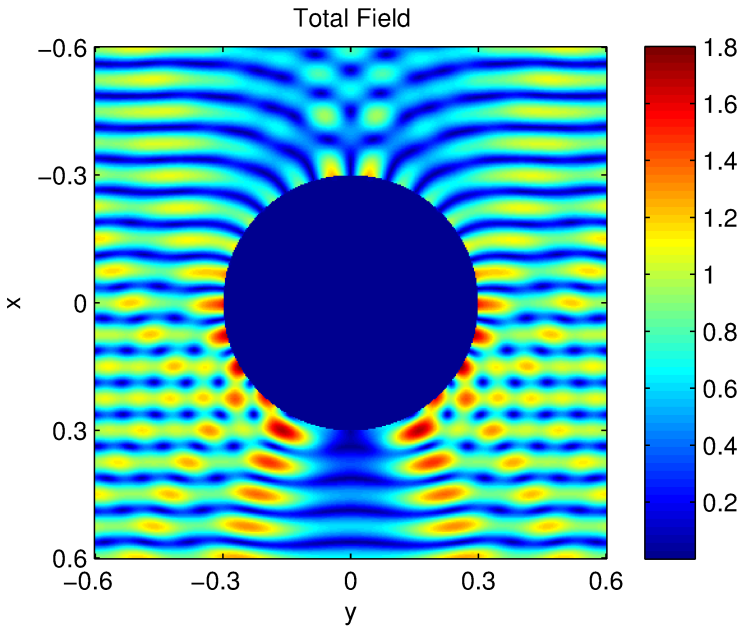

Figure B.2: Real magnitude of the total electric-field intensity.

B.2 Far-Field Calculations

In far-field zone, i.e., $r \gg \lambda$, the scattered electric field can be approximated as

$$\boldsymbol{E}^{sca}(\boldsymbol{r}) \approx \left[\hat{\boldsymbol{\theta}}\hat{\boldsymbol{\theta}} \cdot \boldsymbol{F}(\theta,\phi) + \hat{\boldsymbol{\phi}}\hat{\boldsymbol{\phi}} \cdot \boldsymbol{F}(\theta,\phi)\right] \frac{e^{ikr}}{r}, \tag{B.7}$$

where

$$\boldsymbol{F}(\theta,\phi) = ik\eta \int_S d\boldsymbol{r}\, \boldsymbol{J}_{eq}(\boldsymbol{r}') e^{-ik\hat{\boldsymbol{r}}\cdot\boldsymbol{r}'} \tag{B.8}$$

is the vector current moment. When $\boldsymbol{J}_{eq}$ is discretized with N known basis functions,

$$\boldsymbol{F}(\theta,\phi) = ik\eta \sum_{n=1}^{N} \int_{S_n} d\boldsymbol{r}\, x_n \boldsymbol{b}_n(\boldsymbol{r}') e^{-ik\hat{\boldsymbol{r}}\cdot\boldsymbol{r}'}. \tag{B.9}$$

The magnetic field can be written as

$$\boldsymbol{H}^{sca}(\boldsymbol{r}) \approx \frac{1}{\eta_0}\hat{\boldsymbol{r}} \times \boldsymbol{E}^{sca}(\boldsymbol{r}), \tag{B.10}$$

where η_0 is the intrinsic impedance of free space.

Radar cross-section (RCS) in φ polarization is defined as

$$\begin{aligned} RCS_{\varphi}(\theta,\phi) &= \lim_{r\to\infty} 4\pi r^2 P_{\varphi}^{rad}(\boldsymbol{r}) \\ &= \lim_{r\to\infty} 4\pi r^2 \left|\boldsymbol{E}_{\varphi}^{sca}(\boldsymbol{r})\right|^2 \\ &= 4\pi \left|\hat{\boldsymbol{\varphi}}\cdot\boldsymbol{F}(\theta,\phi)\right|^2 . \end{aligned} \tag{B.11}$$

The unit of RCS is meter square (m^2). When sphere objects are involved, RCS is normalized by the cross-section of the sphere, i.e.,

$$RCS_{\varphi}(\theta,\phi) = \frac{4}{a^2}\left|\hat{\boldsymbol{\varphi}}\cdot\boldsymbol{F}(\theta,\phi)\right|^2 , \tag{B.12}$$

where a is the sphere radius. When normalized with its cross-section, the sphere's RCS becomes unitless. The computational complexity of RCS calculations is $\mathcal{O}(DN)$, where D is the number of directions. Fig. B.3 shows the RCS of the sphere with 2λ.

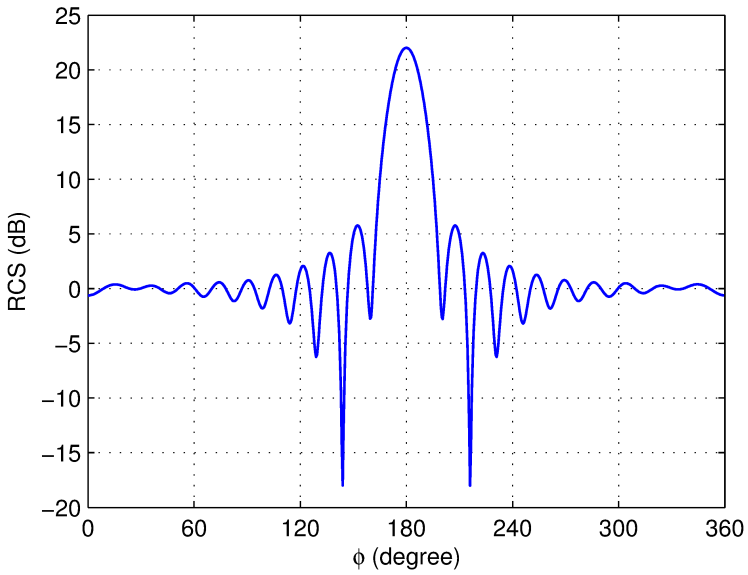

Figure B.3: Bistatic co-polar RCS of the sphere with the radius of 2λ.