Complexity Reduction in Solution of Scattering Problems on Well-Conducting 3-D Object With $\mathcal{H}-$Matrices Accelerated Surface-Volume-Surface Electric Field Integral Equation

by

Jamiu Babatunde Mojolagbe

A Thesis Submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba, Canada

Abstract

The Surface-Volume-Surface Electric Field Integral Equation (SVS-EFIE) is applicable to the solution of the scattering problems on both lossless dielectric and highly conducting metal objects. When the solution of scattering and radiation problems on metal objects is sought the rapidly attenuating field behaviour due to skin-effect can be exploited to reduce the complexity of the Method of Moments (MoM) discretized SVS-EFIE. Among these strategies is the removal of the volume mesh corresponding to the part of the scatterer in which field is expected to attenuate below certain threshold and truncation of the range in MoM interactions due to their attenuation governed by the metal medium Green's function. In this work, the qualitative description of these techniques is provided as well as numerical results demonstrating their validity.

Publications

The author contributed to the following publications while working towards the MSc

- ✓ **Jamiu Mojolagbe**, Reza Gholami, and Vladimir Okhmatovski. On Complexity Reduction in Solution of Scattering Problems on Well-Conducting 3D Objects with Surface-Volume-Surface EFIE. In Applied Computational Electromagnetics Society Symposium (ACES), 2018 International, pages 1-2. IEEE, 2018.

- ✓ Farhad Sheikh Hosseini Lori, Anton Menshov, Reza Gholami, **Jamiu Babatunde Mojolagbe**, and Vladimir I Okhmatovski. Novel Single - Source Surface Integral Equation for Scattering Problems by 3-D Dielectric Objects. IEEE Transactions on Antennas and Propagation, 66(2):797−807, 2018.

- ✓ Reza Gholami, **Jamiu B Mojolagbe**, Anton Menshov, Farhad Sheikh Hosseini Lori, and Vladimir I Okhmatovski (Submitted 2018). An H-Matrix Accelerated Solution of a Surface-Volume-Surface Electric Field Integral Equation for 3-D Scattering Problem.

- ✓ Zhuotong Chen, Reza Gholami, **Jamiu Babatunde Mojolagbe**, and Vladimir I Okhmatovski. Formulation of Surface-Volume-Surface-EFIE for Solution of 3-D Scattering Problems on Composite Dielectric Objects. IEEE Antennas and Wireless Propagation Letters, 2018.

Acknowledgments

My profound gratitude goes to my academic advisor, in person of Dr Vladimir Okhmatovski, for being there for me at all time and for giving me all the supports and training which he dimmed fit for me. My sincere appreciation also goes to Amy Dario without whom I would not have been a student of University of Manitoba; In fact you are a rare gem.

Also, my sincere appreciation goes to those with whom I have had the pleasure to work during my MSc sojourn. I would like to thank all my research mates − Reza Gholami, Anton Menshov, Jamil Feroj, Ammar Aljamal, Osman Md Goni, Farhad Hosseini, Zhuotong Chen, Shucheng Zheng, Hosen Shakander − for their supports.

My sincere gratitude also goes to my friends. I would like to thank Idris Abolaji, Rauf Badmus, Azeez Badmus, Yusuf Ibrahim, Ajileye AbdulAziz and Oluyemi Aina for their supports. I also appreciate my amiable mentor, in person of Dr Toyin Akinde, for his moral, emotional and financial supports. I would like to thank Dr Ibrahim Adeyanju and Prof. R.O.R Kalilu for their supports and guidance.

Nobody has been more supportive to me in this academic pursuit than the mem-

bers of my family. I would like to thank my parents, Mr & Dns Majolagbe for their supports and prayers. I am grateful to my siblings, Bolanle (Mummy Samuel), Biodun and Sakirat for their supports. Most importantly, I am using this media to thank my darling wife, Rashidat, for being there for me during the tough times and also to my daughters, Fatimah and Hafsoh, for their endurance.

This academic journey and/or research work would have been impossible or rough and tough without adequate financial backing. Hence, I would like to thank Natural Sciences and Engineering Research Council (NSERC) for providing the necessary funding. I am also indebted to WestGrid (Compute Canada) for the provision of High Performance Computing (HPC) resources used for obtaining the results of the test cases presented in this work.

To Holy Prophet Muhammad (Peace Be Upon Him).

Contents

List of Figures

List of Tables

Note on Acronyms and Symbols

The common acronyms that will be frequently encountered in this thesis are given in table 0.1.

Table 0.1: Frequently Used Acronyms

Acronym	Meaning
EM	Electromagnetic
CEM	Computational Electromagnetic
FEM	Finite Element Method
FD	Finite Difference
IE	Integral Equation
DE	Differential Equation
MoM	Method of Moments
PDE	Partial Differential Equation
SVS-IE	Surface Volume Surface Integral Equation

Continued on next page

Table 0.1 – continued from previous page

Acronym	Meaning
SIE	Surface Integral Equation
SSIE	Single Source Surface Integral Equation
VIE	Volume Integral Equation
V-EFIE	Volume Electric Field Integral Equation
S-EFIE	Surface Electric Field Integral Equation
PEC	Perfect Electric Conductor
EFIE	Electric Field Integral Equation
SVS-EFIE	Surface Volume Surface Electric Field Integral Equation
MFIE	Magnetic Field Integral Equation
CFIE	Combined Field Integral Equation
ACA	Adaptive Cross Appoximation
SVD	Singular Value Decomposition
LU	Lower and Upper
RWG	Rao-Wilton-Glisson
SLAE	System of Linear Algebra Equation
$\mathcal{H}-$Matrices	Hierarchical Matrices
$\mathcal{H}-$Matrix	Hierarchical Matrix
SE	Skin Effect

Continued on next page

Table 0.1 – continued from previous page

Acronym	Meaning
SD	Skin Depth
2-D	Two Dimensional
3-D	Three Dimensional
SAR	Specific Absorption Rate
CPU	Central Processing Unit

Another important portion of this thesis that needs additional notice are the use of symbols and notations. While the choice of font-type and font-face are itemized below, the list of symbols frequently used are presented in table 0.2.

- **Spatial-vectors:** bold face letters are used for vectors and matrices, such as $\boldsymbol{E}$, $\boldsymbol{H}$, and $\boldsymbol{J}$.
- **Scalar integral operators:** are denoted by upper-case calligraphic letters such as $\mathcal{T}$.
- **Vector integral operators:** are denoted by upper-case bold calligraphic letters such as $\boldsymbol{\mathcal{T}}$.
- **Vector matrices:** are denoted by upper-case bold letters such as $\boldsymbol{Z}$.
- **Scalar matrices:** are denoted by upper-case letters such as Z.

- **Dyads:** are denoted with upper-case letters with two overhead lines such as $\bar{\bar{I}}$ and $\overline{\overline{G}}_{e\epsilon}$.

- **Spatial derivative operators:** spatial gradient, divergence, and curl operators are given as $(\boldsymbol{\nabla})$,$(\boldsymbol{\nabla}\cdot)$ and $(\boldsymbol{\nabla}\times)$.

- **Integration differentials:** dl is the differential element for line integrals, while dS stands for differential operation on surface patches of boundary elements and dV means the differential operation on volumetric elements .

Table 0.2: Frequently Used Symbols

Symbol	Meaning
$\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, $\hat{\mathbf{z}}$	Unit vectors in the x, y and z directions.
$\hat{\mathbf{n}}$	Normal unit vector outward to the boundary.
$\hat{\mathbf{t}}$	Tangential unit vector to the boundary.
$\boldsymbol{r}$, $\boldsymbol{r}'$	Position vectors in the 3-D Cartesian coordinate system.
ϵ_0	Permittivity of free-space.
ϵ	Relative complex permittivity of the scatterer.
σ	Conductivity of the scatterer.
μ_0	Permeability of free-space.
μ_r	Relative permeability of the scatterer.

Continued on next page

Table 0.2 – continued from previous page

Symbol	Meaning
k_0	Wave number of free-space.
k_ϵ	Wave number inside the scatterer.
ω	Radial frequency.
f	Frequency of operation.
$\mathtt{t}$	Time variable.
λ	Wavelength.
Γ	Gram matrix.
$\boldsymbol{E}^{\text{inc}}$	Time-harmonic incident electric-field for a transmitter.
$\boldsymbol{H}^{\text{inc}}$	Time-harmonic incident magnetic-field for a transmitter.
$\boldsymbol{E}^{\text{scat}}$	Time-harmonic scattered electric-field.
$\boldsymbol{E}$	Time-harmonic total electric-field.
$\boldsymbol{J}_k$	Polarization current density.
$\boldsymbol{J}$	Fictitious SVS-EFIE surface source density.
$\overline{\overline{G}}_{e0}$	Dyadic electric field Green's function of free space.
$\overline{\overline{G}}_{e\epsilon}$	Dyadic electric field Green's function of homogeneous space with permittivity ϵ.
∂V	Boundary of the 3-D object.
V	Volume of the 3-D object.

Continued on next page

Table 0.2 – continued from previous page

Symbol	Meaning
ℓ	Length variable.
$\boldsymbol{\nabla}$	Gradient operator.
$\boldsymbol{\nabla}\cdot$	Divergence operator.
$\boldsymbol{\nabla}\times$	Curl operator.
$\boldsymbol{\nabla}\times\boldsymbol{\nabla}\times$	Curl curl operator.
∇^2	The Laplacian.
$(\cdot)^{-1}$	Inverse operator.
$\|\cdot\|$	L_2- norm or Euclidean norm.
$\langle\cdot\,,\cdot\rangle$	Inner product.

Chapter 1

Introduction

1.1 Background Information

The Surface-Volume-Surface Electric Field Integral Equation (SVS-EFIE) formulation is used for solving scattering and radiation problems on both dielectric and highly conducting metal objects [20, 21, 23–25]. SVS-EFIE takes advantage of the relationship between the surface and volumetric currents. It reduces radiation or scattering problems on penetrable object to the problem of determining a single unknown surface current density on the boundary of the object [20, 21, 23–25].

Solving a scattering or radiation problem may involve the use of methods such as finite difference (FD) and finite element method (FEM) that directly discretize pertinent partial differential equations (PDEs). Alternatively, the problem can be converted into the form of a surface integral equation (SIE) or a volume integral

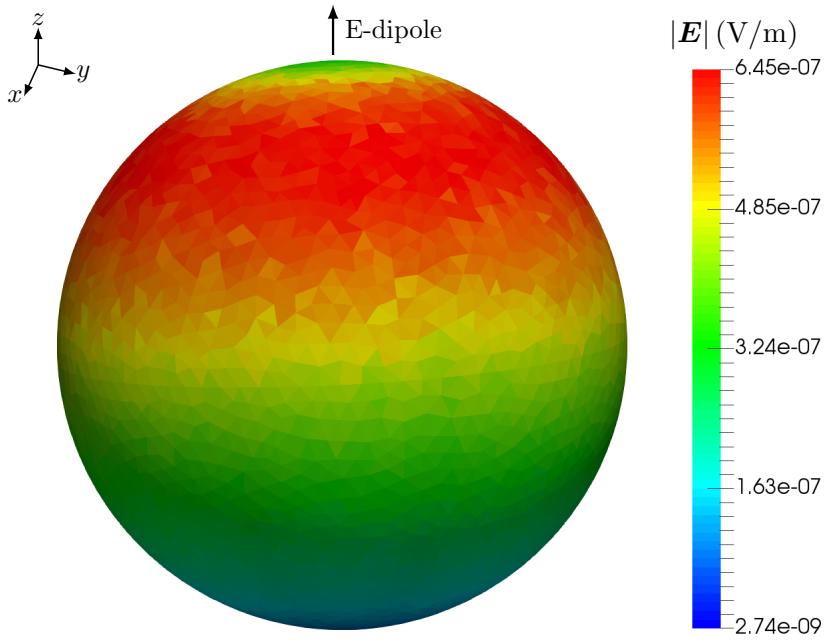


Figure 1.1: Magnitude of the total electric field inside Lead (Pb) sphere produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ obtained by the $\mathcal{H}$-matrix accelerated SVS-EFIE solver at 30 GHz

equation (VIE). When the solution is sought via the SIEs, it can be formulated into Electric Field Integral Equation (EFIE), Combined Field Integral Equation (CFIE), Magnetic Field Integral Equation (MFIE), and other types of SIEs [29].

EFIE can be categorized further based on the problem domain containing the unknown field quantities under consideration. The Surface - Electric Field Integral Equation (S-EFIE) localizes the unknown fields to the surface of the object and, hence, requires only the surface discretization. Alternatively, Volume - Electric Field Integral Equation (V-EFIE) formulated with respect to the unknown field quantities localized to the volume requires discretization of the object's volume [15, 29]. In each of these cases, both the electric field and magnetic field Green's functions are featured.

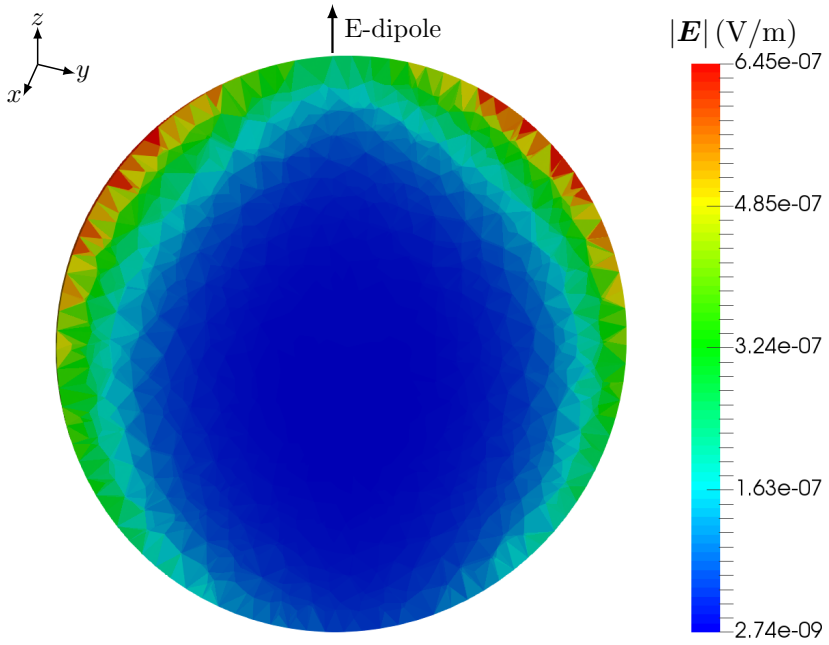


Figure 1.2: Cross-Section: Magnitude of the total electric field inside Lead (Pb) sphere produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ obtained by the $\mathcal{H}$-matrix accelerated SVS-EFIE solver at 30 GHz

However, when the solution is sought through SVS-EFIE, its method of moment (MoM) discretization involves the creation of both the surface and volume meshes because of the presence of Surface-to-Volume and Volume-to-Surface operators [20, 24]. Interestingly, it features only the electric field Green's function, its unknown field quantities are localized to the surface of the analysed scatterer in-place of its volume, and it is derivative free when compared to traditional Integral Equations (IEs) [20–22, 24, 25].

Usually, MoM discretization of a scattering or radiation problem yields dense matrices which are expensive to solve and also memory-wise costly. The concept of Hierarchical Matrices ($\mathcal{H}-$Matrices) is introduced as fast direct approach for the formation of the MoM matrix obtained from the discretization of SVS-EFIE – which has an inherent product of non-square matrices from Surface-to-Volume and Volume-to-Surface operators. The technique of $\mathcal{H}-$Matrices facilitates rank based compression of the SVS-EFIE operators and it also efficiently handles matrix operations like matrix-vector multiplication, matrix - matrix multiplications, matrix addition and inversion [5, 12–14].

There are several applications of solutions of scattering and radiation problems on penetrable conducting objects, viz., antenna design, high speed interconnects, analysis of plasmonic structures and so on [1, 8, 18, 32]. The field calculation throughout the volume of the lossy body tissues are required in the biomedical EM applications for determining the specific absorption rate (SAR) or field penetration depth [1, 9],

SVS-EFIE is applicable for solution of such problems.

1.2 Purpose of the study

When the solution of the scattering problem is sought for penetrable metal objects (at high frequency), the solution of SVS-EFIE becomes extremely large in terms of the memory and time requirements. It then becomes seemingly impractical to achieve within a reasonable time frame [28, 39] because of the presence of a large amount of volumetric mesh elements.

In this work, a certain portion of the volume mesh corresponding to region in which the field is expected to have attenuated is removed as shown in Figure (1.3). The field attenuates very fast (Figures (1.1) and (1.2)) due to the skin-effect below a certain threshold and the truncation of the range, governed by the metal medium Green's function, can be exploited to reduce the complexity of the MoM discretization of SVS-EFIE. In short, these applied techniques involve the reduction in the memory requirement and the computational cost of the problem through: (1) the extraction of the portion of the volumetric mesh elements a few skin depths away where the field is believed to have vanished and (2) truncation of the range of interactions between the sources and observers.

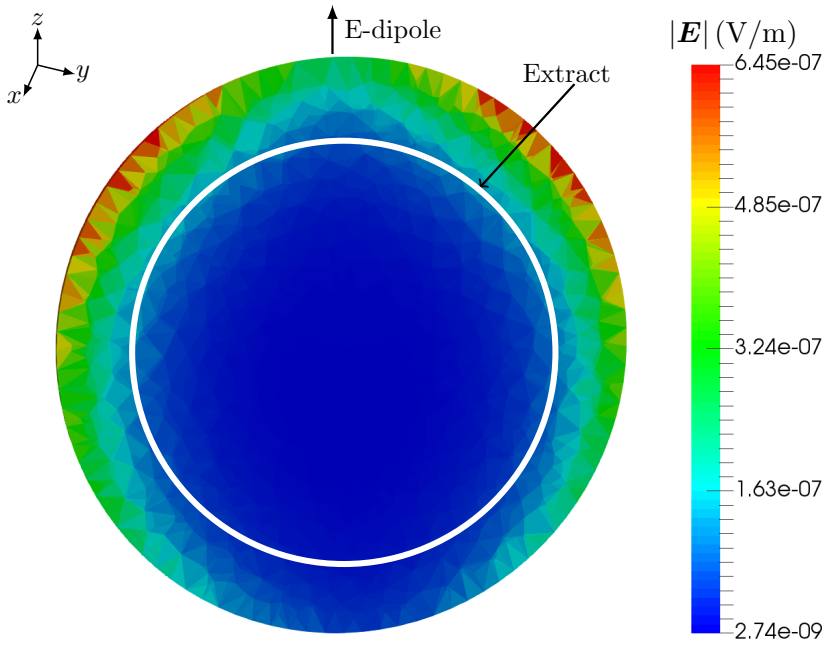


Figure 1.3: Cross-Section: Magnitude of the total electric field inside Lead (Pb) sphere produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ obtained by the $\mathcal{H}$-matrix accelerated SVS-EFIE solver at 30 GHz showing the portion to be extracted

1.3 Statement of the Hypothesis

In this research work, it is hypothesized that:

1. Exploitation of the metal object's skin-effect and truncation of the range of interaction does not significantly affect the accuracy of the solution.

2. There is a reduction in the time to solution and/or the complexity of the solution of MoM discretization of SVS-EFIE for penetrable well-conducting metal objects (at high frequency) with skin-effect exploitation and interaction range truncation.

3. These techniques facilitate partial meshing of the volumetric portion of the scatterer in question so that meshing is done only at few skin-depths into the volume.

4. At high frequency, the MoM discretization of SVS-EFIE for well-conducting metal object is comparable to the MoM discretization of S-EFIE in terms of its combined surface and volume degrees of freedom.

1.4 Research Questions

Applying the volumetric element extraction and interaction range truncation strategies would, undoubtedly, influence the accuracy of the solution. Hence, the questions whose answers are desired are:

1. "Can the effect of the volumetric mesh elements extraction be measured?"

2. "Does the application of the volumetric mesh elements extraction and the truncation of range of interaction introduce additional computational overheads?"

The answers to these questions are the justifications for the importance of the application of the volumetric mesh element extraction and interaction range truncation techniques when solving a scattering problem on a conducting metal object. This implies that without additional computational overheads, solving a scattering problem with the application of these techniques is better off than the one without it. Similarly, being able to measure the effect of the volumetric mesh elements extraction avails the user of these strategies to foresee the likely effect of choosing a certain extraction width.

1.5 Significance of the study

For a given scattering or radiation problem involving a penetrable metal object, the volumetric elements which introduce additional computational complexity to the solution of MoM discretization of SVS-EFIE are reduced by extracting other parts of the mesh a few skin-depths away. Employing the said extraction technique followed by the truncation of the range of interaction greatly influence the complexity of the problem in-terms of the time to solution and memory/storage requirements to obtain a solution.

1.6 Thesis Structure

Chapter (2) discusses the main foundational theories used in this research work. Here, the review of existing literatures are done. Mathematical formulations of the volume equivalence principle and SVS-EFIE are presented.

Chapter (3) briefly enumerates the basic features of $\mathcal{H}-$Matrices. It discusses each of the features with appropriate examples, figures and illustrations where necessary.

Chapter (4) elaborates the volumetric mesh extraction technique, range of interaction truncation procedures, and $\mathcal{H}-$Matrices implementation of SVS-EFIE. It discusses the validity checks and introduces a parameter for validating the extent of the extraction effect. Various test cases conducted are also presented in this chapter.

Chapter (5) contains the results obtained from the application of the new techniques introduced in Chapter 3. The results of the validity checks are also presented and discussed. Similarly, there is discussion on the hypothesis.

Chapter (6) is the panorama of the entire research work. This chapter also summarizes the theoretical and practical implications of using the new techniques. It also contains suggestions for future research on this topic.

Chapter 2

Overview of Equivalence Principle and SVS-EFIE

2.1 Introduction

As briefly mentioned in Chapter (1), the focus of this research work lies in the improvement of the solution of MoM discretized SVS-EFIE formulation of a scattering problem on a conducting metal object. Henceforth, the core theoretical foundations of this research work lie in the SVS-EFIE formulation and its $\mathcal{H}-$Matrices based MoM discretization.

The SVS-EFIE formulation combines the ideas of the Single-Source Surface Integral Equation (SSIE) [30,36] with the traditional volume equivalence principle [17,29]. It was first presented in [21, 23–25] for the analysis of 2-D scattering problems on

homogeneous, piece-wise homogeneous dielectric and well-conducting objects. The detailed description of MoM discretization of the 2-D SVS-EFIE formulation was introduced in [22]. Furthermore, the application of SVS-EFIE is not limited to solving problems on homogeneous objects but can also be applied to the analysis of scattering problems on the multi-layered media [40]. While 2-D scattering problem analysis [21–25] provides insights into the correctness of the formulation itself, its full wave analysis is much more desired and sought for. Therefore, the 3-D scattering problem analysis on homogeneous dielectric objects using MoM discretized SVS-EFIE formulation has been done [20].

The advantages of the SVS-EFIE include the presence of one product of electric field type integral operators, provision for a mixed potential formulation free of hyper-singular integral and other advantages related to the solution of practical problems as well as MoM implementation [20, 21, 23–25, 40]. In contrast to the advantages of the SVS-EFIE, sit the bottlenecks created by the cost of computing field translations from the scatterer surface to its volume (Surface-to-Volume operator computation) and then from its volume back to the surface (Volume-to-Surface operator computation). The result of these translations in SVS-EFIE is that both the surface and the volume of the scatterer have to be discretized. Therefore, considering the number of volumetric mesh elements that will be generated at high frequency, [28, 39] argued that SVS-EFIE formulation is not applicable for solving scattering problems at high frequency.

The following sections detail the derivation of SVS-EFIE as presented in [20, 34] for the interested reader. The final equations used in this thesis can be found in Section (2.5).

2.2 Time – Harmonic Fields

In a region illuminated by time-harmonic electromagnetic field ($e^{j\omega \mathtt{t}}$) having electric current $\boldsymbol{J}$ and magnetic current $\boldsymbol{M}$, the Maxwell's equations are given by

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

$$\boldsymbol{\nabla} \times \boldsymbol{H}(\boldsymbol{r}) = j\omega\varepsilon \boldsymbol{E}(\boldsymbol{r}) + \boldsymbol{J}, \tag{2.2}$$

$$\boldsymbol{\nabla} \cdot (\varepsilon \boldsymbol{E}(\boldsymbol{r})) = \boldsymbol{\rho}_e, \tag{2.3}$$

$$\boldsymbol{\nabla} \cdot (\mu \boldsymbol{H}(\boldsymbol{r})) = \boldsymbol{\rho}_m, \tag{2.4}$$

where, $\boldsymbol{E}$ and $\boldsymbol{H}$ are the electric and magnetic fields respectively. The cyclic frequency is denoted by ω, $\boldsymbol{\rho}$ is the volume charge density, $\boldsymbol{J}$ is the electric current density, ε is the dielectric permittivity, μ is the permittivity of free space. The position vector is given by $\boldsymbol{r}$, $j = \sqrt{-1}$, $\boldsymbol{\rho}_e$ is the electric charge density due to electric current source, and $\boldsymbol{\rho}_m$ is the magnetic charge density due to magnetic current source. Note that magnetic charges have not been shown to exist, but it is often used as a mathematical convention.

The electric and magnetic fields in the Maxwell's equations, for simplicity, can

be broken down into one due to the electric current source and the other due to the magnetic current source [17] thus:

$$\boldsymbol{E} = \boldsymbol{E}_e + \boldsymbol{E}_m, \tag{2.5}$$

$$\boldsymbol{H} = \boldsymbol{H}_e + \boldsymbol{H}_m, \tag{2.6}$$

where $\boldsymbol{E}_e$ is the electric field due to electric current source, $\boldsymbol{E}_m$ is the electric field due to the magnetic current source, $\boldsymbol{H}_e$ and $\boldsymbol{H}_m$ are the magnetic field due to the electric and magnetic current sources respectively, $\boldsymbol{E}$ and $\boldsymbol{H}$ denote the total electric and magnetic fields respectively.

Electric field $\boldsymbol{E}_e$ as used in equation (2.5) satisfies

$$\boldsymbol{\nabla} \times \boldsymbol{E}_e = -j\omega\mu\boldsymbol{H}_e, \tag{2.7}$$

$$\boldsymbol{\nabla} \times \boldsymbol{H}_e = j\omega\varepsilon\boldsymbol{E}_e + \boldsymbol{J}, \tag{2.8}$$

$$\boldsymbol{\nabla} \cdot (\varepsilon\boldsymbol{E}_e) = \boldsymbol{\rho}_e, \tag{2.9}$$

$$\boldsymbol{\nabla} \cdot (\mu\boldsymbol{H}_e) = \boldsymbol{0}, \tag{2.10}$$

where the position-vector $\boldsymbol{r}$ has been expunged for the sake of clarity and conciseness.

Again, $\boldsymbol{E}_m$ and $\boldsymbol{H}_m$ are governed by

$$\boldsymbol{\nabla} \times \boldsymbol{E}_m = -j\omega\mu\boldsymbol{H}_m - \boldsymbol{M}, \tag{2.11}$$

$$\boldsymbol{\nabla} \times \boldsymbol{H}_m = j\omega\varepsilon\boldsymbol{E}_m, \tag{2.12}$$

$$\boldsymbol{\nabla} \cdot (\varepsilon\boldsymbol{E}_m) = \boldsymbol{0}, \tag{2.13}$$

$$\boldsymbol{\nabla} \cdot (\mu\boldsymbol{H}_m) = \boldsymbol{\rho}_m. \tag{2.14}$$

The magnetic field due to the electric current source $\boldsymbol{B}_e = \mu\boldsymbol{H}_e$, which is a vector function in a solenoidal form, can be represented as a curl of another vector – magnetic vector potential $\boldsymbol{A}$– as in

$$\boldsymbol{B}_e = \boldsymbol{\nabla} \times \boldsymbol{A}, \tag{2.15}$$

$$\boldsymbol{H}_e = \frac{1}{\mu}\boldsymbol{\nabla} \times \boldsymbol{A}. \tag{2.16}$$

Inserting equation (2.15) into equation (2.7), results in

$$\boldsymbol{\nabla} \times \boldsymbol{E}_e + j\omega\boldsymbol{\nabla} \times \boldsymbol{A} = \boldsymbol{0}, \qquad \text{therefore,} \tag{2.17}$$

$$\boldsymbol{\nabla} \times (\boldsymbol{E}_e + j\omega\boldsymbol{A}) = \boldsymbol{0}. \tag{2.18}$$

Equation (2.18) can be satisfied by introducing the scalar potential as in $\boldsymbol{E} = -\boldsymbol{\nabla}\phi$,

such that

$$\boldsymbol{E}_e + j\omega\boldsymbol{A} = -\boldsymbol{\nabla}\phi, \tag{2.19}$$

then substituting into equation (2.8), equations (2.16) and (2.19) yields

$$\boldsymbol{\nabla} \times \left(\frac{1}{\mu}\boldsymbol{\nabla} \times \boldsymbol{A}\right) = -j\omega\varepsilon\boldsymbol{\nabla}\phi + \omega^2\varepsilon\boldsymbol{A} + \boldsymbol{J}, \tag{2.20}$$

hence, in a homogeneous media, equation (2.20) becomes

$$\boldsymbol{\nabla}\left(\boldsymbol{\nabla}\cdot\boldsymbol{A}\right) - \nabla^2\boldsymbol{A} = -j\omega\mu\varepsilon\boldsymbol{\nabla}\phi + k^2\boldsymbol{A} + \mu\boldsymbol{J}, \tag{2.21}$$

where k is the free-space wave number defined as $k = \omega\sqrt{\mu\varepsilon}$.

As established in equation (2.15), equation (2.21) can be simplified by setting the divergence of vector potential $\boldsymbol{A}$ as

$$\boldsymbol{\nabla}\cdot\boldsymbol{A} = -j\omega\mu\varepsilon\phi, \tag{2.22}$$

thus, plugging equation (2.22) into equation (2.21) gives

$$\boldsymbol{\nabla}\left(\boldsymbol{\nabla}\cdot\boldsymbol{A}\right) - \nabla^2\boldsymbol{A} = \boldsymbol{\nabla}\left(\boldsymbol{\nabla}\cdot\boldsymbol{A}\right) + k^2\boldsymbol{A} + \mu\boldsymbol{J}, \tag{2.23}$$

$$\nabla^2\boldsymbol{A} + k^2\boldsymbol{A} = -\mu\boldsymbol{J}, \tag{2.24}$$

notice that re-arranging equation (2.23) produces the equation (2.24) which is in the form of inhomogeneous Helmholtz equation.

Assuming that the value of the vector potential $\boldsymbol{A}$ has been found, the electric field due to the electric current source $\boldsymbol{E}_e$ can be expressed from equation (2.19) as

$$\boldsymbol{E}_e = -j\omega\boldsymbol{A} - \boldsymbol{\nabla}\phi, \tag{2.25}$$

$$\boldsymbol{E}_e = -j\omega\boldsymbol{A} + \frac{1}{j\omega\mu\varepsilon}\boldsymbol{\nabla}\left(\boldsymbol{\nabla}\cdot\boldsymbol{A}\right), \tag{2.26}$$

where equation (2.26) is obtained from equations (2.25) and (2.22).

Following the same approach used to obtain (2.26) from equation (2.7), the electric field $\boldsymbol{E}_m$ due to the magnetic current source can be derived in the same manner using (2.8) as:

$$\boldsymbol{E}_m = -\frac{1}{\varepsilon}\boldsymbol{\nabla}\times\boldsymbol{F}, \tag{2.27}$$

where $\boldsymbol{F}$ is the electric vector potential.

Conclusively, inserting equations (2.26) and (2.27) into equation (2.5) gives

$$\boldsymbol{E} = -j\omega\boldsymbol{A} + \frac{1}{j\omega\mu\varepsilon}\boldsymbol{\nabla}\left(\boldsymbol{\nabla}\cdot\boldsymbol{A}\right) - \frac{1}{\varepsilon}\boldsymbol{\nabla}\times\boldsymbol{F}. \tag{2.28}$$

2.3 Field – Source Relationship

In equation (2.28) derived in section (2.2), the magnetic and electric vector potentials respectively are defined as follow [17, 29]

$$\boldsymbol{A}(\boldsymbol{r}) = \frac{\mu}{4\pi}\int_V \boldsymbol{J}(\boldsymbol{r}')\frac{e^{-jk|\boldsymbol{r}-\boldsymbol{r}'|}}{|\boldsymbol{r}-\boldsymbol{r}'|}dV', \tag{2.29}$$

$$\boldsymbol{F}(\boldsymbol{r}) = \frac{\varepsilon}{4\pi}\int_V \boldsymbol{M}(\boldsymbol{r}')\frac{e^{-jk|\boldsymbol{r}-\boldsymbol{r}'|}}{|\boldsymbol{r}-\boldsymbol{r}'|}dV'. \tag{2.30}$$

Equations (2.29) and (2.30) can be re-written by introducing

$$G_0(\boldsymbol{r},\boldsymbol{r}') = \frac{e^{-jk|\boldsymbol{r}-\boldsymbol{r}'|}}{4\pi|\boldsymbol{r}-\boldsymbol{r}'|}, \tag{2.31}$$

where $G_0(\boldsymbol{r},\boldsymbol{r}')$ is known as the 3-D free-space scalar Green's function [17]. Substituting equation (2.31) into equations (2.29) and (2.30) yield

$$\boldsymbol{A}(\boldsymbol{r}) = \frac{\mu}{4\pi}\int_V G_0(\boldsymbol{r},\boldsymbol{r}')\boldsymbol{J}(\boldsymbol{r}')\,dV', \tag{2.32}$$

$$\boldsymbol{F}(\boldsymbol{r}) = \frac{\varepsilon}{4\pi}\int_V G_0(\boldsymbol{r},\boldsymbol{r}')\boldsymbol{M}(\boldsymbol{r}')\,dV'. \tag{2.33}$$

Therefore, plugging equations (2.32) and (2.33) into equation (2.28) produces

$$-j\omega\mu \int_V \left(G_0(\boldsymbol{r},\boldsymbol{r}')\boldsymbol{J}\left(\boldsymbol{r}'\right) + \frac{1}{k^2}\boldsymbol{\nabla}\boldsymbol{\nabla}G_0(\boldsymbol{r},\boldsymbol{r}')\boldsymbol{J}\left(\boldsymbol{r}'\right) \right) dV' - \int_V \boldsymbol{\nabla}G_0(\boldsymbol{r},\boldsymbol{r}')\boldsymbol{M}\left(\boldsymbol{r}'\right) dV' = \boldsymbol{E}\left(\boldsymbol{r}\right). \quad (2.34)$$

Introducing a new mathematical quantity in the form of identity matrix $\overline{\overline{I}} = \hat{\mathbf{x}}\hat{\mathbf{x}} + \hat{\mathbf{y}}\hat{\mathbf{y}} + \hat{\mathbf{z}}\hat{\mathbf{z}}$, then equation (2.34) can be re-written as

$$\boldsymbol{E}\left(\boldsymbol{r}\right) = -j\omega\mu \int_V \overline{\overline{G}}_{e0}(\boldsymbol{r},\boldsymbol{r}') \cdot \boldsymbol{J}(\boldsymbol{r}')dV' - \int_V \overline{\overline{G}}_{m0}(\boldsymbol{r},\boldsymbol{r}') \cdot \boldsymbol{M}\left(\boldsymbol{r}'\right) dV', \quad (2.35)$$

where

$$\overline{\overline{G}}_{e0}(\boldsymbol{r},\boldsymbol{r}') = \left(\frac{\boldsymbol{\nabla}\boldsymbol{\nabla}}{k^2} + \overline{\overline{I}} \right) G_0(\boldsymbol{r},\boldsymbol{r}'), \quad (2.36)$$

$$\overline{\overline{G}}_{m0}(\boldsymbol{r},\boldsymbol{r}') = \boldsymbol{\nabla}G_0(\boldsymbol{r},\boldsymbol{r}') \times \overline{\overline{I}}, \quad (2.37)$$

$\overline{\overline{G}}_{e0}(\boldsymbol{r},\boldsymbol{r}')$ and $\overline{\overline{G}}_{m0}(\boldsymbol{r},\boldsymbol{r}')$ are the dyadic Green's functions of the electric and magnetic types respectively.

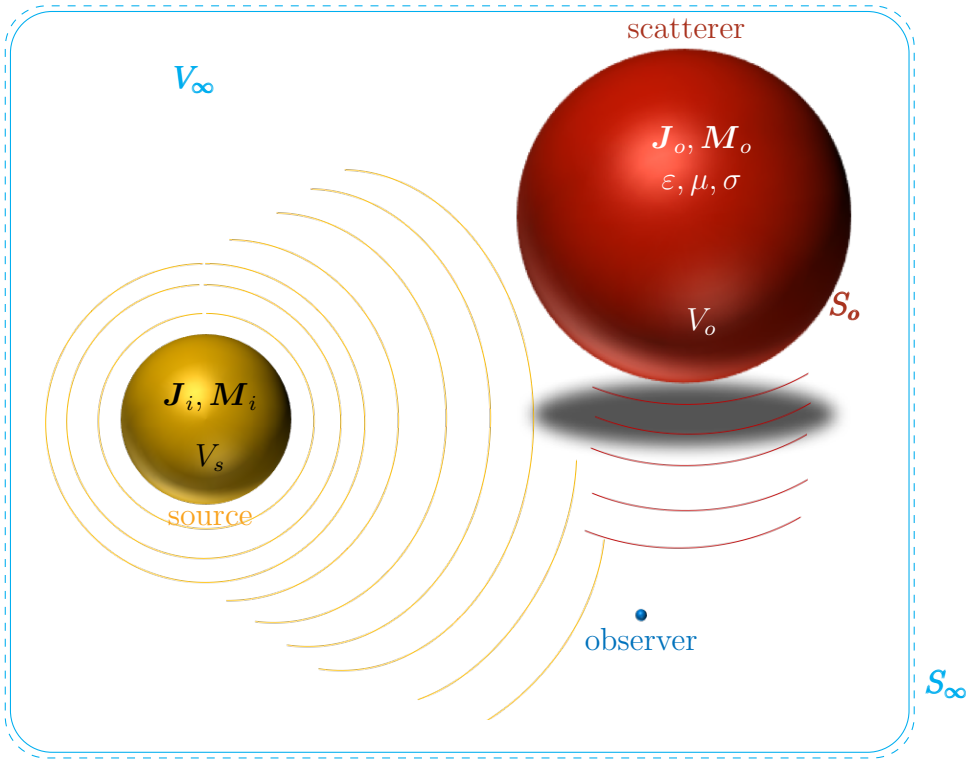


Figure 2.1: Scattering problem showing incident field generated by the source impinging on the surface of the scatterer S_o, the scatterer in turn produces a scatter field such that the superposition of the two fields can be observed by the observer .

2.4 Volume Equivalence Principle

Consider a time-harmonic electromagnetic field, with sources $\boldsymbol{J}_i$ and $\boldsymbol{M}_i$, impinging on a well-conducting penetrable scatterer (Figure (2.1)) in time-harmonic form with the cyclic frequency ω and time convention $e^{j\omega \mathtt{t}}$. If V is the volume of the scatterer and $\mathtt{t}$ is the time variable, then $e^{j\omega \mathtt{t}}$ is assumed and $j = \sqrt{-1}$. The volumetric equivalence principle [29] can be written, for the total electric field $\boldsymbol{E}$ at an arbitrary observation point $\boldsymbol{r}$ in space and position vector $\boldsymbol{r}'$ inside the scatterer. The radiated

field which can be observed in the vicinity of the scatterer must satisfy

$$\boldsymbol{\nabla} \times \boldsymbol{E}(\boldsymbol{r}) = -j\omega\mu(\boldsymbol{r})\boldsymbol{H}(\boldsymbol{r}) - \boldsymbol{M}_i, \tag{2.38}$$

$$\boldsymbol{\nabla} \times \boldsymbol{H}(\boldsymbol{r}) = j\omega\varepsilon\boldsymbol{E}(\boldsymbol{r}) + \boldsymbol{J}_i. \tag{2.39}$$

Obtaining the direct solution of (2.38) requires a lot of effort, however, the equivalent sources which exist only in the volume of the scatterer can be used to replace the scatterer (Figure (2.1)) [29] as

$$\boldsymbol{\nabla} \times \boldsymbol{E}(\boldsymbol{r}) = -j\omega\mu(\boldsymbol{r})\boldsymbol{H}(\boldsymbol{r}) - \boldsymbol{M}_i - \boldsymbol{M}_o, \tag{2.40}$$

$$\boldsymbol{\nabla} \times \boldsymbol{H}(\boldsymbol{r}) = j\omega\varepsilon(\boldsymbol{r})\boldsymbol{E}(\boldsymbol{r}) + \boldsymbol{J}_i + \boldsymbol{J}_o, \tag{2.41}$$

where $\boldsymbol{J}_o$ and $\boldsymbol{M}_o$ are the equivalent electric and magnetic sources respectively. The equivalent sources as used in equations (2.40) and (2.41) are expressed as

$$\boldsymbol{M}_o = j\omega\mu_0\left(\mu_r - 1\right)\boldsymbol{H}, \tag{2.42}$$

$$\boldsymbol{J}_o = j\omega\varepsilon_0\left(\varepsilon_r - 1\right)\boldsymbol{E}, \tag{2.43}$$

where ε_0 is the permittivity of the free-space, ε_r is the relative permittivity of the scatterer's media, μ_0 is the free-space permeability, and μ_r is the relative permeability of the media.

From the field-source relationship detailed in section (2.3), equation (2.40) can be

expressed in terms of equation (2.35) thus

$$\begin{aligned}\boldsymbol{E}(\boldsymbol{r}) = &- j\omega\mu \int_{V_s} \overline{\overline{G}}_{e0}(\boldsymbol{r},\boldsymbol{r}') \cdot \boldsymbol{J}_i(\boldsymbol{r}')dV' - \int_{V_s} \overline{\overline{G}}_{m0}(\boldsymbol{r},\boldsymbol{r}') \cdot \boldsymbol{M}_i(\boldsymbol{r}')dV' \\ &- j\omega\mu \int_{V_o} \overline{\overline{G}}_{e0}(\boldsymbol{r},\boldsymbol{r}') \cdot \boldsymbol{J}_o(\boldsymbol{r}')dV' - \int_{V_o} \overline{\overline{G}}_{m0}(\boldsymbol{r},\boldsymbol{r}') \cdot \boldsymbol{M}_o(\boldsymbol{r}')dV',\end{aligned} \tag{2.44}$$

where V_s and V_o denote the volumes of the source and the object respectively (as shown in Figure (2.1)). The first-two terms with integral over V_s denote the incident field [17], while the remaining two integrals represent the scattered field. Therefore, equation (2.44) can be re-written in a more concise form as

$$\boldsymbol{E}(\boldsymbol{r}) = \boldsymbol{E}^{\text{inc}}(\boldsymbol{r}) - j\omega\mu \int_{V_o} \overline{\overline{G}}_{e0}(\boldsymbol{r},\boldsymbol{r}') \cdot \boldsymbol{J}_o(\boldsymbol{r}')dV' - \int_{V_o} \overline{\overline{G}}_{m0}(\boldsymbol{r},\boldsymbol{r}') \cdot \boldsymbol{M}_o(\boldsymbol{r}')dV', \tag{2.45}$$

where $\boldsymbol{E}^{\text{inc}}$ denotes the incident electric field.

Substituting equations (2.42) and (2.43) into equation (2.45) gives

$$\begin{aligned}\boldsymbol{E}(\boldsymbol{r}) = \boldsymbol{E}^{\text{inc}}(\boldsymbol{r}) &+ \omega^2\mu_0\varepsilon_0\mu_r(\varepsilon_r - 1) \int_{V_o} \overline{\overline{G}}_{e0}(\boldsymbol{r},\boldsymbol{r}') \cdot \boldsymbol{E}(\boldsymbol{r}')dV' \\ &- j\omega\mu_0(\mu_r - 1) \int_{V_o} \overline{\overline{G}}_{m0}(\boldsymbol{r},\boldsymbol{r}') \cdot \boldsymbol{H}(\boldsymbol{r}')dV',\end{aligned} \tag{2.46}$$

while putting $k = \omega\sqrt{\mu_0\varepsilon_0}$ in equation (2.46) results in

$$\boldsymbol{E}(\boldsymbol{r}) = \boldsymbol{E}^{\text{inc}}(\boldsymbol{r}) + k^2\mu_r(\varepsilon_r - 1)\int_{V_o} \overline{\overline{G}}_{e0}(\boldsymbol{r},\boldsymbol{r}') \cdot \boldsymbol{E}(\boldsymbol{r}')dV' - j\omega\mu_0(\mu_r - 1)\int_{V_o} \overline{\overline{G}}_{m0}(\boldsymbol{r},\boldsymbol{r}') \cdot \boldsymbol{H}(\boldsymbol{r}')dV'. \quad (2.47)$$

2.5 3-D Surface-Volume-Surface EFIE Formulation

Consider the scattering problem in Figure (2.1) for a homogeneous and non-magnetic well-conducting penetrable metal object. The volume equivalence principle can be obtained for the problem from equation (2.47) thus:

$$\boldsymbol{E}(\boldsymbol{r}) = \boldsymbol{E}^{\text{inc}}(\boldsymbol{r}) + k^2(\epsilon_r - 1)\int_{V_o} \overline{\overline{G}}_{e0}(\boldsymbol{r},\boldsymbol{r}') \cdot \boldsymbol{E}(\boldsymbol{r}')dV', \quad (2.48)$$

where $\mu_r = 1$, $\boldsymbol{r}$ is an arbitrary observation point in space, $\boldsymbol{r}'$ denotes the position-vector inside the scatterer, and ϵ_r is the complex relative permittivity of the media defined as

$$\epsilon_r = \varepsilon_r + \frac{\sigma}{j\omega\varepsilon_0}, \quad (2.49)$$

where σ represents the conductivity of the media. The field inside the object satisfies homogeneous curl-curl Helmholtz equation

$$\boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \boldsymbol{E}(\boldsymbol{r}') - k_\epsilon^2 \boldsymbol{E}(\boldsymbol{r}') = \boldsymbol{0}, \boldsymbol{r}' \in V \setminus \partial V, \quad (2.50)$$

where $k_\epsilon = \sqrt{\epsilon}_r k_0$ is the wave number of the media.

To arrive at the SVS-EFIE formulation, the total electric field inside the scatterer is expressed as the superposition of the primary waves originating from the scatterer's boundary [20, 23–25] as

$$-j\omega\mu_0 \oint_{\partial V} \overline{\overline{G}}_{e\epsilon}(\boldsymbol{r}', \boldsymbol{r}'') \cdot \boldsymbol{J}(\boldsymbol{r}'')\, dS'' = \boldsymbol{E}(\boldsymbol{r}') \tag{2.51}$$

where the tangential current density $\boldsymbol{J}(\boldsymbol{r}'')$ is the weighting for the waves $\overline{\overline{G}}_{e\epsilon}(\boldsymbol{r}', \boldsymbol{r}'')$ and it is defined on surface ∂V of the object. The electric type dyadic Green's function of the media, $\overline{\overline{G}}_{e\epsilon}(\boldsymbol{r}', \boldsymbol{r}'')$ is define as follows

$$\overline{\overline{G}}_{e\epsilon}(\boldsymbol{r}', \boldsymbol{r}'') = \left(\frac{\boldsymbol{\nabla}'\boldsymbol{\nabla}'}{k_\epsilon^2} + \overline{\overline{I}}\right) G_\epsilon(\boldsymbol{r}', \boldsymbol{r}'') \tag{2.52}$$

$$\text{where,} \qquad G_\epsilon(\boldsymbol{r}', \boldsymbol{r}'') = \frac{e^{-jk_\epsilon|\boldsymbol{r}'-\boldsymbol{r}''|}}{4\pi\,|\boldsymbol{r}' - \boldsymbol{r}''|}. \tag{2.53}$$

The waves $\overline{\overline{G}}_{e\epsilon}$ satisfies the homogeneous equation

$$\boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \overline{\overline{G}}_{e\epsilon} - k_\epsilon^2 \overline{\overline{G}}_\epsilon = \overline{\overline{0}}. \tag{2.54}$$

Finally, substituting equations (2.36), (2.51) and, (2.52) into the (2.48) and localizing the observation domain to the surface of the object produces the SVS-EFIE equation

[20, 34]

$$
\begin{aligned}
&- j\omega\mu_0 \hat{\mathbf{t}} \cdot \oint_{\partial V} \overline{\overline{G}}_{e\epsilon}(\boldsymbol{r}, \boldsymbol{r}'') \cdot \boldsymbol{J}(\boldsymbol{r}'') dS'' \\
&\qquad + j\omega\mu_0 k^2 \left(\epsilon_r - 1\right) \hat{\mathbf{t}} \cdot \int_V \overline{\overline{G}}_{e0}(\boldsymbol{r}, \boldsymbol{r}') \cdot \oint_{\partial V} \overline{\overline{G}}_{e\epsilon}(\boldsymbol{r}', \boldsymbol{r}'') \cdot \boldsymbol{J}(\boldsymbol{r}'') dS'' dV' \\
&\qquad\qquad = \hat{\mathbf{t}} \cdot \boldsymbol{E}^{\text{inc}}(\boldsymbol{r}), \quad \boldsymbol{r} \in \partial V. \quad (2.55)
\end{aligned}
$$

where the tangential vector to the boundary ∂V is denoted by $\hat{\boldsymbol{t}}$. However, since the observation points are limited to the boundary of the scatterer, the primary waves $\overline{\overline{G}}_{e\epsilon}$ is weighted by the auxiliary tangential current density $\boldsymbol{J}$. $\overline{\overline{G}}_{e\epsilon}(\boldsymbol{r}, \boldsymbol{r}'')$ is expressed in-terms of 3-D scalar Green's function as

$$
\overline{\overline{G}}_{e\epsilon}(\boldsymbol{r}, \boldsymbol{r}'') = \left(\frac{\boldsymbol{\nabla}\boldsymbol{\nabla}}{k_\epsilon^2} + \overline{\overline{I}} \right) G_\epsilon(\boldsymbol{r}, \boldsymbol{r}'') \quad (2.56)
$$

$$
\text{where,} \qquad G_\epsilon(\boldsymbol{r}, \boldsymbol{r}'') = \frac{e^{-jk_\epsilon |\boldsymbol{r} - \boldsymbol{r}''|}}{4\pi \left| \boldsymbol{r} - \boldsymbol{r}'' \right|}. \quad (2.57)
$$

2.6 Operator Form of SVS-EFIE Equation

Dealing with the equation (2.55) directly is error prone because of the length of the equation. Therefore, for the sake of convenience, the equation can be expressed in operator forms [20, 34].

Equation (2.55) can be written with note thus:

$$\underbrace{-j\omega\mu_0\hat{\mathbf{t}}\cdot\oint_{\partial V}\overline{\overline{G}}_{e\epsilon}(\boldsymbol{r},\boldsymbol{r}'')\cdot\boldsymbol{J}(\boldsymbol{r}'')dS''}_{\text{Part I}}$$
$$\underbrace{+\,j\omega\mu_0 k^2\left(\epsilon_r-1\right)\hat{\mathbf{t}}\cdot\int_V\overline{\overline{G}}_{e0}(\boldsymbol{r},\boldsymbol{r}')\cdot\oint_{\partial V}\overline{\overline{G}}_{e\epsilon}(\boldsymbol{r}',\boldsymbol{r}'')\cdot\boldsymbol{J}(\boldsymbol{r}'')dS''dV'}_{\text{Part II}}$$
$$=\hat{\mathbf{t}}\cdot\boldsymbol{E}^{\text{inc}}(\boldsymbol{r}),\quad \boldsymbol{r}\in\partial V. \tag{2.58}$$

2.6.1 Surface-to-Surface Operator

Consider the part labelled "Part I" in equation (2.58)

$$j\omega\mu_0\hat{\mathbf{t}}\cdot\oint_{\partial V}\overline{\overline{G}}_{e\epsilon}(\boldsymbol{r},\boldsymbol{r}'')\cdot\boldsymbol{J}(\boldsymbol{r}'')dS'', \tag{2.59}$$

substituting equation (2.56) into equation (2.59) gives

$$j\omega\mu_0\hat{\mathbf{t}}\cdot\oint_{\partial V}\left[\left(\frac{\boldsymbol{\nabla}\boldsymbol{\nabla}}{k_\epsilon^2}+\overline{\overline{I}}\right)\boldsymbol{G}_\epsilon(\boldsymbol{r},\boldsymbol{r}'')\right]\cdot\boldsymbol{J}(\boldsymbol{r}'')dS'' \tag{2.60}$$

this implies

$$j\omega\mu_0\hat{\mathbf{t}}\cdot\oint_{\partial V}\frac{\boldsymbol{\nabla}\boldsymbol{\nabla}}{k_\epsilon^2}\boldsymbol{G}_\epsilon(\boldsymbol{r},\boldsymbol{r}'')\cdot\boldsymbol{J}(\boldsymbol{r}'')dS''+j\omega\mu_0\hat{\mathbf{t}}\cdot\oint_{\partial V}\boldsymbol{G}_\epsilon(\boldsymbol{r},\boldsymbol{r}'')\boldsymbol{J}(\boldsymbol{r}'')dS''. \tag{2.61}$$

By using vector identity

$$\boldsymbol{A}(\boldsymbol{r}'')\cdot\boldsymbol{\nabla}\phi(\boldsymbol{r})=\boldsymbol{\nabla}\cdot\boldsymbol{A}(\boldsymbol{r}'')\phi(\boldsymbol{r})-\left(\boldsymbol{\nabla}\cdot\boldsymbol{A}(\boldsymbol{r}'')\right)\phi(\boldsymbol{r}), \tag{2.62}$$

and the applying surface divergence theorem

$$\oint_{\partial V}\phi(\boldsymbol{r})\boldsymbol{J}(\boldsymbol{r}'')\cdot\hat{\mathbf{u}}dl=0, \tag{2.63}$$

equation (2.61) becomes

$$\hat{\mathbf{t}}\cdot\underbrace{\underbrace{-\boldsymbol{\nabla}\frac{j\omega\mu_0}{k_\epsilon^2}\oint_{\partial V}\boldsymbol{G}_\epsilon(\boldsymbol{r},\boldsymbol{r}'')\boldsymbol{\nabla}_{\mathrm{s}}''\cdot\boldsymbol{J}(\boldsymbol{r}'')dS''}_{\boldsymbol{\nabla}\mathcal{T}_{\epsilon,\Phi}^{\partial V,\partial V}}+\underbrace{j\omega\mu_0\hat{\mathbf{t}}\cdot\oint_{\partial V}\boldsymbol{G}_\epsilon(\boldsymbol{r},\boldsymbol{r}'')\boldsymbol{J}(\boldsymbol{r}'')dS''}_{\boldsymbol{\mathcal{T}}_{\epsilon,A}^{\partial V,\partial V}}}_{\mathcal{T}_\epsilon^{\partial V,\partial V}}, \tag{2.64}$$

where $\mathcal{T}_{\epsilon,\Phi}^{\partial V,\partial V}$ and $\boldsymbol{\mathcal{T}}_{\epsilon,A}^{\partial V,\partial V}$ are known as the scalar potential operator and vector potential operator respectively as a breakdown of the Surface-to-Surface operator $\mathcal{T}_\epsilon^{\partial V,\partial V}$ and surface divergence is given by $\boldsymbol{\nabla}_{\mathrm{s}}\cdot$.

2.6.2 Volume-to-Surface Operator

Consider the sub-equation labelled "Part II" in equation (2.58)

$$j\omega\mu_0k^2\left(\epsilon_r-1\right)\hat{\mathbf{t}}\cdot\int_V\overline{\overline{G}}_{e0}(\boldsymbol{r},\boldsymbol{r}')\cdot\oint_{\partial V}\overline{\overline{G}}_{e\epsilon}(\boldsymbol{r}',\boldsymbol{r}'')\cdot\boldsymbol{J}(\boldsymbol{r}'')dS''dV'. \tag{2.65}$$

Inserting equation (2.51) into equation (2.65) yields

$$k^2\left(\epsilon_r-1\right)\hat{\mathbf{t}}\cdot\int_V\overline{\overline{G}}_{e0}(\boldsymbol{r},\boldsymbol{r}')\cdot\boldsymbol{E}(\boldsymbol{r}')dV', \tag{2.66}$$

substituting $\boldsymbol{J}_k(\boldsymbol{r}')=k^2\left(\epsilon_r-1\right)\boldsymbol{E}(\boldsymbol{r}')$ into equation (2.66) gives

$$\hat{\mathbf{t}}\cdot\int_V\overline{\overline{G}}_{e0}(\boldsymbol{r},\boldsymbol{r}')\cdot\boldsymbol{J}_k(\boldsymbol{r}')dV'. \tag{2.67}$$

Then, plugging equation (2.36) into equation (2.67) results in

$$\underbrace{\hat{\mathbf{t}}\cdot\underbrace{\frac{\boldsymbol{\nabla}}{k^2}\int_V\boldsymbol{\nabla}\boldsymbol{G}_0(\boldsymbol{r},\boldsymbol{r}')\cdot\boldsymbol{J}_k(\boldsymbol{r}')dV'}_{\boldsymbol{\nabla}\mathcal{T}_{0,\varphi}^{\partial V,V}}+\hat{\mathbf{t}}\cdot\underbrace{\int_V\boldsymbol{G}_0(\boldsymbol{r},\boldsymbol{r}')\boldsymbol{J}_k(\boldsymbol{r}')dV'}_{\boldsymbol{\mathcal{T}}_{0,a}^{\partial V,V}}}_{\mathcal{T}_0^{\partial V,V}}, \tag{2.68}$$

where $\mathcal{T}_{0,\nabla\varphi}^{\partial V,V}$ and $\boldsymbol{\mathcal{T}}_{0,a}^{\partial V,V}$ are the scalar potential operator and vector potential operator respectively as sub-divisions of Volume-to-Surface operator $\mathcal{T}_0^{\partial V,V}$.

2.6.3 Surface-to-Volume Operator

Consider the portion of equation (2.51) substituted into equation (2.65) in the preceding section (2.6.2)

$$j\omega\mu_0\oint_{\partial V}\overline{\overline{G}}_{e\epsilon}(\boldsymbol{r}',\boldsymbol{r}'')\ \cdot\ \boldsymbol{J}(\boldsymbol{r}'')\ dS''. \tag{2.69}$$

Inserting equation (2.52) into equation (2.69) and shifting the first gradient operator to the unknown current density $\boldsymbol{J}(\boldsymbol{r}'')$ as it is done in section (2.6.1) yield

$$\underbrace{\underbrace{\boldsymbol{\nabla}'\frac{j\omega\mu_0}{k_\epsilon^2}\oint_{\partial V} G_\epsilon(\boldsymbol{r}',\boldsymbol{r}'')\boldsymbol{\nabla}''_{\mathrm{s}}\cdot\boldsymbol{J}(\boldsymbol{r}'')dS''}_{\boldsymbol{\nabla}\mathcal{T}^{V,\partial V}_{\epsilon,\Phi}} + \underbrace{j\omega\mu_0\oint_{\partial V} G_\epsilon(\boldsymbol{r}',\boldsymbol{r}'')\boldsymbol{J}(\boldsymbol{r}'')dS''}_{\mathcal{T}^{V,\partial V}_{\epsilon,A}}}_{\mathcal{T}^{V,\partial V}_{\epsilon}}, \tag{2.70}$$

where $\mathcal{T}^{V,\partial V}_{\epsilon,\Phi}$ and $\mathcal{T}^{V,\partial V}_{\epsilon,A}$ are the scalar potential operator and vector potential operator of the Surface-to-Volume operator $\mathcal{T}^{V,\partial V}_{\epsilon}$.

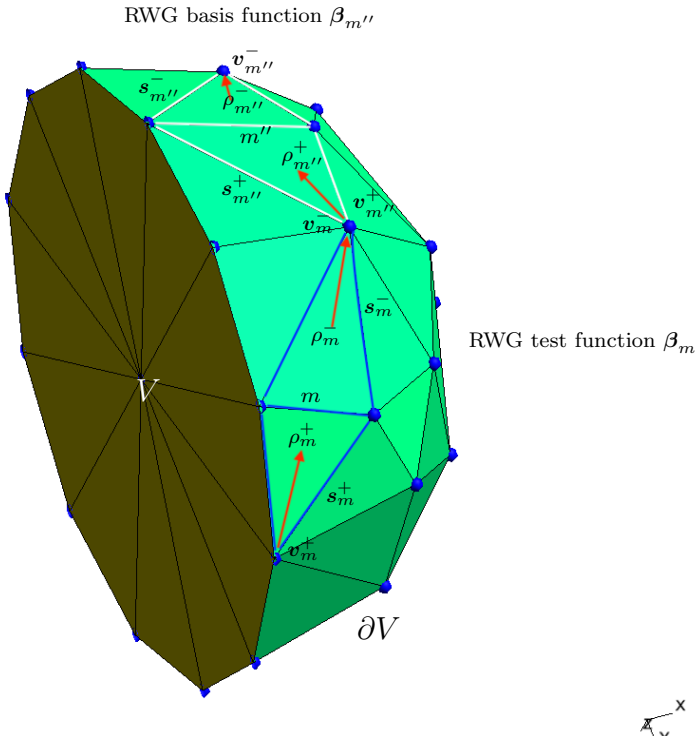


Figure 2.2: Tetrahedral volume mesh and triangle surface mesh utilized in the MoM solution of the SVS-EFIE (2.55) and RWG basis functions on ∂V obtained from Gmsh software [10].

2.7 Discretization of SVS-EFIE Operators

In order to solve the SVS-EFIE with MoM, the volume of the scatterer V is discretized with 3-D mesh consisting of N tetrahedral elements while the boundary of the scatterer ∂V is discretized with 2-D mesh consisting of M triangle elements. The RWG [31] basis functions are used on the surface triangle elements and pulse basis functions are used at the centroids of tetrahedral volume elements [20, 34]. The meshing obtained from Gmsh software [10] is shown in Figure (2.2).

Consider the pair of triangles $\boldsymbol{s}^-_{m''}$ and $\boldsymbol{s}^+_{m''}$ in Figure (2.2), RWG basis function $\boldsymbol{\beta}_{m''}$ defined on the common edge between these triangle elements is given by

$$\boldsymbol{\beta}_{m''}(\boldsymbol{r}'') = \begin{cases} \boldsymbol{r}'' \in s^-_{m''} & : \quad \dfrac{\boldsymbol{\rho}^-_{m''}}{2A_{s^-_{m''}}}\ell_{m''}, \\ \boldsymbol{r}'' \in s^+_{m''} & : \quad \dfrac{\boldsymbol{\rho}^+_{m''}}{2A_{s^+_{m''}}}\ell_{m''}, \\ \text{Otherwise} & : \quad 0 \end{cases} \tag{2.71}$$

where $\boldsymbol{\rho}^-_{m''} = \boldsymbol{v}^-_{m''} - \boldsymbol{r}''$, $\boldsymbol{v}^-_{m''}$ being the non-common vertex on triangle $s^-_{m''}$, $\boldsymbol{\rho}^+_{m''} = \boldsymbol{r}'' - \boldsymbol{v}^+_{m''}$, $\boldsymbol{v}^+_{m''}$ is the non-common vertex on triangle $s^+_{m''}$, $\ell_{m''}$ is the length of the common edge m'', $A_{s^-_{m''}}$ and $A_{s^+_{m''}}$ represent the areas of triangles $s^-_{m''}$ and $s^+_{m''}$ respectively.

Next, expand the unknown current density $\boldsymbol{J}$ in-terms of the RWG basis function expressed in equation (2.71). The unknown $\boldsymbol{J}$ can be expanded thus

$$\boldsymbol{J}(\boldsymbol{r}'') \approx \sum_{m''=1}^{P} I_{m''}\boldsymbol{\beta}_{m''}(\boldsymbol{r}''), \tag{2.72}$$

where m'' denotes a given common edge and ranges from 1 to P, P is the total number of common edges - RWG basis functions, and $I_{m''}$ is the weight of a given RWG basis function m''.

The presence of surface divergence operators in the scalar potential part of Surface-to-Surface operator (2.64) and Surface-to-Volume operator (2.70) demand for the definition of surface divergence whilst using RWG basis function. Therefore, surface divergence is defined as

$$\boldsymbol{\nabla}''_{\mathrm{s}} \cdot \boldsymbol{\beta}_{m''}(\boldsymbol{r}'') = \begin{cases} \boldsymbol{r}'' \in s^{-}_{m''} & : \quad -\dfrac{\ell_{m''}}{A_{s^{-}_{m''}}}, \\ \boldsymbol{r}'' \in s^{+}_{m''} & : \quad \dfrac{\ell_{m''}}{A_{s^{+}_{m''}}} \\ \text{Otherwise} & : \quad 0. \end{cases} \tag{2.73}$$

The pulse basis function defined at the centroids of the tetrahedral volume elements is given by

$$p_{n'}(\boldsymbol{r}') = \begin{cases} \boldsymbol{r}' \in V_{n'} & : \quad 1, \\ \boldsymbol{r}' \notin V_{n'} & : \quad 0, \end{cases} \tag{2.74}$$

where $V_{n'}$ is the volume of the $n'th$ tetrahedron, $n' = 1, \ldots, N$ and N is the total number of tetrahedral elements.

Furthermore, the SVS-EFIE equation is tested using Galerkin's method. Then, MoM matrix element corresponding to Surface-to-Surface operator $\mathcal{T}_{\epsilon}^{\partial V, \partial V}$ is $Z_{\epsilon}^{\partial V, \partial V}$ matrix. Hence, testing the scalar potential operator $\mathcal{T}_{\epsilon,\Phi}^{\partial V, \partial V}$ and vector potential

operator $\boldsymbol{\mathcal{T}}_{\epsilon,A}^{\partial V,\partial V}$ of the Surface-to-Surface operator with weighting function $\boldsymbol{\beta}_m$ gives

$$\boldsymbol{Z}_{\epsilon,\nabla\Phi\, mm''}^{\partial V,\partial V} = \left\langle \boldsymbol{\beta}_m, \boldsymbol{\nabla}\mathcal{T}_{\epsilon,\Phi}^{\partial V,\partial V} \circ \boldsymbol{\beta}_{m''} \right\rangle, \tag{2.75}$$

$$\boldsymbol{Z}_{\epsilon,A\, mm''}^{\partial V,\partial V} = \left\langle \boldsymbol{\beta}_m, \boldsymbol{\mathcal{T}}_{\epsilon,A}^{\partial V,\partial V} \circ \boldsymbol{\beta}_{m''} \right\rangle, \tag{2.76}$$

where $m'' = 1,\ldots,P$ and $m = 1,\ldots,P$. Consequently, testing the scalar potential and vector potential operators of Volume-to-Surface operator $\mathcal{T}_0^{\partial V,V}$ with both the RWG basis function $\boldsymbol{\beta}_m$ and pulse basis function $p_{n'}$ result in matrix elements corresponding to $Z_0^{\partial V,V}$ thus

$$\boldsymbol{Z}_{0,\nabla\varphi\, mn'}^{\partial V,V} = \left\langle \boldsymbol{\beta}_m, \boldsymbol{\nabla}\mathcal{T}_{0,\varphi}^{\partial V,V} \circ p_{n'} \right\rangle, \tag{2.77}$$

$$\boldsymbol{Z}_{0,a\, mn'}^{\partial V,V} = \left\langle \boldsymbol{\beta}_m, \boldsymbol{\mathcal{T}}_{0,a}^{\partial V,V} \circ p_{n'} \right\rangle, \tag{2.78}$$

where $m = 1,\ldots,P$ and $n' = 1,\ldots,N$. Similarly, to obtain the MoM matrix element $Z_\epsilon^{V,\partial V}$ corresponding to Surface-to-Volume operator $\mathcal{T}_\epsilon^{V,\partial V}$, the scalar potential and vector potential operators are weighted with both the pulse basis function $p_{n'}$ and RWG basis function $\boldsymbol{\beta}_m$ as follows

$$\boldsymbol{Z}_{\epsilon,\nabla\Phi\, n'm''}^{V,\partial V} = \left\langle p_{n'}, \boldsymbol{\nabla}\mathcal{T}_{\epsilon,\Phi}^{V,\partial V} \circ \boldsymbol{\beta}_{m''} \right\rangle, \tag{2.79}$$

$$\boldsymbol{Z}_{\epsilon,A\, n'm''}^{V,\partial V} = \langle p_{n'}, \mathcal{T}_{\epsilon,A}^{V,\partial V} \circ \boldsymbol{\beta}_{m''} \rangle, \tag{2.80}$$

where $m'' = 1,\ldots,P$ and $n' = 1,\ldots,N$. Likewise, the right hand side of the SVS-

EFIE equation is tested with RWG basis function $\boldsymbol{\beta}_m$ as

$$\mathcal{V}_m = \left\langle \boldsymbol{\beta}_m, \boldsymbol{E}^{\text{inc}} \right\rangle, \tag{2.81}$$

where $m = 1, \ldots, P$.

Finally, assembling the components of MoM matrix Z produces

$$\begin{aligned} &- \left(\left[Z_{\epsilon,A}^{\partial V,\partial V} \right] + \left[Z_{\epsilon,\nabla\Phi}^{\partial V,\partial V} \right] \right) \cdot [I] \\ &+ \sum_{c\,=1}^{3} \left(\left(\left[Z_{0,a}^{\partial V,V} \right]_c + \left[Z_{0,\nabla\varphi}^{\partial V,V} \right]_c \right) \cdot \Gamma^{-1} \cdot \left(\left[Z_{\epsilon,A}^{V,\partial V} \right]_c + \left[Z_{\epsilon,\nabla\Phi}^{V,\partial V} \right]_c \right) \right) \cdot [I] = [\mathcal{V}], \end{aligned} \tag{2.82}$$

where $c = 1, \ldots 3$. The counter 1, 2 and 3 correspond to x, y and z components respectively, and Γ is the Gramian matrix defined in [20, 34].

From the foregoing, it can be seen that the SVS-EFIE equation has three different matrices namely, Surface-to-Surface matrix, Volume-to-Surface matrix, and Surface-to-Volume matrix. Surface-to-Volume matrix being $P \times P$ dimension, Volume-to-Surface is $P \times N$, and Surface-to-Volume is $N \times P$ - without considering the x, y and z components separately. However, with those components taken into consideration, the Volume-to-Surface matrix is of $P \times 3N$ dimension and Surface-to-Volume matrix is of $3N \times P$ dimension. Therefore, the resultant Z matrix is $P \times P$ dimension. This clearly shows that, both Surface-to-Volume and Volume-to-Surface matrices are rectangular matrices while the resultant matrix and Surface-to-Surface matrix are

square matrices.

The naïve implementation of SVS-EFIE equation can be found in [20, 34]. The computational cost and memory requirements needed for matrix operations and the storage of the resultant and intermediate MoM matrices make the naïve implementation of SVS-EFIE seemingly impractical for solving large scale scattering problems. To leverage this misnomer, $\mathcal{H}-$Matrices [5, 12–14] is often times employed to reduce the computational cost in matrix operations and the storage requirements. Therefore, $\mathcal{H}-$Matrices based MoM discretization of SVS-EFIE equation is presented in section (4.3) of Chapter (4). However, some of the features of $\mathcal{H}-$Matrices that would be referred to in Chapter (4) are discussed in Chapter (3).

Chapter 3

Overview of Hierarchical Matrices

3.1 Introduction

The main ideas behind $\mathcal{H}-$Matrices are the rank-based compression and data-sparse representation of a matrix in a tree-like manner where there exist two binary trees resulting in a quad-form structure. At the leaf node of each of these trees lies either a $full-rank$ matrix or a $low-rank$ matrix block known as the $rkmatrix$ or $\mathcal{R}$k-block.

A $full-rank$ or $fullmatrix$ block is a dense matrix which cannot be compressed and all of its entries are stored. In contrast, a $low-rank$ block is a matrix which can be compressed and stored in a factorized form as AB^T through compression via Adaptive Cross Approximation (ACA) [2–4, 37], Singular Value Decomposition (SVD) [3], Lanczos Algorithm [35] or their variants.

Hierarchical matrices as the name suggests are built on hierarchical tree structure called *block cluster tree*. A *block cluster tree* is constructed from *cluster tree*. A *cluster tree* is a product of hierarchical partitioning of an *index set* which represents sub-blocks.

Termination of hierarchical partitioning depends largely on the admissibility criterion. Both admissibility criterion and the leaf size (n_{min}) tell whether a block should be divided/partitioned further or should be a leaf node.

In order to use $\mathcal{H}-$Matrices, the $\mathcal{H}-$Matrix library (HLib) is used. Each of the features previously mentioned are already implemented in the library in C programming language. However, C being the precursor to C++, the library can be used in the same way in C++ programming language. The full reference on this topic can be found in [5, 12–14, 16].

3.2 Basic Features of $\mathcal{H}-$Matrices

Definition 3.2.1 (Index Set)**.** An index set is a set of finite indices/labels whose members index/label members of another set.

Remark. In this thesis, index set $\mathcal{I}$ represents a set of RWG basis functions and index set $\mathcal{J}$ denotes a set of centroids of tetrahedral elements.

$\mathcal{I} = \{0, \ldots, n-1\}$ and $\mathcal{J} = \{0, \ldots, m-1\}$, where $n = |\mathcal{I}|$ and $m = |\mathcal{J}|$.

Definition 3.2.2 (Tree $\mathcal{T}$)**.** Given a non-empty set of nodes V and edges E being a binary mapping on V, $\mathcal{T} = (V, E)$ is a tree if there is no duplicate reference, the unique node $v \in V$ is the root, and none points to the root.

Definition 3.2.3 (Cluster Tree $\mathcal{T}_{\mathcal{I}}$)**.** A tree $\mathcal{T}_{\mathcal{I}}$ is a cluster tree if it is defined over an index set $\mathcal{I}$ with root labelled $\mathcal{I}$ such that $t \in \mathcal{T}_{\mathcal{I}}$ has children and it is represented as the union of its sons' disjointed labels or indices. Thus,

$$\hat{t} = \bigcup_{t' \in sons(t)} \hat{t'},$$

and $\forall\ t_1, t_2, t_3 \in sons(t)$ then

$$t_1 \neq t_2, \qquad t_1 \neq t_3, \qquad t_2 \neq t_3,$$

$$t_1 \cap t_2 = \emptyset, \qquad t_1 \cap t_3 = \emptyset \qquad t_2 \cap t_3 = \emptyset.$$

Remark. There are other properties of a cluster tree. Given $t, s \in \mathcal{T}_{\mathcal{I}}$ at the same level of partitioning, $level(t) = level(s)$, then, $t \neq s$ and $\hat{t} \cap \hat{s} = \emptyset$ is known as the level-wise disjointness. Likewise, $\forall\ i \in \hat{t}\ \exists\ t \in \mathcal{T}_{\mathcal{I}}$ with $i \in \hat{t}$. Similarly, the indices of the leaf level clusters are disjoint partitions of $\mathcal{I}$.

For instance, consider an index set $\mathcal{I} = \{4, 10, 2, 8, 3, 6, 7, 5\}$, the cluster tree $\mathcal{T}_{\mathcal{I}}$ can be obtained by partitioning the index set with $n_{min} = 1$ as shown in Figure (3.1). Figure (3.1) shows a cluster tree $\mathcal{T}_{\mathcal{I}}$. The leaf nodes are coloured red and these are the

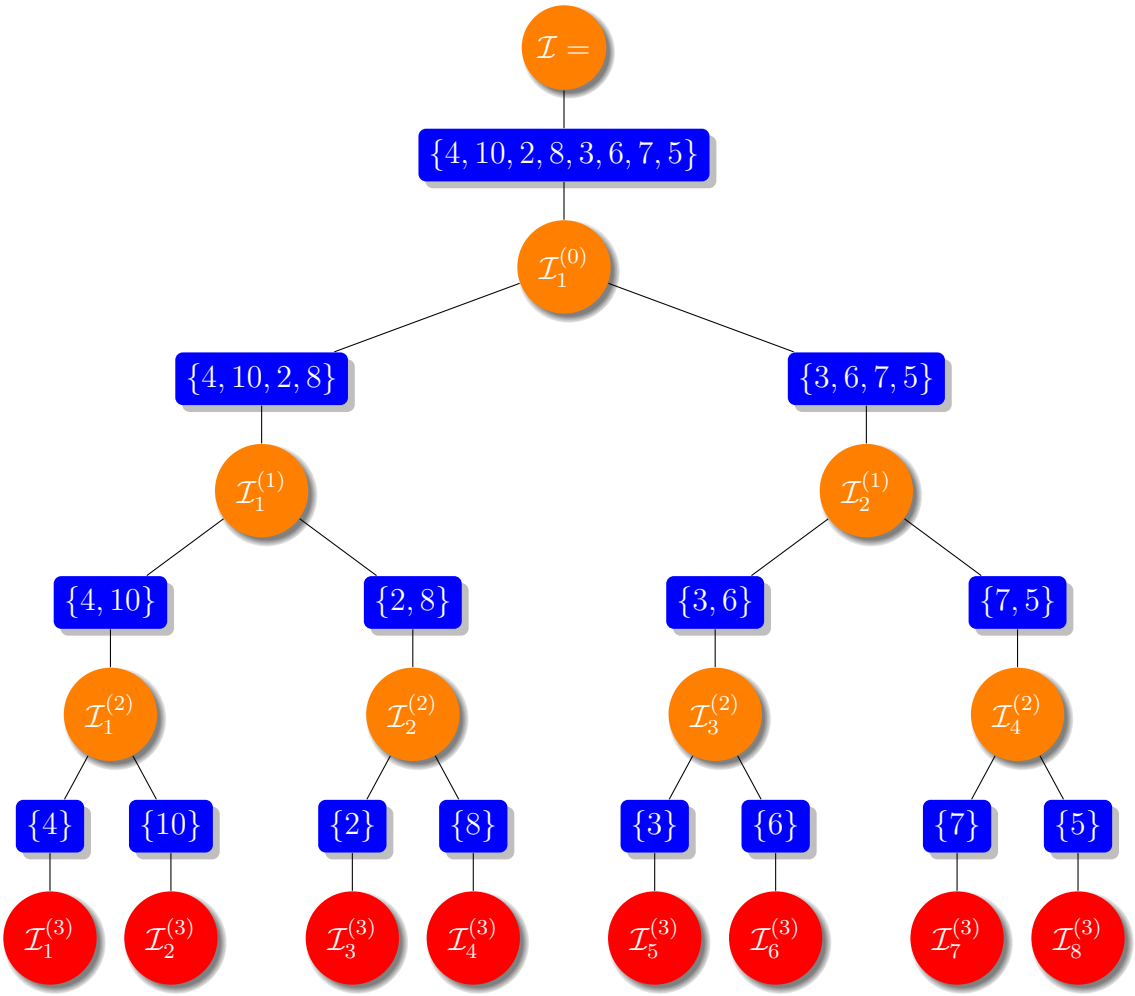


Figure 3.1: Cardinality balanced binary cluster tree $\mathcal{T}_{\mathcal{I}}$ of index set $\mathcal{I} = \{4, 10, 2, 8, 3, 6, 7, 5\}$ with leaf size, $n_{min} = 1$ and tree depth $p = 3$.

points at which the partitioning is terminated. The tree depth p is 3 and the leaf size n_{min} is set to 1. It is important to state that parameter n_{min} determines the tree depth p. The maximum possible tree depth is obtained by setting n_{min} to 1. Similarly, n_{min} should be in the powers of 2, such that $n_{min} = 1, 2, 4, 8, 16, 32, 64, \ldots$ [16]. Further partitioning stops normally when there exists exactly only one element.

The cluster tree in Figure (3.1) uses binary splitting and it is said to be cardinality

balanced binary tree [13, 16]. However, others forms of partitioning techniques are also possible [13, 16]. The signature of the routine for creating cluster tree in HLib is given by [13]

```
pclustertree
create_subclustertree(pclusterfactory factory,
      const int *index, int n, ClusterStrategy strategy, int leafsize);
```

Definition 3.2.4 (Admissibility Condition/Criterion)**.** Admissibility condition is used during the construction of block cluster trees to check whether a block needs to be further partitioned/divided into sub-blocks or not (that is, make it a leaf block). It also specifies if a block is to be represented as $\mathcal{R}k$-block approximation.

The admissibility condition is given by

$$min(diam(\tau), diam(\sigma)) \leq \eta dist(\tau, \sigma), \tag{3.1}$$

where sub-domains τ and σ are defined over domains $\mathcal{T}_{\mathcal{I}}$ and $\mathcal{T}_{\mathcal{J}}$ respectively. This type of admissibility condition is known as the *standard admissibilty condition* [13, 16]. Thence, replacing the *min* to *max* as in

$$max(diam(\tau), diam(\sigma)) \leq \eta dist(\tau, \sigma), \tag{3.2}$$

results in what is known as the *strong admissibility condition* [13, 16]. Where the symbol η is a user chosen number, usually 2, which either relaxes the admissibility

condition or makes it more stringent.

Remark. There are other forms of admissibility conditions [13,16]. To efficiently check the admissibility condition, the sub-domains τ and σ are replaced with larger domains or are enclosed in the so-called bounding boxes $\mathcal{B}_\tau$ and $\mathcal{B}_\sigma$ respectively as shown in Figure (3.2). Therefore, the admissibility condition (3.1) can be re-written thus

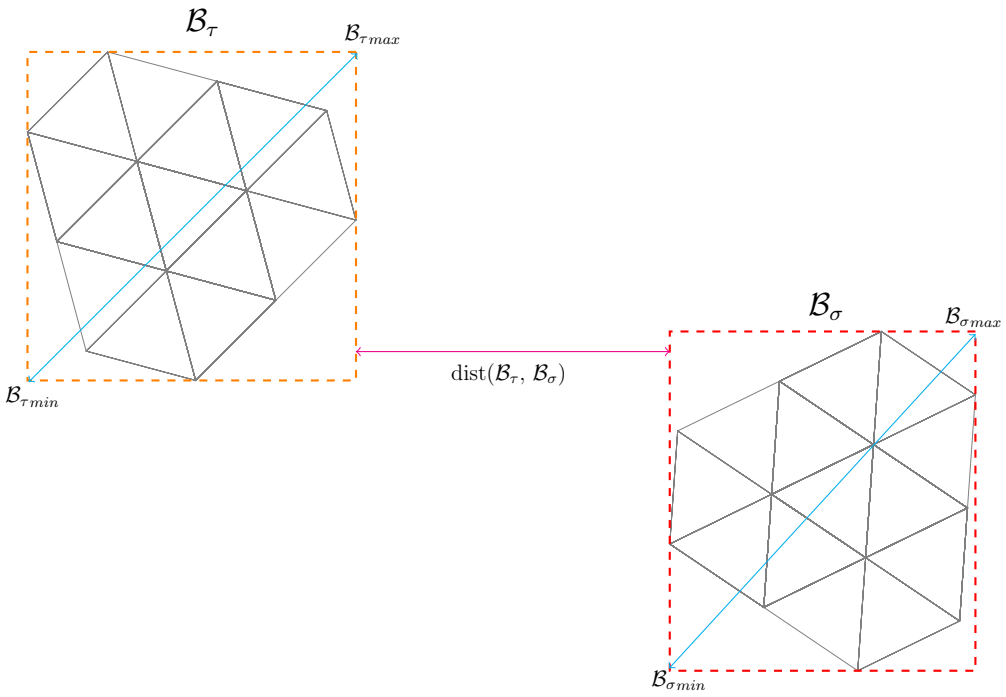


Figure 3.2: Domains τ and σ enclosed in bounding boxes $\mathcal{B}_\tau$ and $\mathcal{B}_\sigma$ respectively.

$$min(diam(\tau), diam(\sigma)) \leq \eta dist(\tau, \sigma) \approx$$

$$min(diam(\mathcal{B}_\tau), diam(\mathcal{B}_\sigma)) \leq \eta dist(\mathcal{B}_\tau, \mathcal{B}_\sigma), \quad (3.3)$$

where

$$diam(\tau) \coloneqq \max_{x_i, x_j \in \tau} \|x_i - x_j\|, \tag{3.4}$$

$$diam(\sigma) \coloneqq \max_{x_i, x_j \in \sigma} \|x_i - x_j\|, \tag{3.5}$$

$$dist(\tau, \sigma) \coloneqq \min_{x_i \in \tau, x_j \in \sigma} \|x_i - x_j\|, \tag{3.6}$$

are the Euclidean diameters and distance respectively. The parameters x_i and x_j are double variables of d-dimensional arrays corresponding to the minimal and maximal coordinates of the bounding boxes.

Definition 3.2.5 (Block Cluster Tree)**.** In definition (3.2.3), cluster tree is defined as a partition of index set $\mathcal{I}$, however, the partition over index sets $\mathcal{I} \times \mathcal{I}$ or $\mathcal{I} \times \mathcal{J}$ is known as the block cluster tree $-$ $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$ or $\mathcal{T}_{\mathcal{I}\times\mathcal{J}}$.

Remark. Given a block cluster tree $\mathcal{T}_{\mathcal{I}\times\mathcal{J}}$, the following properties hold

- The root of $\mathcal{T}_{\mathcal{I}\times\mathcal{J}}$ is $\mathcal{I} \times \mathcal{J}$,
- A given node $a \in \mathcal{T}_{\mathcal{I}\times\mathcal{J}}$ is a pair $a = (\tau, \sigma)$ where $\tau \in \mathcal{T}_{\mathcal{I}}$ and $\sigma \in \mathcal{T}_{\mathcal{J}}$,
- The notation $a = (\tau, \sigma) \in \mathcal{T}_{\mathcal{I}\times\mathcal{J}}$ is the Cartesian product of τ and σ,

- The *sons* of $a = (\tau, \sigma) \in \mathcal{T}_{\mathcal{I}\times\mathcal{J}}$ are given as

$$sons(a) = \begin{cases} sons(\sigma) = \{\}, & \{(\tau', \sigma) : \tau' \in sons(\tau)\}, \\ sons(\tau) = \{\}, & \{(\tau, \sigma') : \sigma' \in sons(\sigma)\}, \\ \text{Otherwise}, & \{(\tau', \sigma') : \tau' \in sons(\tau), \sigma' \in sons(\sigma)\}. \end{cases} \tag{3.7}$$

For instance, given a pair of index sets $\mathcal{I} = \{0, 1, 2, 3\}$ and $\mathcal{J} = \{0, 1, 2, 3, 4, 5, 6, 7\}$, the root $\mathcal{I} \times \mathcal{J}$ is $\{0, 1, 2, 3\} \times \{0, 1, 2, 3, 4, 5, 6, 7\}$. The level-wise partitioning is done using the interval $\{0, 1, 2, 3\}$ as $[0, 1]$ and interval $\{0, 1, 2, 3, 4, 5, 6, 7\}$ as $[0, 1]$ as follow

Level 0: The block is inadmissible based on the admissibility condition defined in definition (3.2.4). That is, comparing the block with itself yields $diam([0, 1]) = 1 \not\leq 0 = dist([0, 1], [0, 1])$ as shown in Figure (3.3(a)).

Level 1: Since the block is not admissible in Level 0, then it is partitioned thus

$$\{0, 1\} \times \{0, 1, 2, 3\}, \qquad \{0, 1\} \times \{4, 5, 6, 7\},$$
$$\{2, 3\} \times \{0, 1, 2, 3\}, \qquad \{2, 3\} \times \{4, 5, 6, 7\},$$

where these partition are shown in Figure (3.3(b)). Again, using the admissibility condition, none of the blocks are admissible thus the blocks are further partitioned in level 2.

Level 2: Individual sub-blocks obtained in level 1 are further partitioned as

$$\begin{array}{llll} \{0\}\times\{0,1\}, & \{0\}\times\{2,3\}, & \{0\}\times\{4,5\}, & \{0\}\times\{6,7\}, \\ \{1\}\times\{0,1\}, & \{1\}\times\{2,3\}, & \{1\}\times\{4,5\}, & \{1\}\times\{6,7\}, \\ \{2\}\times\{0,1\}, & \{2\}\times\{2,3\}, & \{2\}\times\{4,5\}, & \{2\}\times\{6,7\}, \\ \{3\}\times\{0,1\}, & \{3\}\times\{2,3\}, & \{3\}\times\{4,5\}, & \{3\}\times\{6,7\}, \end{array}$$

these highlighted partitions are shown in Figure (3.3(c)).

The blocks painted red are the inadmissible blocks while the admissible blocks are painted green. Taking the *row* 0 into consideration, it can be shown that blocks

$$\begin{aligned} \{0\}\times\{4,5\}: &\quad diam\left(\left[0,\frac{1}{4}\right]\right) = \frac{1}{4} = dist\left(\left[0,\frac{1}{4}\right],\left[\frac{2}{4},\frac{3}{4}\right]\right), \\ \{0\}\times\{6,7\}: &\quad diam\left(\left[0,\frac{1}{4}\right]\right) = \frac{1}{4} < dist\left(\left[0,\frac{1}{4}\right],\left[\frac{3}{4},\frac{4}{4}\right]\right), \end{aligned}$$

are both admissible blocks. The same procedure can be followed for the remaining blocks. The signature of the routine for creating block cluster tree is [13]

```
pblockcluster
build_blockcluster(pccluster row, pccluster col,
        BlockAdmissiblityCriterion adm,
        BlockHomogeneity hom, double eta, int leafsize);
```

Definition 3.2.6 ($\mathcal{R}k$-approximation of a matrix)**.** A matrix block of dimension $\tau\times\sigma$

and rank of at most k is said to be in a low-rank format if it is stored in a factorized form $\mathcal{R}k = AB^T,\ A \in \mathbb{C}^{\tau\times k},\ B \in \mathbb{C}^{\sigma\times k}$, where the matrices A and B are of rank k and are full matrices.

The storage requirement of an *rkmatrix* is $\mathcal{O}\left(k\left(\tau+\sigma\right)\right)$ while its corresponding full block is of $\mathcal{O}\left(\tau\times\sigma\right)$.

Remark. This low-rank approximation of a matrix block can be represented diagrammatically as shown in Figure (3.4). The routine provided for the construction of an *rkmatrix* is [13]

```
prkmatrix
new_rkmatrix(int k, int rows, int cols);
```

Definition 3.2.7 (Hierarchical Matrix, $\mathcal{H}-$Matrix)**.** A matrix $A \in \mathcal{H}(\mathcal{T}_{\mathcal{I}\times\mathcal{J}}, k)$ is stored in $\mathcal{H}-$Matrix format if all its admissible blocks are represented as compressible *rkmatrix* blocks and all its inadmissible blocks are stored as full matrices as shown in Figure (3.3(c)).

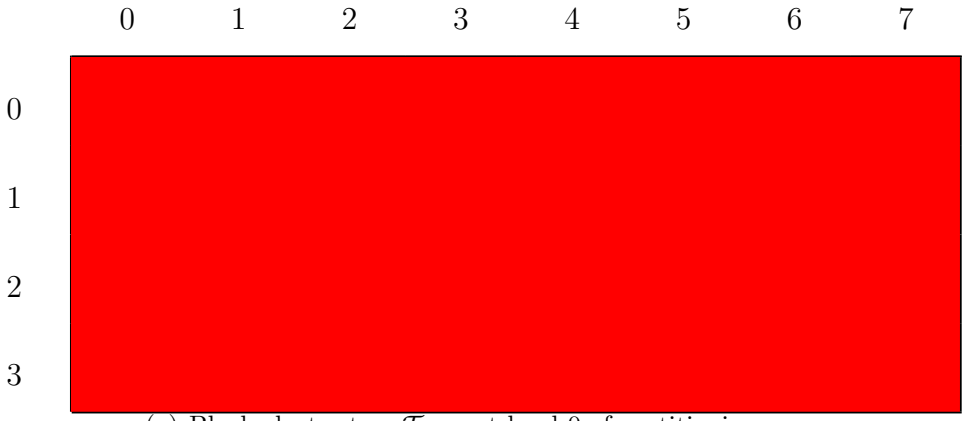
Remark. Individual block, says $\tau\times\sigma$, in a *block cluster tree* (for instance Figure (3.3(c))) is given by

$$\tau\times\sigma = \begin{cases} \text{leaf} \in \text{admissible}, & rkmatrix, \\ \text{leaf} \notin \text{admissible}, & fullmatrix, \\ \text{Otherwise}, & supermatrix\left\{sons(\tau',\sigma') : \tau' \in sons(\tau), \sigma' \in sons(\sigma)\right\}, \end{cases} \tag{3.8}$$

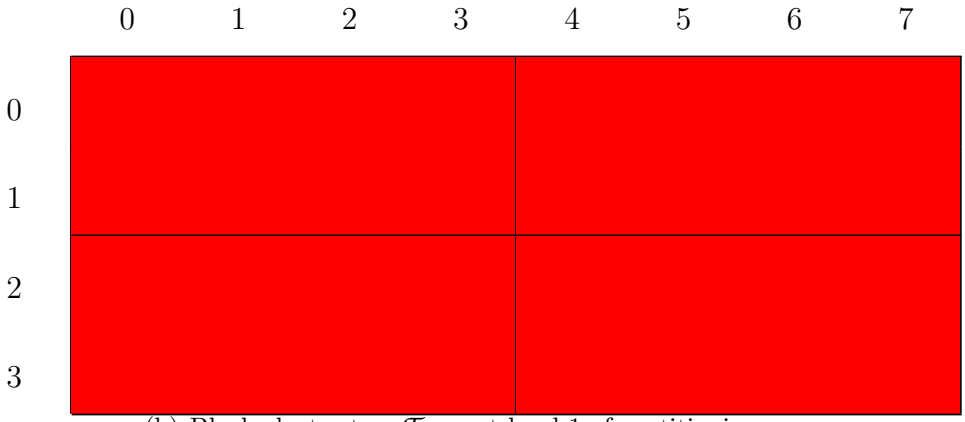
where the structure of a *supermatrix* is given in Figure (3.5). By implementation, if a block is a *rkmatrix* then both the *fullmatrix* and *supermatrix* nodes of the block are set to *nullptr*. Similarly, if a block is a *fullmatrix* then both the *rkmatrix* and *supermatrix* nodes of the block are *nullptrs*. Finally, if a block is a *supermatrix*, both the *rkmatrix* and *fullmatrix* nodes are *nullptrs*.

Examining the structure of a *supermatrix* and a *block cluster tree* shows that both structures closely resemble. Hence, obtaining a *supermatrix* from a *block cluster tree* can be done directly using the HLib method [13]

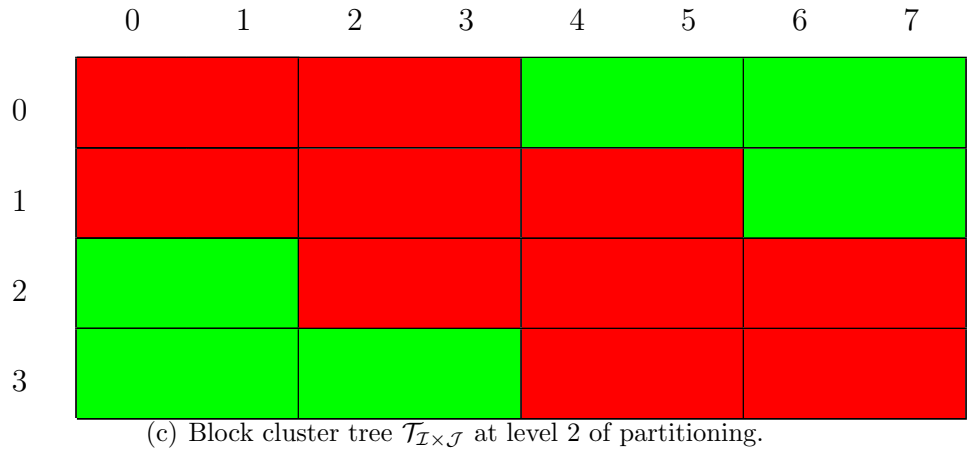
```
1 psupermatrix
2 build_supermatrix_from_blockcluster(pblockcluster bc, int k)
```



(a) Block cluster tree $\mathcal{T}_{\mathcal{I}\times\mathcal{J}}$ at level 0 of partitioning.



(b) Block cluster tree $\mathcal{T}_{\mathcal{I}\times\mathcal{J}}$ at level 1 of partitioning.



(c) Block cluster tree $\mathcal{T}_{\mathcal{I}\times\mathcal{J}}$ at level 2 of partitioning.

Figure 3.3: Hierarchical partitioning of index set $\mathcal{I} = \{0,1,2,3\}$ and $\mathcal{J} = \{0,1,2,3,4,5,6,7\}$ to form the block-cluster tree $\mathcal{T}_{\mathcal{I}\times\mathcal{J}}$

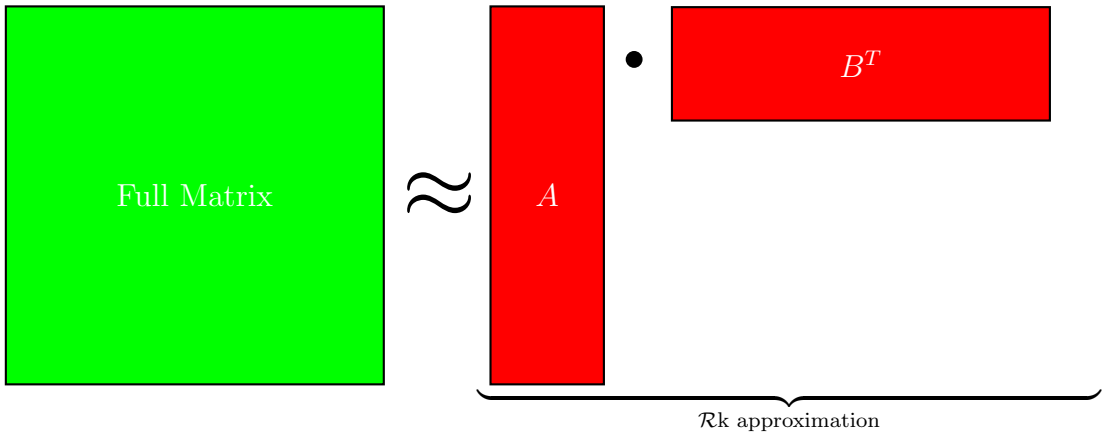


Figure 3.4: Low-rank ($\mathcal{R}$k) approximation of an admissible block in a compressed form AB^T.

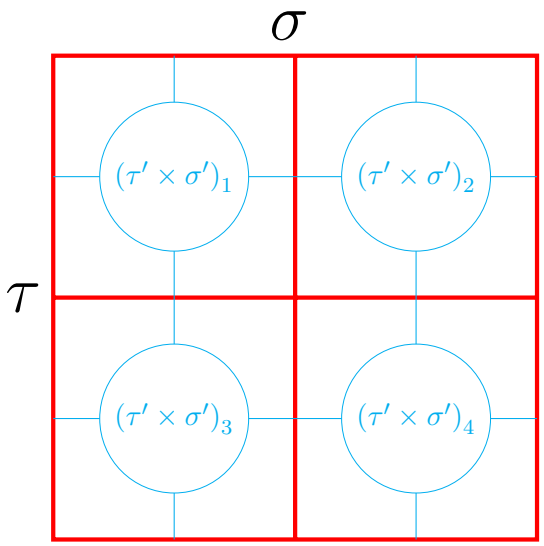


Figure 3.5: Supermatrix structure of $\tau \times \sigma$ with $sons(\tau', \sigma') : \tau' \in sons(\tau), \sigma' \in sons(\sigma)$.

Chapter 4

Memory and Computational Costs Reduction Strategies

4.1 Introduction

This chapter introduces the step-by-step procedure involved in the proposed complexity reduction techniques of a scattering problem on a conducting metal object. Therefore, the focus of this chapter is the introduction of new strategies which are applicable for reducing the size of the problem.

Two (2) new techniques are introduced in this research work for reducing the complexity of the problem. Each of these techniques relies on the exploitation of the skin effect of the object's medium. These techniques are:

1. Model Based Technique: It involves the extraction of the volumetric mesh ele-

ments where the size of the extraction depends on a chosen value.

2. Interaction Range Truncation: This requires zeroing out the values of the matrix entries where the distance between the sources and observers exceed a certain value.

4.2 Model Design

In order to implement Item 1 of Section (4.1) – that is, the volume elements extraction, the object model is created in such a way that the volumetric section of the model can be extracted. Therefore, the sphere model is designed to have:

- Exterior Radius (R): It determines the size of the sphere object being modelled.
- Interior Radius (r): It serves as the probe used to set the extraction width. It depends on R, skin depth δ and a chosen $factor$ as depicted in equation (4.2).

$$\text{Skin Depth}, \delta = \sqrt{\frac{1}{\pi f \mu_0 \mu_r \sigma}} \tag{4.1}$$

$$\text{Interior Radius}, r = R - (\delta * factor) \tag{4.2}$$

$$\text{Characteristics Length}, s = \frac{\delta}{Number} \tag{4.3}$$

Figure (4.1) shows the ordering of the interior and the exterior radii during the design. This layout allows for extraction of the volumetric mesh elements by varying the $factor$.

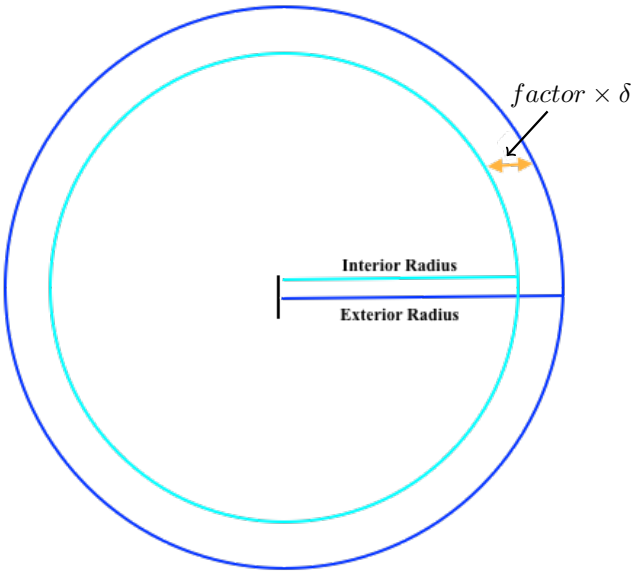


Figure 4.1: Object model showing the layout of the interior and exterior radii as a function of the skin depth ($factor \times \delta$) of the conducting object

The object model depends on:

- Skin Depth (δ): The skin depth of the material medium can be obtained by using the equation (4.1), where f is the frequency, μ_0 is the permeability of the free space, μ_r is the the relative permeability of the object and σ is the conductivity of the object.

- Factor ($factor$): A user chosen scalar number which determines the extent of the extraction width.

- Number of elements per skin depth ($Number$): It determines the mesh element's characteristics length (s) as shown in equation (4.3).

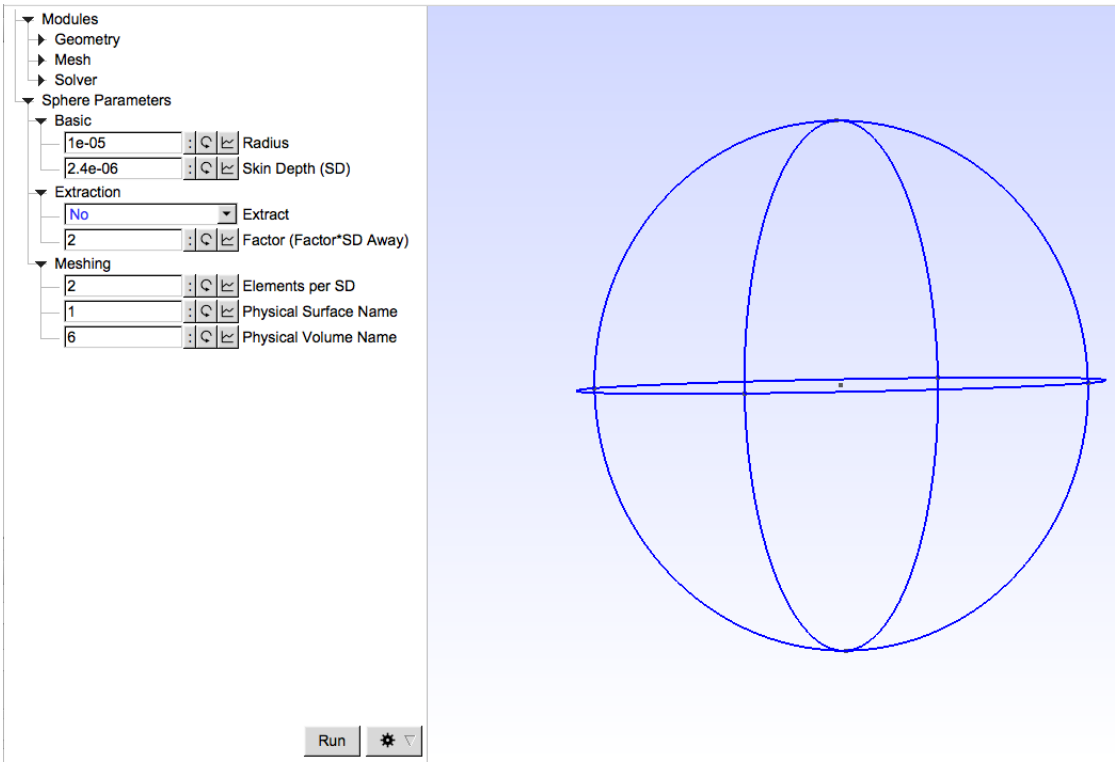


Figure 4.2: Model object screen-shot obtained from Gmsh software [10] showing the model when option for volumetric element extraction is set to 'No'. Notice the input parameters on the left menu section.

In this work, the model object is designed with Gmsh software [10]. The comparison between the solutions obtained from the conducting object before and after the extraction is needed. In order to cater for this in the object model, there is an option for selecting whether the object model should be extracted or not as shown in Figures (4.2) and (4.3). Appendix (A) contains the Gmsh [10] code used for the creation of the sphere model.

Alternatively, this model designed section can be avoided by meshing the object in its entirety and manually skipping the volumetric elements that do not fall within the region of interest. However, this kind of technique has been found to increase the

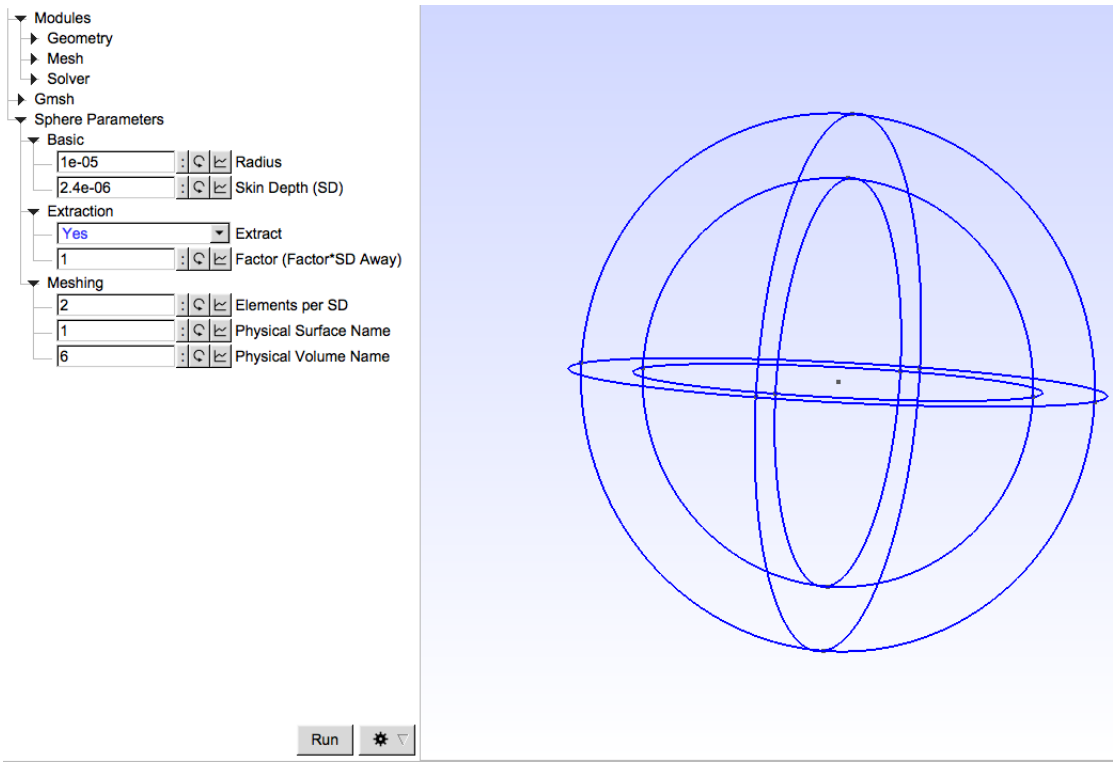


Figure 4.3: Model object screen-shot obtained from Gmsh software [10] showing the model when option for volumetric element extraction is set to 'Yes'. Notice the input parameters on the left menu section.

time taken to read the mesh file of the object model.

4.3 SVS-EFIE $\mathcal{H}-$Matrices Based Discretization

As shown in Section (2.7), SVS-EFIE has three (3) matrices, namely Surface-to-Surface, Volume-to-Surface and Surface-to-Volume matrices. Each of Surface-to-Volume and Volume-to-Surface matrices has x, y, and z components. Considering these components coupled with the Surface-to-Surface matrix results in a total of seven (7) matrices.

In order to employ $\mathcal{H}-$Matrices concepts, each of these matrices is represented in $\mathcal{H}-$Matrix form. Below are the procedural steps needed to apply $\mathcal{H}-$Matrices to SVS-EFIE:

(1.) Set the *index set* $\mathcal{I}$ to be the indices of the RWG basis functions on the surface triangular patches. The total number of RWG basis functions being P, therefore, the size of $\mathcal{I}$ is P.

(2.) Set the *index set* $\mathcal{J}$ to be the indices of the centroids of the tetrahedral elements. N is the total number of all the tetrahedral elements and the length of $\mathcal{J}$ is N.

(3.) Construct the *cluster tree*, ct_s, from the *index set* $\mathcal{I}$ to represent the *cluster tree* of the triangle elements.

(4.) Construct the *cluster tree*, ct_v, from the *index set* $\mathcal{J}$ which denotes the *cluster tree* of the volumetric tetrahedral elements.

(5.) Create *block cluster tree* $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$ from cluster trees $\mathcal{T}_{\mathcal{I}}$ and $\mathcal{T}_{\mathcal{I}}$ representing row and

column trees respectively. This *block cluster tree* corresponds to the Surface-to-Surface matrix $Z_{\epsilon}^{\partial V,\partial V}$.

(6.) Convert the *block cluster tree* $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$ in Item (5.) to *supermatrix* $Z_{\epsilon}^{\partial V,\partial V}$. The size of this *supermatrix* is $P \times P$.

(7.) Create *block cluster tree* $\mathcal{T}_{\mathcal{I}\times\mathcal{J}}$ from cluster trees $\mathcal{T}_{\mathcal{I}}$ and $\mathcal{T}_{\mathcal{J}}$ representing row and column trees respectively. This *block cluster tree* corresponds to the Volume-to-Surface matrix $Z_{0}^{\partial V,V}$.

(8.) Convert the *block cluster tree* $\mathcal{T}_{\mathcal{I}\times\mathcal{J}}$ in Item (7.) to *supermatrix* $Z_{0}^{\partial V,V}$. The size of this *supermatrix* being $P \times N$.

(9.) Since $Z_{0}^{\partial V,V}$ has x, y, and $z-$components, hence repeat Items (7.) and (8.) for each of these components such that there exist supermatrices $\left[Z_{0}^{\partial V,V}\right]_x$, $\left[Z_{0}^{\partial V,V}\right]_y$ and $\left[Z_{0}^{\partial V,V}\right]_z$.

(10.) Create *block cluster tree* $\mathcal{T}_{\mathcal{J}\times\mathcal{I}}$ from cluster trees $\mathcal{T}_{\mathcal{J}}$ and $\mathcal{T}_{\mathcal{I}}$ representing row and column trees respectively. This *block cluster tree* corresponds to the Surface-to-Volume matrix $Z_{\epsilon}^{V,\partial V}$.

(11.) Convert the *block cluster tree* $\mathcal{T}_{\mathcal{J}\times\mathcal{I}}$ in Item (10.) to *supermatrix* $Z_{\epsilon}^{V,\partial V}$. The dimension of this *supermatrix* is $N \times P$.

(12.) Since $Z_{\epsilon}^{V,\partial V}$ has x, y, and $z-$components, hence repeat Items (10.) and (11.) for each of these components such that there exist supermatrices $\left[Z_{\epsilon}^{V,\partial V}\right]_x$, $\left[Z_{\epsilon}^{V,\partial V}\right]_y$

and $\left[Z_{\epsilon}^{V,\partial V}\right]_z$.

(13.) For each *supermatrix* created in Items (6.), (8.) and (11.), traverse the blocks of the supermatrix by creating a loop through the *sons* of the column tree and *sons* of the row tree. Within these loops, compute the interactions between a given pair of clusters or sons, denoted as *row* and *column*, and check if the block is a leaf block and if it is admissible or not. If it it inadmissible then compute all the matrix entries and store it as a $fullmatrix$ block. Otherwise, compute the entries of the block and compress with either ACA or SVD and store it as a *rkmatrix* block.

(14.) Having computed and filled the blocks of the supermatrices, the final matrix Z_{SVS} can be obtained by computing the product of the rectangular supermatrices $\left[Z_0^{\partial V,V}\right] \cdot \left[Z_{\epsilon}^{V,\partial V}\right]$ and adding the results to the $\left[Z_{\epsilon}^{\partial V,\partial V}\right]$ using the HLib routine [13]

```
void
muladd_supermatrix(psupermatrix c, psupermatrix a, psupermatrix b);
```

so that the supermatrix $\left[Z_{\epsilon}^{\partial V,\partial V}\right]$ becomes the $[Z_{SVS}]$ matrix as shown in Figure (4.4).

(15.) Finally, the SLAE $[Z_{SVS}]\,[I] = [V]$ obtained is solved using LU decomposition. However, the system of equation can also be solved iteratively. Solving the system iteratively does not necessarily require the explicit creation of Z_{SVS}

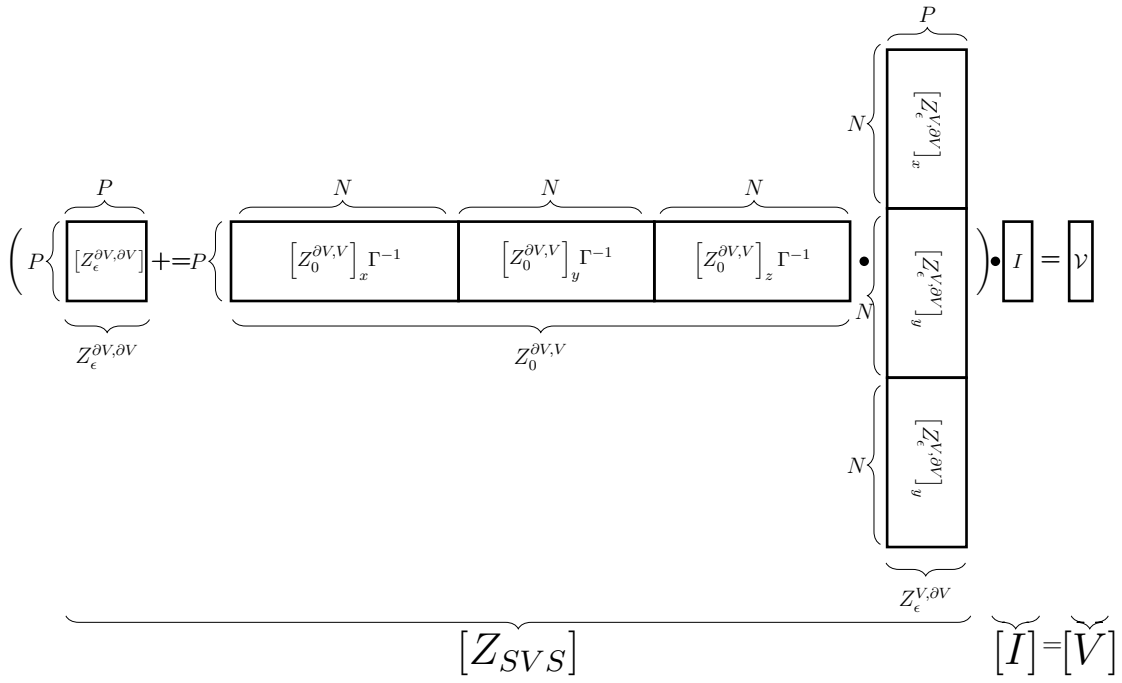


Figure 4.4: $\mathcal{H}-$Matrices based method of moment discretization of Surface-Volume-Surface Electric Field Integral Equation.

supermatrix.

4.4 Interactions Range Truncation

Following the steps highlighted in Section (4.3) produces $\mathcal{H}-$Matrices based SVS-EFIE solver. In order to achieve interaction range truncation stated in Item 2 of Section (4.1), the procedural steps in Section (4.3) have to be modified or updated.

The steps are modified for supermatrices $\left[Z_\epsilon^{V,\partial V}\right]$ and $\left[Z_\epsilon^{\partial V,\partial V}\right]$, the steps in Section (4.3) remain intact for $\left[Z_0^{\partial V,V}\right]$ supermatrix computation. This implies that, interaction range truncation will only be applicable to the supermatrices $\left[Z_\epsilon^{V,\partial V}\right]_x$,

$\left[Z_{\epsilon}^{V,\partial V}\right]_y$, $\left[Z_{\epsilon}^{V,\partial V}\right]_z$ and $\left[Z_{\epsilon}^{\partial V,\partial V}\right]$.

The range truncation depends on the exploitation of the skin-effect of the metal object. It is expected that the interaction between a pair of sons or clusters, says $\tau \times \sigma : \tau \in sons(\mathcal{T}_{\mathcal{J}}), \sigma \in sons(\mathcal{T}_{\mathcal{I}})$ of a block cluster tree or a supermatrix $\mathcal{T}_{\mathcal{J}\times\mathcal{I}}$, stops after the distance between these sons or clusters exceed a given value called the truncation distance $\mathcal{D}_T$. The parameter $\mathcal{D}_T$ is defined thus

$$\text{Truncation distance}, \mathcal{D}_T = (\delta * factor)\,, \tag{4.4}$$

where parameter δ is the skin-depth defined in equation (4.1) and $factor$ is another user chosen scalar number just like the one defined in Section (4.2).

4.4.1 Distance between two clusters

As shown in Definition (3.2.4), the distance between a pair of clusters can be computed efficiently by introducing bounding boxes. Therefore, HLib library is programmed in such way that those clusters are enclosed in the so-called bounding boxes. Calculating distance between a pair of clusters is equivalent to computing the distance between their bounding boxes as shown in Figure (3.2) using the equation (3.6). While the computation of the distance between a pair of clusters or sons is not really the challenge, the position of a given cluster relative to the other requires adequate consideration in order to obtain accurate distance. All the possible combinations

have to be considered. As depicted in Figure (3.2), the details about a given bounding box are known. The relative positions of a pair of bounding boxes are presented in Figure (4.5).

Coding up all of the possibilities illustrated in Figure (4.5) with the equation (3.6) requires the following pseudo-code steps:

(1.) Set procedure name as `getDistance`

(2.) Get input clusters `A` and `B`

(3.) Initialize variable `sigma:=0` to hold the sum of squares of the distances

(4.) Set variable `d` to be the spatial dimension of one of the bounding boxes, `d:=A->d`

(5.) For `counter:=1,...,d`

  (a) Get the minimum value of the bounding box of `A` and store it in variable `aMin`, such that `aMin:=A->bmin[counter]`

  (b) Get the maximum value of the bounding box of `A` and store it in variable `aMax`, such that `aMax:=A->bmax[counter]`

  (c) Get the minimum value of the bounding box of `B` and store it in variable `bMin`, such that `bMin:=B->bmin[counter]`

  (d) Get the maximum value of the bounding box of `B` and store it in variable `bMax`, such that `bMax:=B->bmax[counter]`

  (e) Set variable `abMax := |aMax - bMax|`

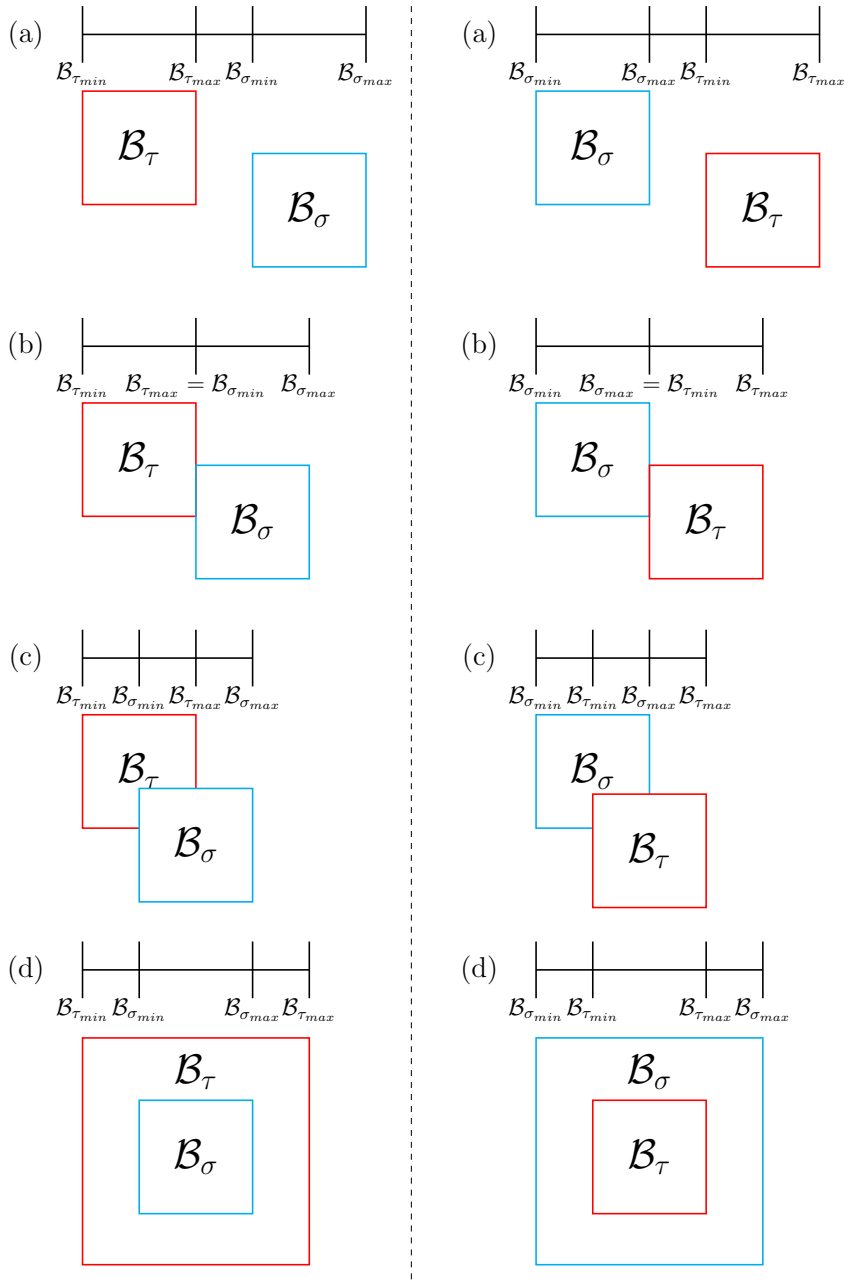


Figure 4.5: All possible relative positions of a pair of bounding boxes $\mathcal{B}_\tau$ and $\mathcal{B}_\sigma$ enclosing the clusters τ and σ respectively.

(f) If `aMin > bMin` set

```
sigma:= sigma + (if(bMax>aMin) then min(abMax,bMax-aMin) else
```

`aMin-bMax)`^{2}

(g) `Else` set

`sigma:= sigma + (if(aMax>bMin) then min(abMax,aMax-bMin) else bMin-aMax)`^{2}

(6.) Computer the square root of `sigma`

(7.) Return the the computed square root value

The C++ programming language implementation of the pseudo-code is given in the Appendix (B).

4.4.2 Modifications to $\mathcal{H}-$Matrices Steps

All the necessary modifications are done prior to filling the supermatrices. The filling of a supermatrix requires the computation of the interactions between the sources and observers − basically row and column clusters. The results of these computations are stored in a $fullmatrix$ form or in compressed $rkmatrix$ format. The filling procedure of a supermatrix is described in the Item (13.) of Section (4.3). Therefore, the modifications are applied only to the Item (13.) of Section (4.3), where the remaining steps are left intact. The applied changes are given thus:

(13.)* When traversing the supermatrix, given a leaf block $S = (\tau, \sigma) \in \mathcal{T}_{\mathcal{I}\times\mathcal{I}}$, the distance between τ and σ is first computed by making use of the routine `getDistance(`S`->`τ`, `S`->`σ`)` coded in Section (4.4.1).

The HLib routine [13]

```
void
reallocstructure_rkmatrix(prkmatrix r, int rank);
```

can be used to resize a given *rkmatrix* from rank k to 0 - literally converting an *rkmatrix* block to an empty block. It is desired that interactions are truncated for some blocks exceeding the truncation distance $\mathcal{D}_T$, meaning, those blocks beyond $\mathcal{D}_T$ are basically empty blocks. If the distance between the two clusters is less than $\mathcal{D}_T$ the unmodified step in Item (13.) of Section (4.3) is followed otherwise this modified Item (13.)* is followed. In order to take advantage of this routine, the block S is checked if it a *fullmatrix* or an *rkmatrix* block.

If the block is a *fullmatrix* block, then S->f is deleted and set to *nullptr*. Then, the node S->r, which normally is a *nullptr* if a leaf block is a *fullmatrix*, is then set to a newly created *rkmatrix* block of rank 1. Thereafter, the node S->r is resized to size 0 using the `reallocstructure_rkmatrix()` routine. This finally makes the block an empty block.

However, if the block is an *rkmatrix* block, ACA or SVD compression stage is totally eliminated. The only operation here is to convert the block to an empty block using `reallocstructure_rkmatrix()` routine.

It should be noted that, in both cases, *fullmatrix* and *rkmatrix* cases, all the operations are done prior to the computations of the interactions - or simply before the matrix fill. The stepwise structure of these operations as explained

above is depicted in equation (4.5), where the method `del_fullmatrix()` is the routine for deleting a *fullmatrix*, `new()` is actually `new_rkmatrix()` routine for creating a new *rkmatrix* block and the routine `realloc()` is actually `reallocstructure_rkmatrix()` method for resizing an *rkmatrix* - it is used here to create an empty block.

$$S = \begin{cases} \texttt{dist}\,(S\texttt{->}\tau, S\texttt{->}\sigma) > \mathcal{D}_T & : \begin{cases} S\texttt{->f} \neq \texttt{nullptr} : & \texttt{del\_fullmatrix}(S\texttt{->f}), \\ & S\texttt{->f} = \texttt{nullptr}, \\ & S\texttt{->r} = \texttt{new(1, rows, cols)}, \\ & \texttt{realloc(S->r, 0)}, \\ S\texttt{->r} \neq \texttt{nullptr} : & \texttt{realloc(S->r, 0)}, \end{cases} \\ \texttt{otherwise} & : \text{Fill the matrix block as appropriate,} \end{cases} \tag{4.5}$$

4.5 Validity

In order to validate the results, the solution obtained is compared against the Mie series solution [20, 34]. However, to be able to track the extent of the effect of the extraction and/or range truncation on the final solution a new parameter is needed. This quest leads to the computation of the Specific Absorption Rate (SAR). Since only the volumetric mesh elements are extracted, the SAR is then computed at the centroids of the tetrahedral elements. This implies that the SAR is computed for

both full sphere model and for the extracted model in order to be able to calculate the extent of the extraction effect on the solution.

The formula for obtaining the SAR is given by [11]

$$\texttt{SAR} = \frac{\sigma}{mass} \sum_{i\ =1}^{N} \left[\left(|\boldsymbol{J}_{ix}|^2 + |\boldsymbol{J}_{iy}|^2 + |\boldsymbol{J}_{iz}|^2 \right) \cdot \Delta V_i \right], \tag{4.6}$$

where N is the total number of tetrahedral elements, $mass$ is the mass of the scatterer in kg, σ is the conductivity of the scatterer, ΔV_i is the volume of ith tetrahedral, while $\boldsymbol{J}_{ix}$,$\boldsymbol{J}_{iy}$ and $\boldsymbol{J}_{iz}$ are the electric field components in the ith tetrahedral for x, y, and $z-$components respectively.

4.6 Test Cases Analysis

Having completed the architectural design of the techniques and implementation of the procedures in C++ programming language, a certain test case at different frequencies has to be conducted using the developed C++ program.

The so-called test cases are tailored in such a way as to addressed the set objectives and also to proffer answers to the research questions highlight in Sections (1.3) and (1.4) of Chapter (1).

The set-up details of the test cases are presented in Table 4.1. From the table, it can be observed that both the extraction width and interaction range truncation are fixed with respect to the electrical size of the problem. This is desired so as to

have a nearly perfect comparisons which are based solely on the frequency - since the frequency varies.

Table 4.1: Set-up information of the test-cases.

Set-up Item	Details
Object Models (.msh files)	Full sphere and Extracted Sphere
Frequencies (GHz)	10, 15, 20, 25, 30
Incident Electric Field	Electric dipole with electric dipole moment 1(A.m) located at $z = 0.00003m$
Scatterer Properties	$R = 10\mu m, \epsilon_r = 1.5, \sigma = 4.5e^{+06}$
Model Extraction	Extraction Factor, $factor = 2$
Interaction Range Truncation	Truncation Factor, $factor = \frac{R}{\delta}$, then $\mathcal{D}_T = R$
SAR Computation	Full Sphere SAR (from C++ code), Extracted Sphere SAR (from C++ code), Full Sphere SAR (from Analytical Mie series solution)
Machine Specifications	CPU model = Intel Xeon X5650, Clock Frequency = 2.66GHz

Continued on next page

Table 4.1 – continued from previous page

Set-up Item	Details
$\mathcal{H}-$Matrices Parameters	$n_{min} = 64$, Maximum $\mathcal{H}-$Matrices Level = 20, Clustering Strategy: Geometrically Balanced, Admissibility Condition: Standard, $\eta = 2$, Block Compression Algorithm: SVD, $\mathcal{H}-$Arithmetic Tolerance: 10^{-3}, Compression Tolerance: 10^{-5}

It can be noted from Table 4.1 that SVD was chosen in-place of ACA as the compression algorithm. Although, both the ACA and SVD was implemented in this research work. The choice of SVD over ACA was as a result of the inability of the ACA to compress the majority of the admissible blocks owing to the nature of the ACA algorithm.

Chapter 5

Results

5.1 Brief Overview

All the test cases presented in Section (4.6) of Chapter (3) are conducted with respect to the stated parametric inputs. These results are presented in this chapter in ascending order of the chosen frequency range.

Moreover, while Table 4.1 in Section (4.6) of Chapter (3) highlighted some of the input parameters that are general to all the test cases, however, there are other input parameters that are not included in the table. The missing parameters are, the number of triangle and tetrahedral mesh elements and the degree of freedom of each problem. These parameters are frequency dependent thus, they are included as part of the results for each frequency. Special emphasis is placed on the CPU time (time complexity) and memory usage (memory complexity) at each frequency.

Furthermore, visual images of the supermatrices are included so as to showcase the effect of the range of interaction truncation on the structure of the supermatrices. However, only the Surface-to-Surface supermatrix structures are presented in order to save space. The volumetric current distribution, and relative error distribution with respect to the Mie series solution are included at each frequency so as to be able to visualize the accuracy or otherwise of the proposed techniques in this research work.

Conclusively, the descriptive analysis of the results obtained is done in order to verify the validity and/or correctness of the proposed techniques in-line with the statements of the hypothesis.

5.2 Results of the Test Cases

In the developed C++ code the parameters were set based on the given input values (Table 4.1), additionally, the frequency was set based on the cases presented − that is, $10-30$ GHz. Hence, after computing the skin-depth δ and meshing the sphere model, the additional set-up values obtained are given in Tables 5.1, 5.4, 5.7, 5.10 and 5.13. Having obtained all the set-up parameters needed, the simulation is started and after the end of the simulation, the results relating to the time and memory usages are tabulated in Tables 5.3, 5.6, 5.9, 5.12 and 5.15. Moreover, the results relating to the computed SAR are presented in Tables 5.2, 5.5, 5.8, 5.11 and 5.14. The supermatrix structures are compared in Figures (5.1), (5.5), (5.9), (5.13) and (5.17) where the red blocks represent the $fullmatrix$ blocks, the green blocks denote the $rkmatrix$

blocks and white blocks are the empty blocks representing the skipped blocks that fall beyond the truncation distant $\mathcal{D}_T$. Figures (5.3), (5.7), (5.11), (5.15) and (5.19) show the magnitude of electric field in the volume of the scatterer. Figures (5.4), (5.8), (5.12), (5.16) and (5.20) show the relative error distribution of the solutions of the full sphere and the extracted sphere relative to the Mie series analytical solution. Lastly, the magnitude of the electric field as a function of the polar angle θ at the centroids of the tetrahedral elements nearest the exterior radius R of the extracted sphere and nearest the interior radius r are plotted against the Mie series counterpart as shown in Figures (5.2), (5.6), (5.10), (5.14) and (5.18).

5.2.1 Case 1: 10 GHz

Table 5.1: Set-up parameteric values for 10 GHz.

ADDITIONAL SET-UP PARAMETERS		
Skin Depth	2.36μm	
	Full Sphere	**Extracted Sphere**
No of 2-D Elements	2, 312	2, 312
No of 3-D Elements	11, 496	9, 953
No of RWG Basis Functions	3, 468	3, 468
No of Degree of Freedom	37, 956	33, 327

Table 5.2: SAR results obtained at 10 GHz.

Specific Absorption Rate (SAR) Results	
SAR of the Mie serie solution $\mathtt{SAR}_{Mie}$	$9.12e^{-09}$
SAR of the full sphere, $\mathtt{SAR}_{FullSphere}$	$8.82e^{-09}$
SAR of the extracted sphere, $\mathtt{SAR}_{ExtractedSphere}$	$8.43e^{-09}$
$\frac{\lvert\mathtt{SAR}_{Mie} - \mathtt{SAR}_{FullSphere}\rvert}{\lvert\mathtt{SAR}_{Mie}\rvert}$	03%
$\frac{\lvert\mathtt{SAR}_{Mie} - \mathtt{SAR}_{ExtractedSphere}\rvert}{\lvert\mathtt{SAR}_{Mie}\rvert}$	08%
$\frac{\lvert\mathtt{SAR}_{FullSphere} - \mathtt{SAR}_{ExtractedSphere}\rvert}{\lvert\mathtt{SAR}_{FullSphere}\rvert}$	04%

Table 5.3: Time and memory usages at 10 GHz.

RESULTS (Time: hrs, Memory: GB)						
	Full Sphere			**Extracted Sphere**		
	$Z_\epsilon^{\partial V,\partial V}$	$Z_\epsilon^{V,\partial V}$	$Z_0^{\partial V,V}$	$Z_\epsilon^{\partial V,\partial V}$	$Z_\epsilon^{V,\partial V}$	$Z_0^{\partial V,V}$
Filling Time	0.03	0.10	0.16	0.03	0.08	0.13
Memory	0.15	1.25	1.13	0.14	1.05	0.93
Time to Form Z_{SVS}	0.42			0.40		
LU Time	0.03			0.04		
Solution Time	$2.21e^{-05}$			$5.92e^{-05}$		
Total Time	**0.75**			**0.68**		

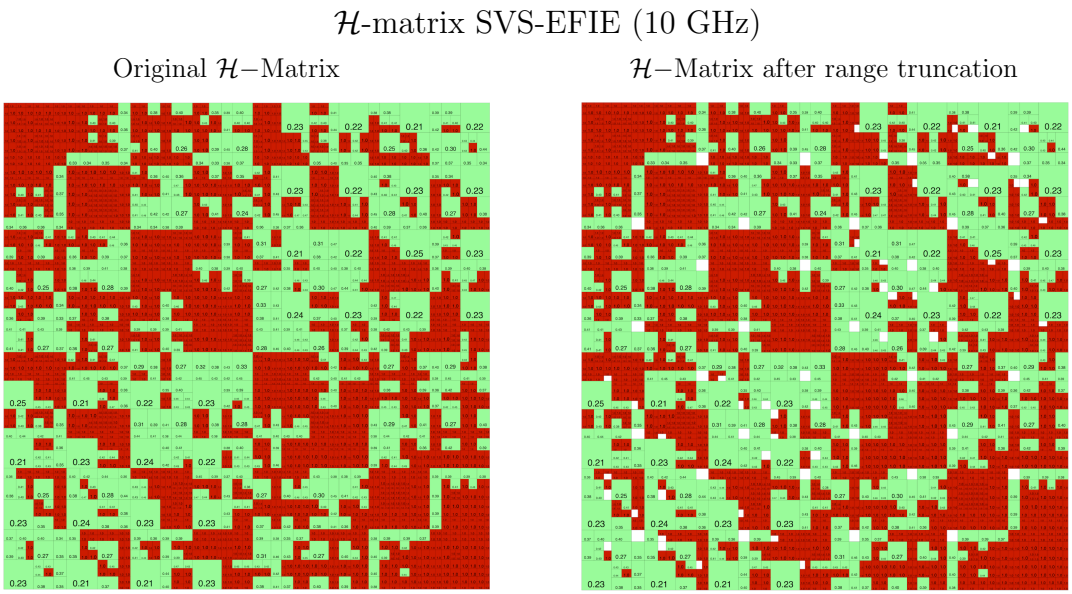


Figure 5.1: Surface-to-Surface $\mathcal{H}-$Matrix structures before and after the range truncation due to the skin-effect of the Lead sphere produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ obtained by the $\mathcal{H}$-matrix accelerated SVS-EFIE solver at $10\,\mathrm{GHz}$, radius of the sphere $R = 10\mu m$, and truncation distance $\mathcal{D}_T = R$.

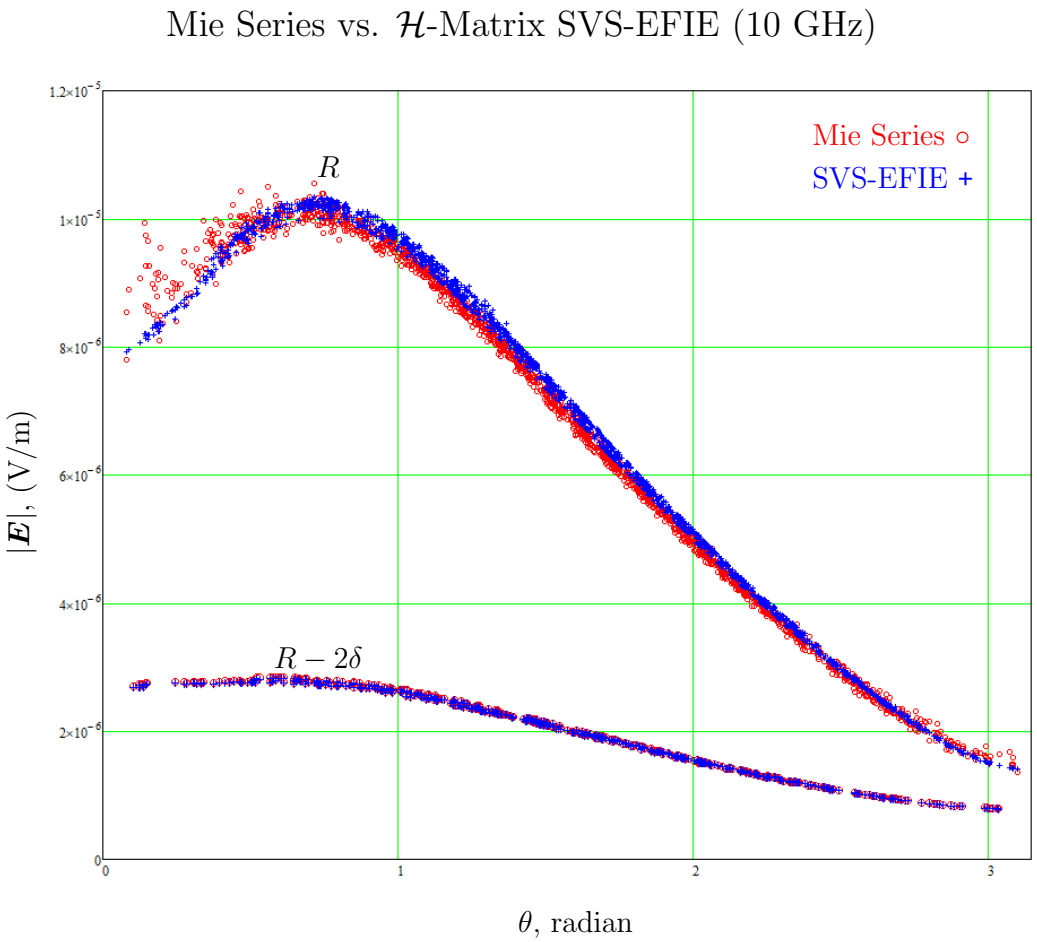


Figure 5.2: Magnitude of the electric field in Lead sphere of radius $10\mu m$ obtained via $\mathcal{H}-$Matrices accelerated SVS-EFIE due to radial electric dipole at 10 GHz as a function of the polar angle θ at the centroids of the tetrahedral elements nearest to the outer radius R of the extracted sphere (figure (5.3)) and nearest to the inner radius $r = R - 2\delta$ of the volume removed from the sphere due to skin-effect attenuation.

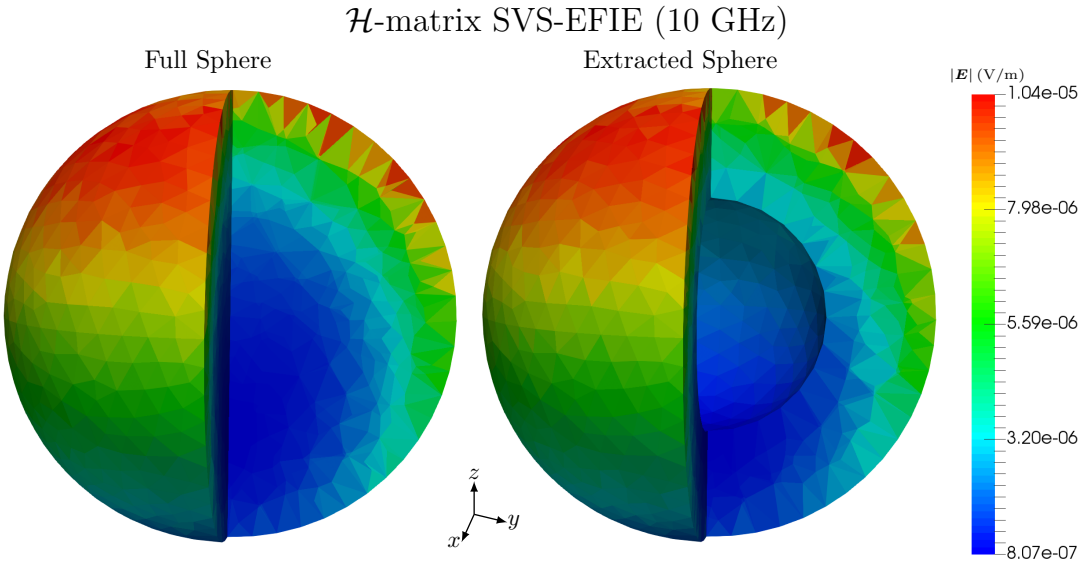


Figure 5.3: Magnitude of the total electric field inside Lead (Pb) sphere produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ obtained by the $\mathcal{H}$-matrix accelerated SVS-EFIE solver at 10 GHz

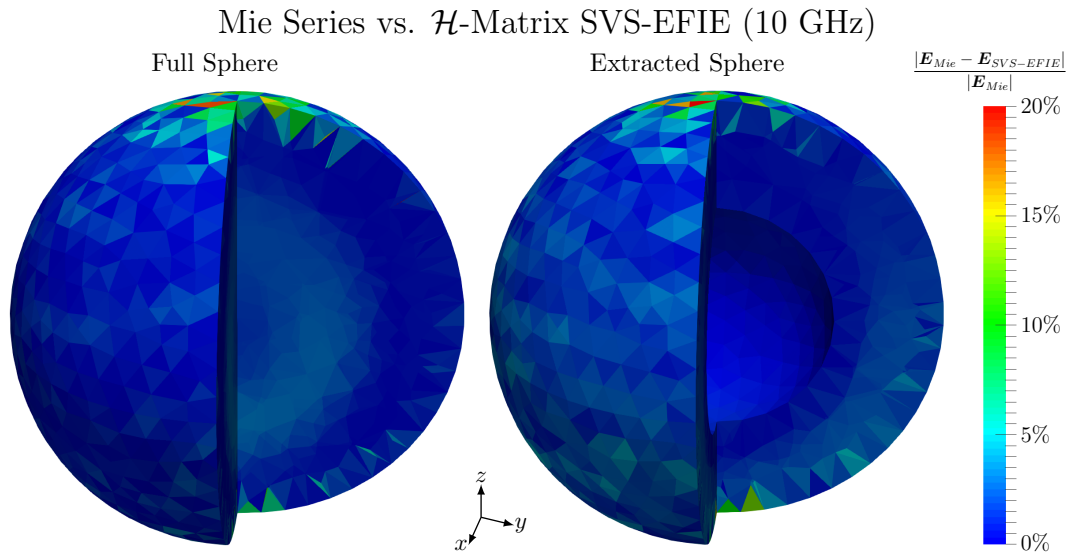


Figure 5.4: The relative error distribution of the total electric field inside extracted Lead (Pb) sphere, produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ at 10 GHz, in Mie series solution against SVS-EFIE.

5.2.2 Case 2: 15 GHz

Table 5.4: Set-up parameteric values for 15 GHz.

ADDITIONAL SET-UP PARAMETERS		
Skin Depth	1.93μm	
	Full Sphere	**Extracted Sphere**
No of 2-D Elements	4, 078	4, 078
No of 3-D Elements	24, 305	15, 609
No of RWG Basis Functions	6, 117	6, 117
No of Degree of Freedom	79, 032	52, 944

Table 5.5: SAR results obtained at 15 GHz.

Specific Absorption Rate (SAR) Results	
SAR of the Mie serie solution $\mathtt{SAR}_{Mie}$	$9.68e^{-10}$
SAR of the full sphere, $\mathtt{SAR}_{FullSphere}$	$9.41e^{-10}$
SAR of the extracted sphere, $\mathtt{SAR}_{ExtractedSphere}$	$8.77e^{-10}$
$\frac{\lvert\mathtt{SAR}_{Mie} - \mathtt{SAR}_{FullSphere}\rvert}{\lvert\mathtt{SAR}_{Mie}\rvert}$	03%
$\frac{\lvert\mathtt{SAR}_{Mie} - \mathtt{SAR}_{ExtractedSphere}\rvert}{\lvert\mathtt{SAR}_{Mie}\rvert}$	09%
$\frac{\lvert\mathtt{SAR}_{FullSphere} - \mathtt{SAR}_{ExtractedSphere}\rvert}{\lvert\mathtt{SAR}_{FullSphere}\rvert}$	07%

Table 5.6: Time and memory usages at 15 GHz.

RESULTS (Time: hrs, Memory: GB)						
	Full Sphere			**Extracted Sphere**		
	$Z_{\epsilon}^{\partial V,\partial V}$	$Z_{\epsilon}^{V,\partial V}$	$Z_{0}^{\partial V,V}$	$Z_{\epsilon}^{\partial V,\partial V}$	$Z_{\epsilon}^{V,\partial V}$	$Z_{0}^{\partial V,V}$
Filling Time	0.08	0.39	0.59	0.06	0.21	0.31
Memory	0.35	3.45	2.99	0.26	1.95	1.68
Time to Form Z_{SVS}	1.91			1.02		
LU Time	0.20			0.12		
Solution Time	$1.59e^{-04}$			$1.02e^{-04}$		
Total Time	**3.18**			**1.72**		

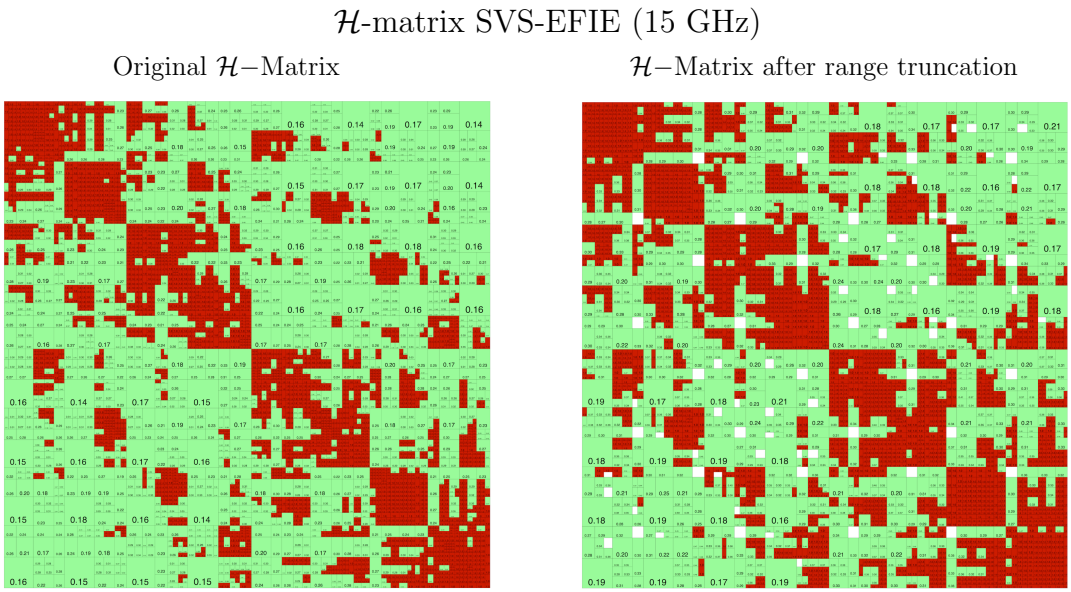


Figure 5.5: Surface-to-Surface $\mathcal{H}-$Matrix structures before and after the range truncation due to the skin-effect of the Lead sphere produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ obtained by the $\mathcal{H}$-matrix accelerated SVS-EFIE solver at $15\,\mathrm{GHz}$, radius of the sphere $R = 10\mu m$, and truncation distance $\mathcal{D}_T = R$.

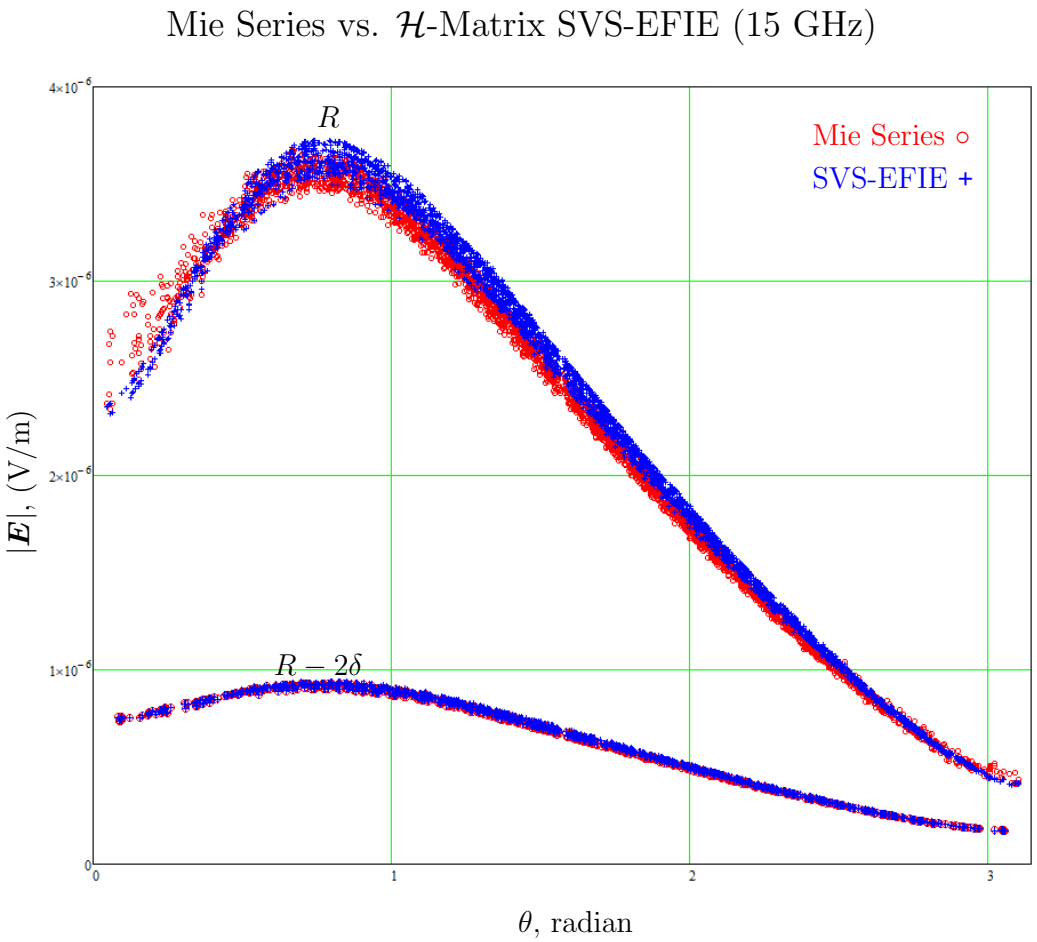


Figure 5.6: Magnitude of the electric field in Lead sphere of radius $10\mu m$ obtained via $\mathcal{H}-$Matrices accelerated SVS-EFIE due to radial electric dipole at 15 GHz as a function of the polar angle θ at the centroids of the tetrahedral elements nearest to the outer radius R of the extracted sphere (figure (5.7)) and nearest to the inner radius $r = R - 2\delta$ of the volume removed from the sphere due to skin-effect attenuation.

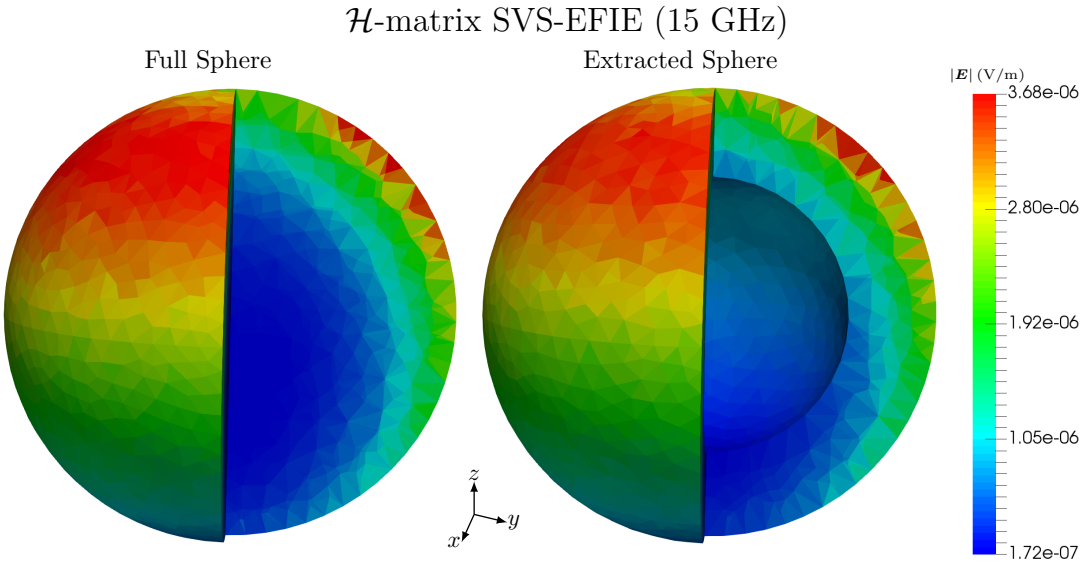


Figure 5.7: Magnitude of the total electric field inside Lead (Pb) sphere produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ obtained by the $\mathcal{H}$-matrix accelerated SVS-EFIE solver at 15 GHz

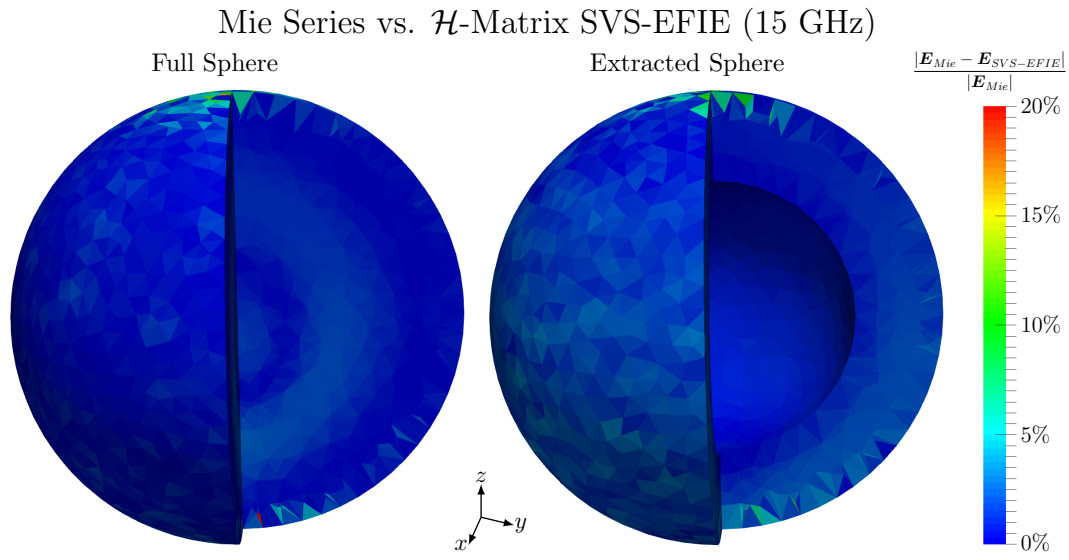


Figure 5.8: The relative error distribution of the total electric field inside extracted Lead (Pb) sphere, produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ at 15 GHz, in Mie series solution against SVS-EFIE.

5.2.3 Case 3: 20 GHz

Table 5.7: Set-up parameteric values for 20 GHz.

ADDITIONAL SET-UP PARAMETERS		
Skin Depth	1.67μm	
	Full Sphere	**Extracted Sphere**
No of 2-D Elements	4, 220	4, 220
No of 3-D Elements	28, 431	21, 664
No of RWG Basis Functions	6, 330	6, 330
No of Degree of Freedom	91, 623	71, 322

Table 5.8: SAR results obtained at 20 GHz.

Specific Absorption Rate (SAR) Results	
SAR of the Mie serie solution $\mathtt{SAR}_{Mie}$	$1.98e^{-10}$
SAR of the full sphere, $\mathtt{SAR}_{FullSphere}$	$1.91e^{-10}$
SAR of the extracted sphere, $\mathtt{SAR}_{ExtractedSphere}$	$1.77e^{-10}$
$\frac{\lvert\mathtt{SAR}_{Mie} - \mathtt{SAR}_{FullSphere}\rvert}{\lvert\mathtt{SAR}_{Mie}\rvert}$	03%
$\frac{\lvert\mathtt{SAR}_{Mie} - \mathtt{SAR}_{ExtractedSphere}\rvert}{\lvert\mathtt{SAR}_{Mie}\rvert}$	10%
$\frac{\lvert\mathtt{SAR}_{FullSphere} - \mathtt{SAR}_{ExtractedSphere}\rvert}{\lvert\mathtt{SAR}_{FullSphere}\rvert}$	07%

Table 5.9: Time and memory usages at 20 GHz.

RESULTS (Time: hrs, Memory: GB)						
	Full Sphere			**Extracted Sphere**		
	$Z_{\epsilon}^{\partial V,\partial V}$	$Z_{\epsilon}^{V,\partial V}$	$Z_{0}^{\partial V,V}$	$Z_{\epsilon}^{\partial V,\partial V}$	$Z_{\epsilon}^{V,\partial V}$	$Z_{0}^{\partial V,V}$
Filling Time	0.09	0.49	0.75	0.07	0.27	0.57
Memory	0.37	4.14	3.54	0.32	2.64	2.46
Time to Form Z_{SVS}	2.31			1.52		
LU Time	0.24			0.26		
Solution Time	$2.41e^{-04}$			$1.75e^{-04}$		
Total Time	**3.38**			**2.70**		

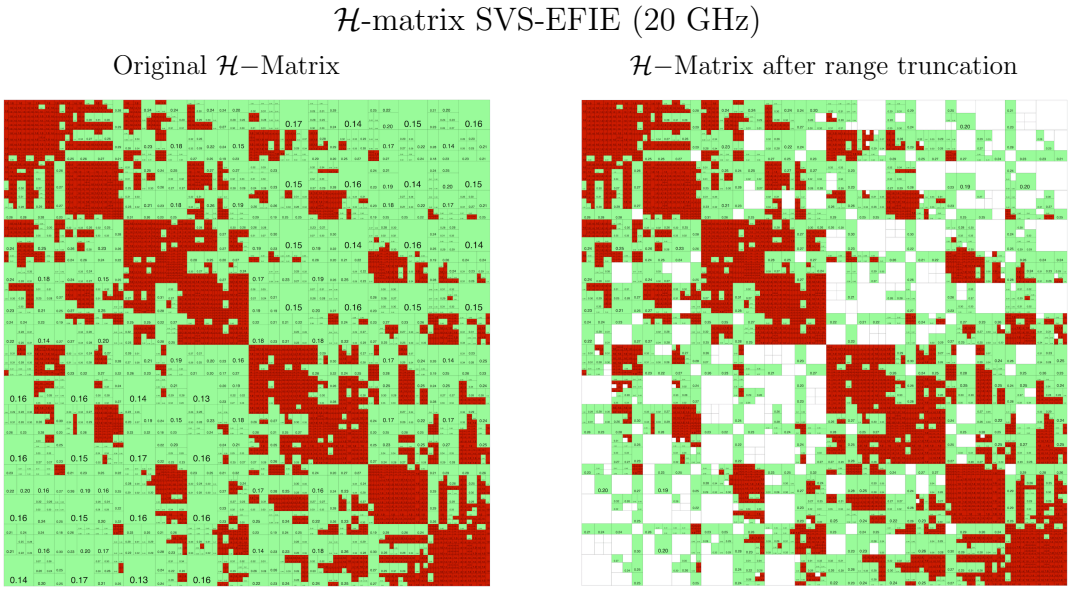


Figure 5.9: Surface-to-Surface $\mathcal{H}-$Matrix structures before and after the range truncation due to the skin-effect of the Lead sphere produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ obtained by the $\mathcal{H}$-matrix accelerated SVS-EFIE solver at $20\,\mathrm{GHz}$, radius of the sphere $R = 10\mu m$, and truncation distance $\mathcal{D}_T = R$.

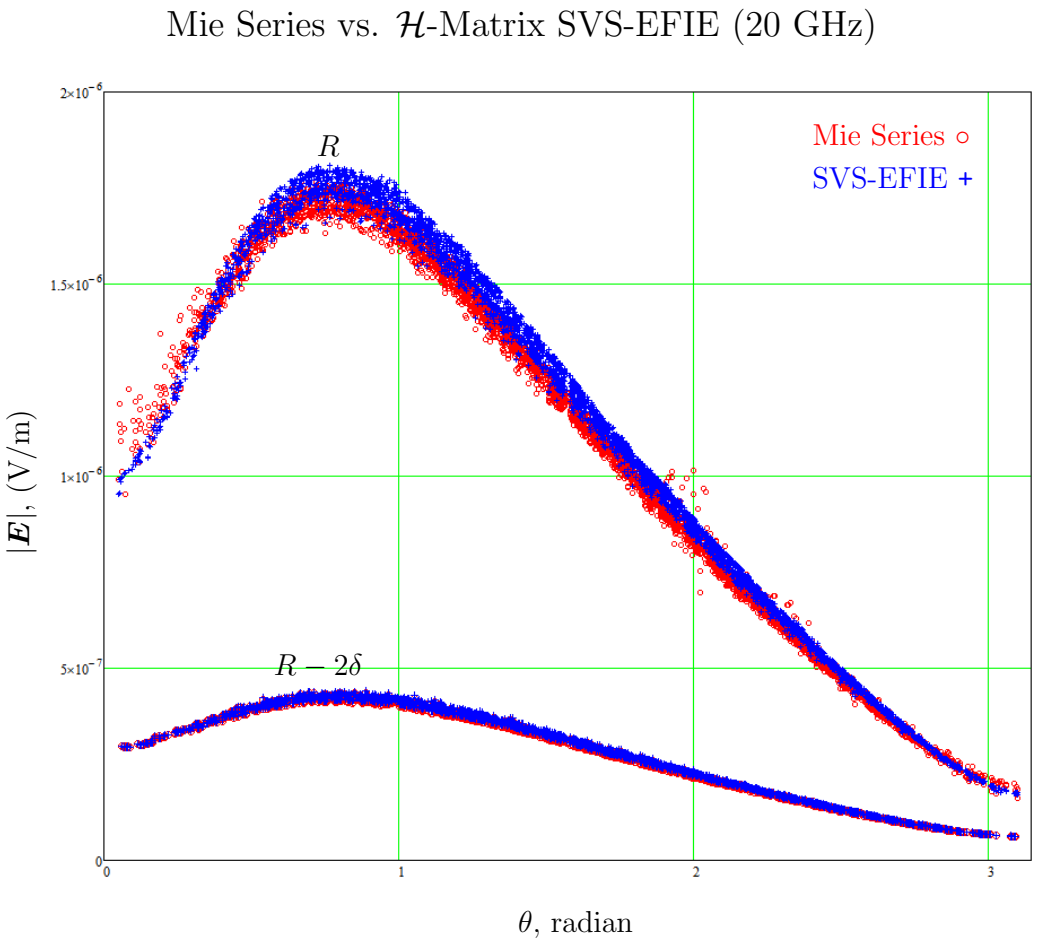


Figure 5.10: Magnitude of the electric field in Lead sphere of radius $10\mu m$ obtained via $\mathcal{H}-$Matrices accelerated SVS-EFIE due to radial electric dipole at 20 GHz as a function of the polar angle θ at the centroids of the tetrahedral elements nearest to the outer radius R of the extracted sphere (figure (5.11)) and nearest to the inner radius $r = R - 2\delta$ of the volume removed from the sphere due to skin-effect attenuation.

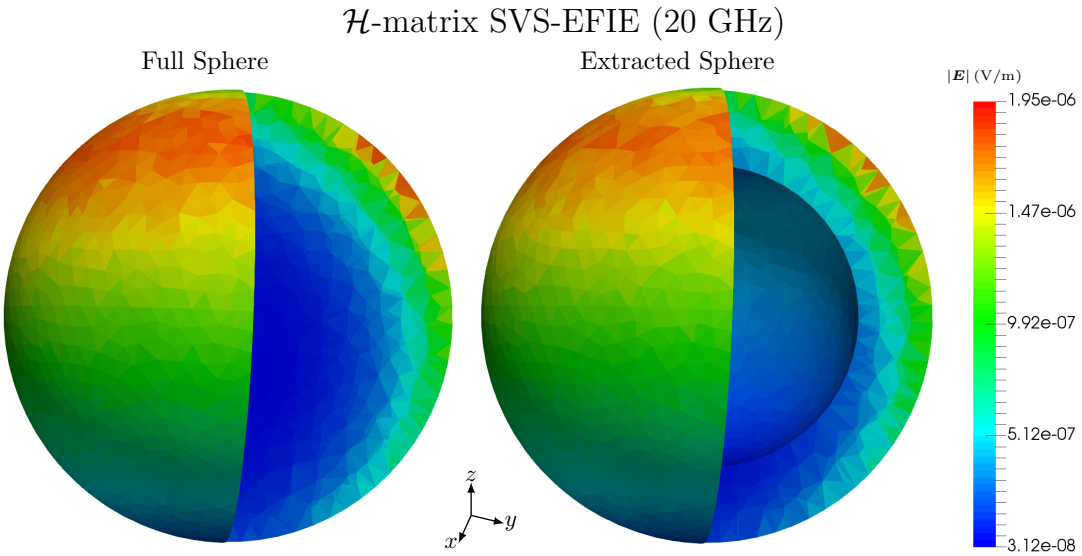


Figure 5.11: Magnitude of the total electric field inside Lead (Pb) sphere produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ obtained by the $\mathcal{H}$-matrix accelerated SVS-EFIE solver at 20 GHz

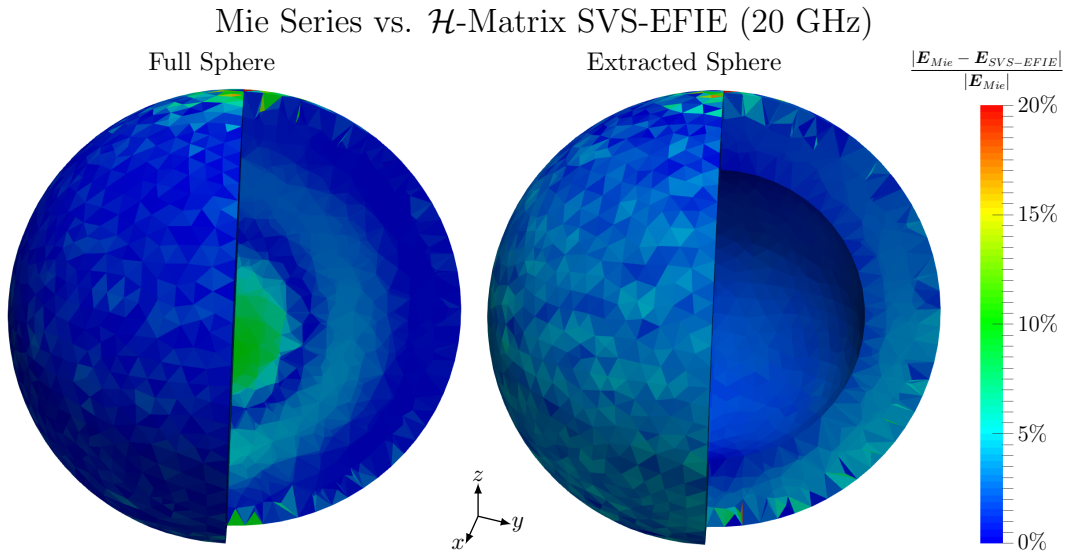


Figure 5.12: The relative error distribution of the total electric field inside extracted Lead (Pb) sphere, produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ at 20 GHz, in Mie series solution against SVS-EFIE.

5.2.4 Case 4: 25 GHz

Table 5.10: Set-up parameteric values for 25 GHz.

ADDITIONAL SET-UP PARAMETERS		
Skin Depth	1.49μm	
	Full Sphere	**Extracted Sphere**
No of 2-D Elements	5, 754	5, 754
No of 3-D Elements	46, 084	29, 278
No of RWG Basis Functions	8, 631	8, 631
No of Degree of Freedom	146, 883	96, 465

Table 5.11: SAR results obtained at 25 GHz.

Specific Absorption Rate (SAR) Results	
SAR of the Mie serie solution $\mathtt{SAR}_{Mie}$	$5.78e^{-11}$
SAR of the full sphere, $\mathtt{SAR}_{FullSphere}$	$5.60e^{-11}$
SAR of the extracted sphere, $\mathtt{SAR}_{ExtractedSphere}$	$5.15e^{-11}$
$\frac{\lvert\mathtt{SAR}_{Mie} - \mathtt{SAR}_{FullSphere}\rvert}{\lvert\mathtt{SAR}_{Mie}\rvert}$	03%
$\frac{\lvert\mathtt{SAR}_{Mie} - \mathtt{SAR}_{ExtractedSphere}\rvert}{\lvert\mathtt{SAR}_{Mie}\rvert}$	11%
$\frac{\lvert\mathtt{SAR}_{FullSphere} - \mathtt{SAR}_{ExtractedSphere}\rvert}{\lvert\mathtt{SAR}_{FullSphere}\rvert}$	08%

Table 5.12: Time and memory usages at 25 GHz.

RESULTS (Time: hrs, Memory: GB)						
	Full Sphere			**Extracted Sphere**		
	$Z_{\epsilon}^{\partial V,\partial V}$	$Z_{\epsilon}^{V,\partial V}$	$Z_{0}^{\partial V,V}$	$Z_{\epsilon}^{\partial V,\partial V}$	$Z_{\epsilon}^{V,\partial V}$	$Z_{0}^{\partial V,V}$
Filling Time	0.16	1.33	2.07	0.17	0.78	1.16
Memory	0.57	7.51	6.55	0.56	4.26	3.72
Time to Form Z_{SVS}	5.33			3.33		
LU Time	0.60			0.70		
Solution Time	$3.25e^{-04}$			$3.50e^{-04}$		
Total Time	**9.51**			**6.15**		

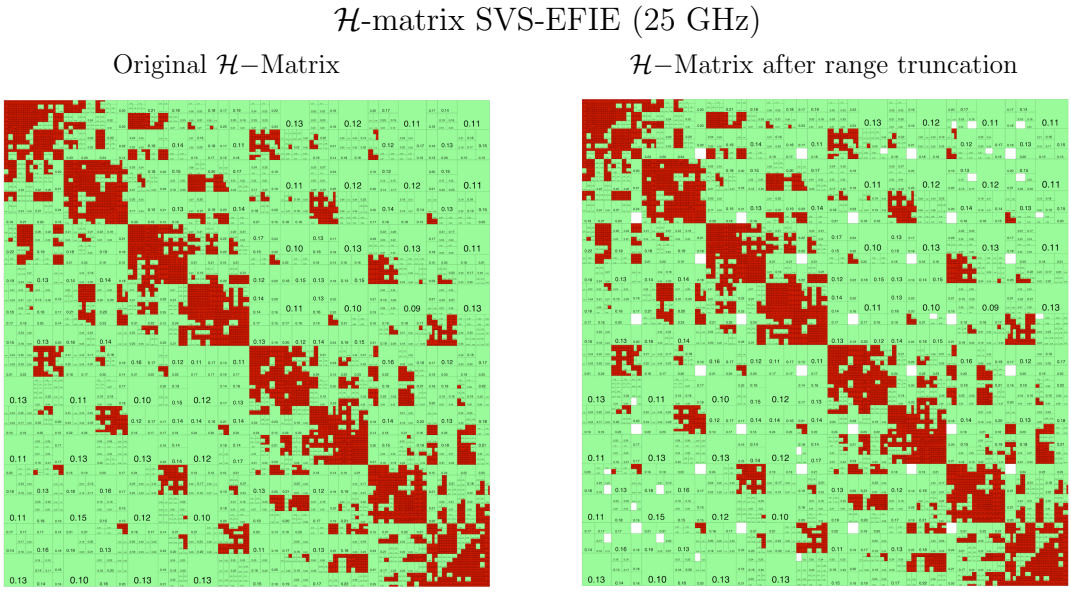


Figure 5.13: Surface-to-Surface $\mathcal{H}-$Matrix structures before and after the range truncation due to the skin-effect of the Lead sphere produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ obtained by the $\mathcal{H}$-matrix accelerated SVS-EFIE solver at 25 GHz, radius of the sphere $R = 10\mu m$, and truncation distance $\mathcal{D}_T = R$.

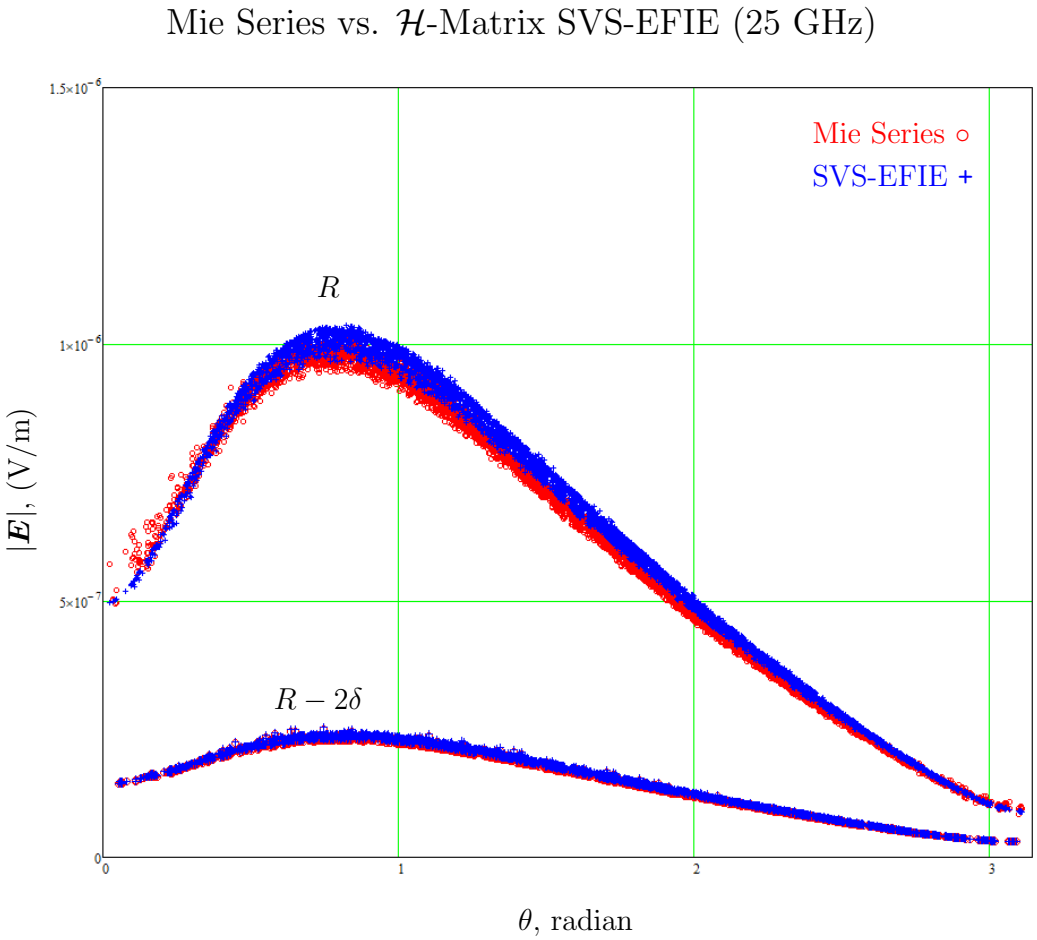


Figure 5.14: Magnitude of the electric field in Lead sphere of radius $10\mu m$ obtained via $\mathcal{H}-$Matrices accelerated SVS-EFIE due to radial electric dipole at 25 GHz as a function of the polar angle θ at the centroids of the tetrahedral elements nearest to the outer radius R of the extracted sphere (figure (5.15)) and nearest to the inner radius $r = R - 2\delta$ of the volume removed from the sphere due to skin-effect attenuation.

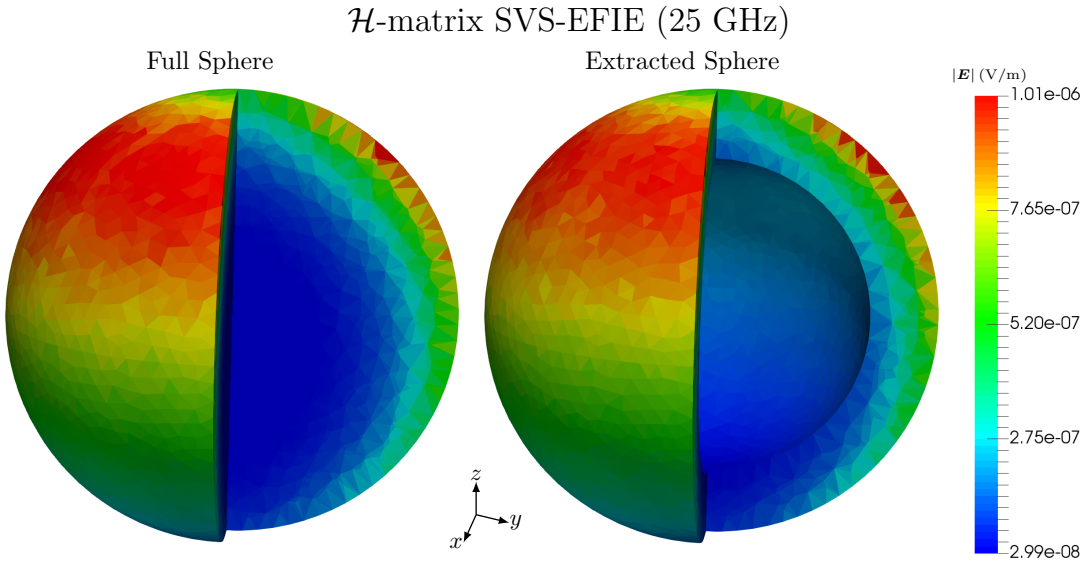


Figure 5.15: Magnitude of the total electric field inside Lead (Pb) sphere produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ obtained by the $\mathcal{H}$-matrix accelerated SVS-EFIE solver at 25 GHz

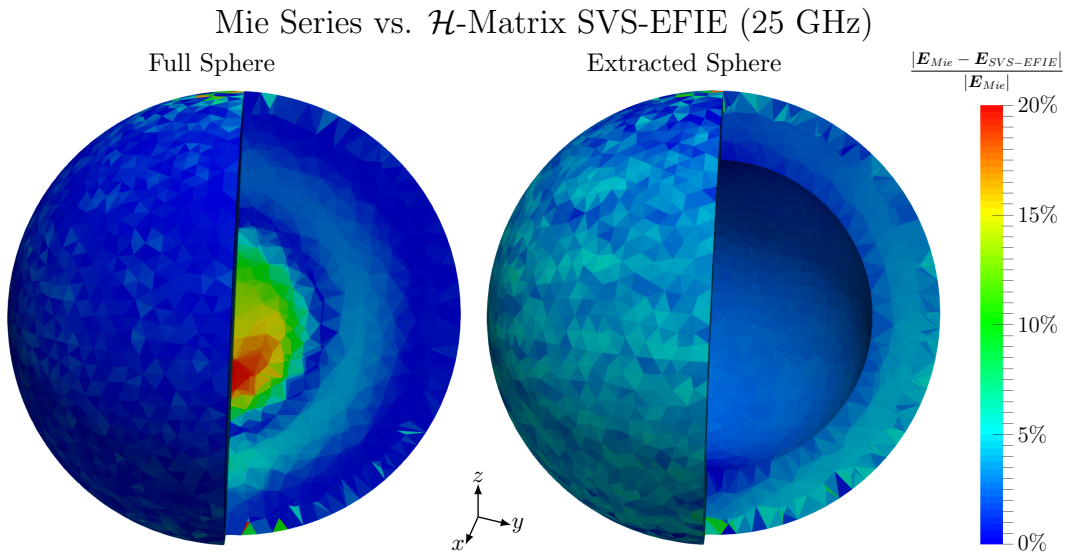


Figure 5.16: The relative error distribution of the total electric field inside extracted Lead (Pb) sphere, produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ at 25 GHz, in Mie series solution against SVS-EFIE.

5.2.5 Case 5: 30 GHz

Table 5.13: Set-up parameteric values for 30 GHz.

ADDITIONAL SET-UP PARAMETERS		
Skin Depth	1.36μm	
	Full Sphere	**Extracted Sphere**
No of 2-D Elements	6, 884	6, 884
No of 3-D Elements	56, 279	35, 731
No of RWG Basis Functions	10, 326	10, 326
No of Degree of Freedom	179, 163	117, 519

Table 5.14: SAR results obtained at 30 GHz.

Specific Absorption Rate (SAR) Results	
SAR of the Mie serie solution $\mathtt{SAR}_{Mie}$	$2.11e^{-11}$
SAR of the full sphere, $\mathtt{SAR}_{FullSphere}$	$2.05e^{-11}$
SAR of the extracted sphere, $\mathtt{SAR}_{ExtractedSphere}$	$1.87e^{-11}$
$\frac{\lvert\mathtt{SAR}_{Mie} - \mathtt{SAR}_{FullSphere}\rvert}{\lvert\mathtt{SAR}_{Mie}\rvert}$	03%
$\frac{\lvert\mathtt{SAR}_{Mie} - \mathtt{SAR}_{ExtractedSphere}\rvert}{\lvert\mathtt{SAR}_{Mie}\rvert}$	11%
$\frac{\lvert\mathtt{SAR}_{FullSphere} - \mathtt{SAR}_{ExtractedSphere}\rvert}{\lvert\mathtt{SAR}_{FullSphere}\rvert}$	09%

Table 5.15: Time and memory usages at 30 GHz.

RESULTS (Time: hrs, Memory: GB)						
	Full Sphere			**Extracted Sphere**		
	$Z_{\epsilon}^{\partial V,\partial V}$	$Z_{\epsilon}^{V,\partial V}$	$Z_{0}^{\partial V,V}$	$Z_{\epsilon}^{\partial V,\partial V}$	$Z_{\epsilon}^{V,\partial V}$	$Z_{0}^{\partial V,V}$
Filling Time	0.24	2.06	2.92	0.24	1.21	1.80
Memory	0.72	9.94	8.97	0.71	5.46	4.99
Time to Form Z_{SVS}	8.25			5.19		
LU Time	1.11			1.06		
Solution Time	$5.22e^{-04}$			$4.89e^{-04}$		
Total Time	**14.60**			**9.51**		

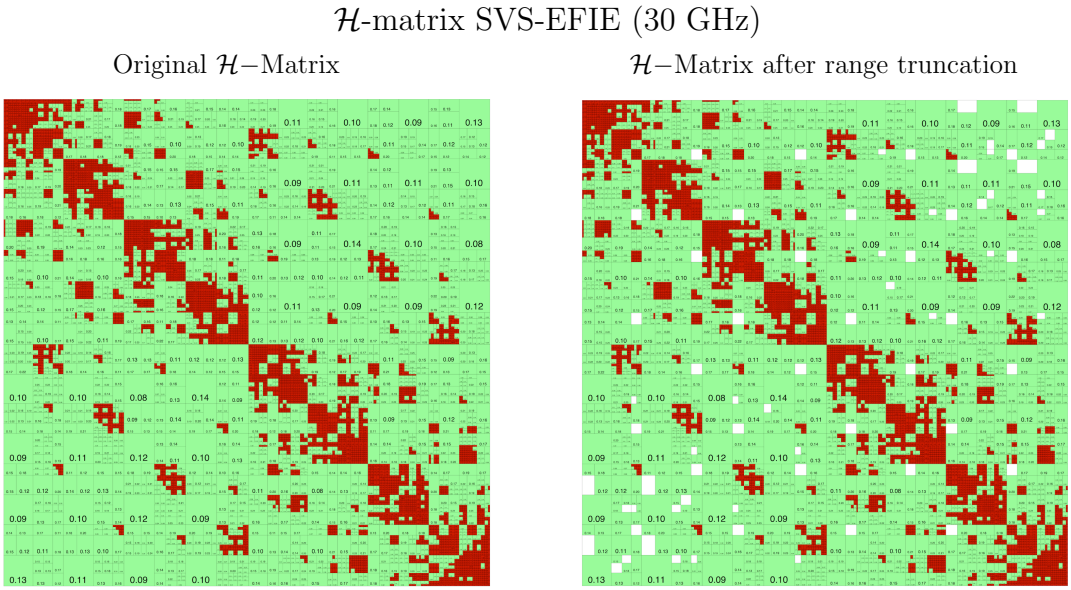


Figure 5.17: Surface-to-Surface $\mathcal{H}-$Matrix structures before and after the range truncation due to the skin-effect of the Lead sphere produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ obtained by the $\mathcal{H}$-matrix accelerated SVS-EFIE solver at 30 GHz, radius of the sphere $R = 10\mu m$, and truncation distance $\mathcal{D}_T = R$.

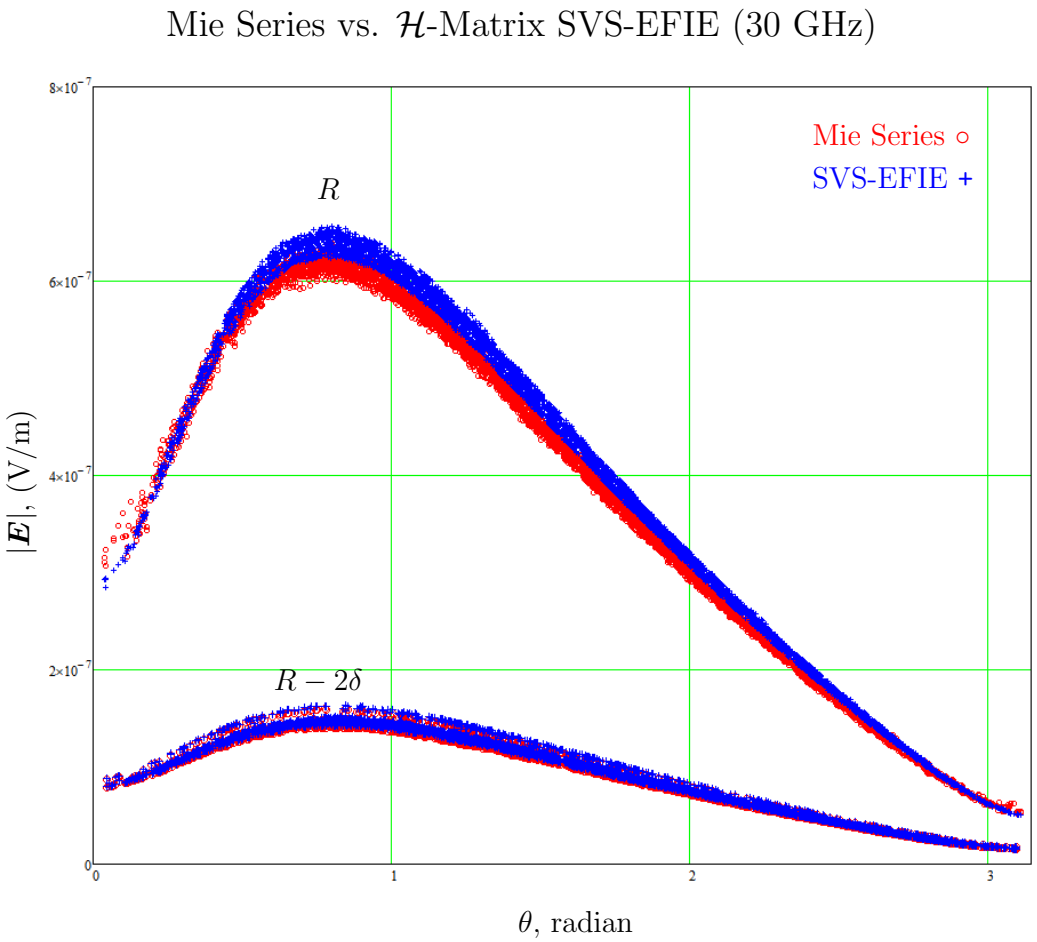


Figure 5.18: Magnitude of the electric field in Lead sphere of radius $10\mu m$ obtained via $\mathcal{H}-$Matrices accelerated SVS-EFIE due to radial electric dipole at 30 GHz as a function of the polar angle θ at the centroids of the tetrahedral elements nearest to the outer radius R of the extracted sphere (figure (5.19)) and nearest to the inner radius $r = R - 2\delta$ of the volume removed from the sphere due to skin-effect attenuation.

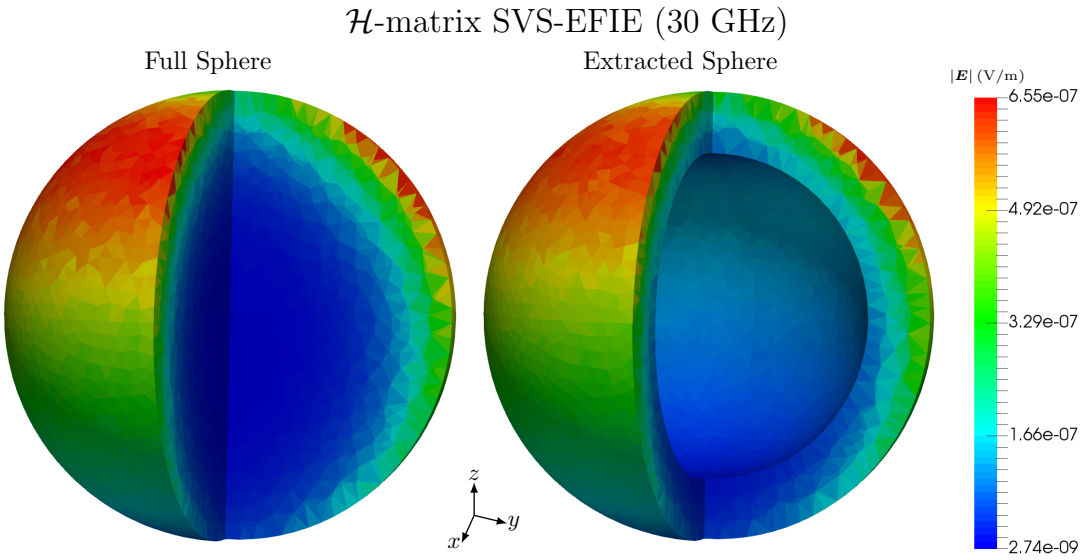


Figure 5.19: Magnitude of the total electric field inside Lead (Pb) sphere produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ obtained by the $\mathcal{H}$-matrix accelerated SVS-EFIE solver at 30 GHz

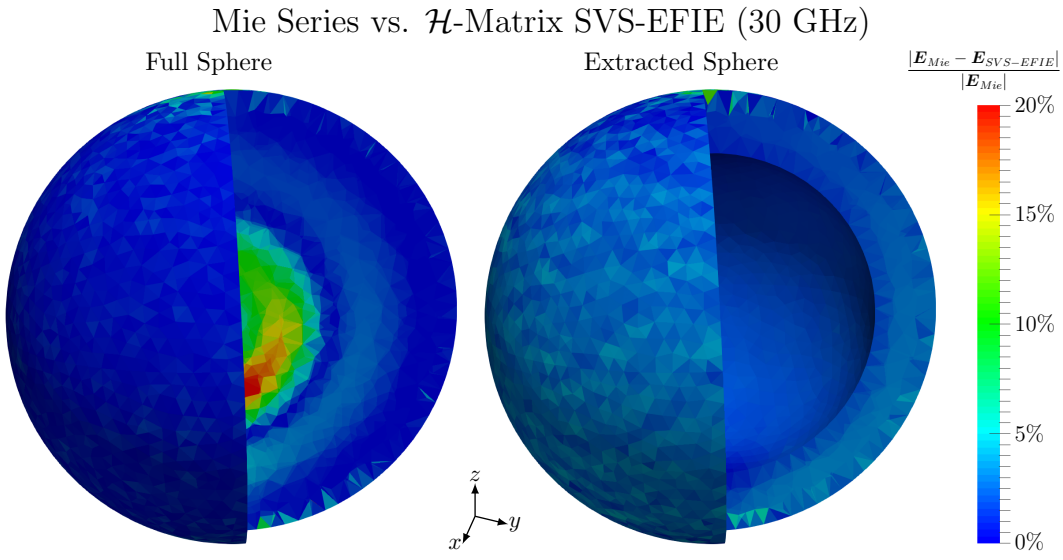


Figure 5.20: The relative error distribution of the total electric field inside extracted Lead (Pb) sphere, produced by a z-directed electric dipole situated at $x' = 0\,\mathrm{m}$, $y' = 0\,\mathrm{m}$, $z' = 0.00003\,\mathrm{m}$ at 30 GHz, in Mie series solution against SVS-EFIE.

5.2.6 Time and Memory Complexities

The time and memory requirements for the frequency range are plotted in order to obtain the big "O" complexity behaviour of the introduced techniques. Therefore, the memory requirement for storing Surface-to-Surface supermatrix is plotted against the number of RWG basis functions P (which is also the number of the unknowns) as shown in Figure (5.21)(a), while Figure (5.21)(b) shows the time complexity as a plot of time taken to fill the Surface-to-Surface supermatrix against the number of unknowns (RWG basis functions P).

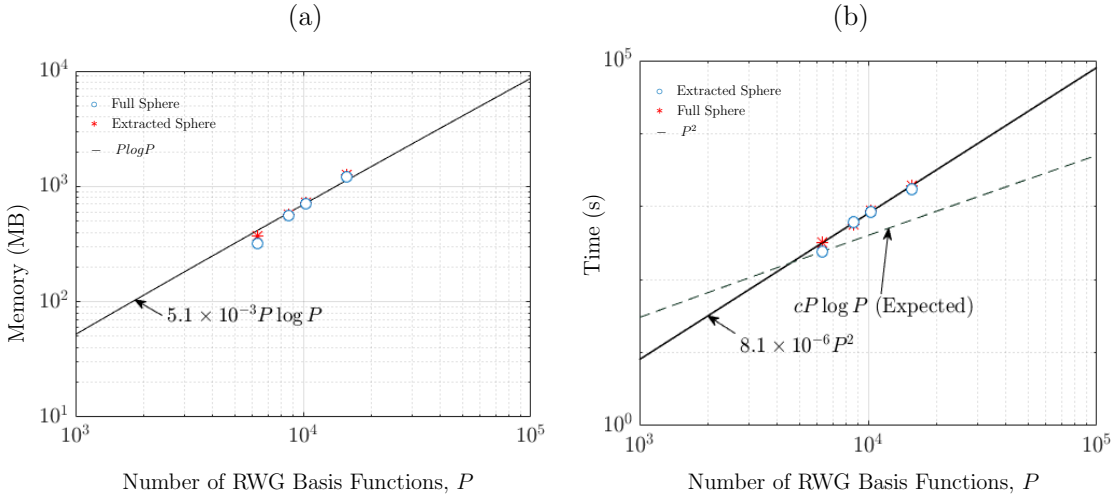


Figure 5.21: Complexity plot of $Z_{\epsilon}^{\partial V, \partial V}$ supermatrix (a) Memory complexity (b) Time complexity.

Similarly, Figure (5.22)(a) contains the plot of the memory requirement for storing the Surface-to-Volume supermatrix versus the number of RWG basis functions P. This, however, shows the memory complexity of Surface-to-Volume supermatrix. The time complexity of filling the Surface-to-Volume supermatrix is depicted in Fig-

ure (5.22)(b) as a plot of the time taken to fill the supermatrix against the number of RWG basis functions P.

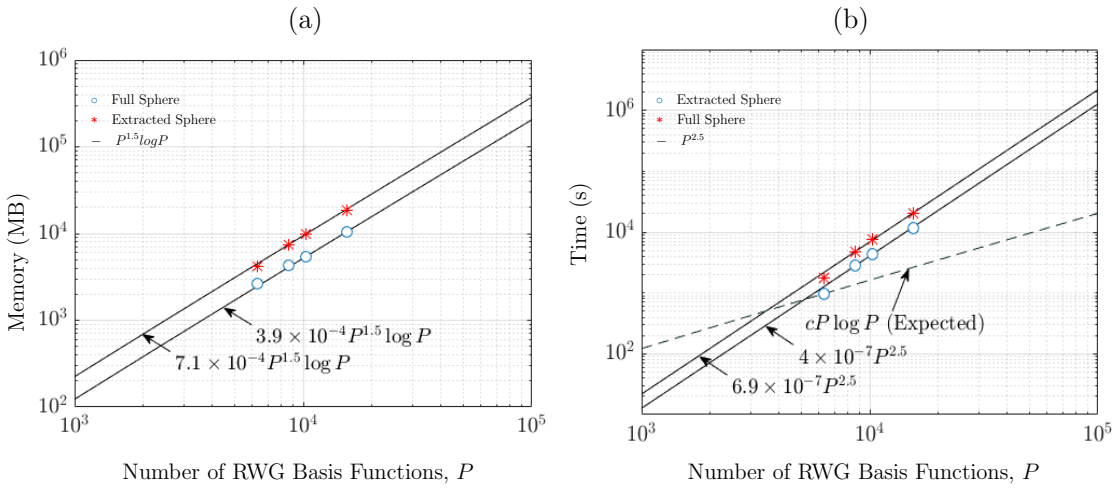


Figure 5.22: Complexity plot of $Z_{\epsilon}^{V,\partial V}$ supermatrix (a) Memory complexity (b) Time complexity.

Additionally, Figure (5.23)(a) is the plot of the memory used in storing the Volume-to-Surface supermatrix versus the number of unknowns P. This, however, shows the memory complexity of the Volume-to-Surface supermatrix. The time complexity of filling the Volume-to-Surface supermatrix is depicted in Figure (5.23)(b) as a plot of the time taken to fill the supermatrix versus the number of unknowns P.

Finally, the time complexity behaviour of forming the Z_{SVS} supermatrix is shown in Figure (5.24)(a) as a plot of time taken to form the supermatrix against the number of RWG basis functions P, while in Figure (5.24)(b) the overall time required to obtain a solution is plotted against the number of RWG basis functions P.

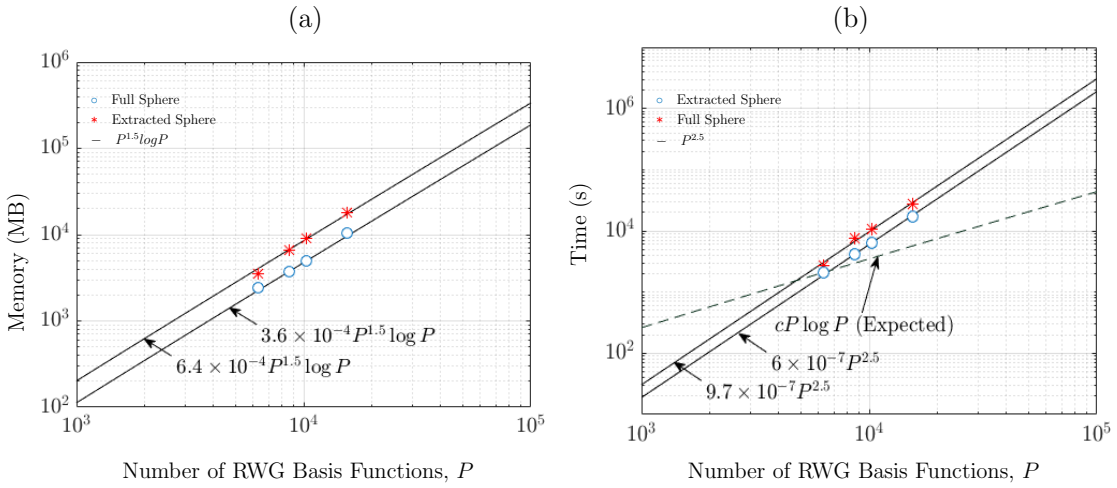


Figure 5.23: Complexity plot of $Z_0^{\partial V,V}$ supermatrix (a) Memory complexity (b) Time complexity.

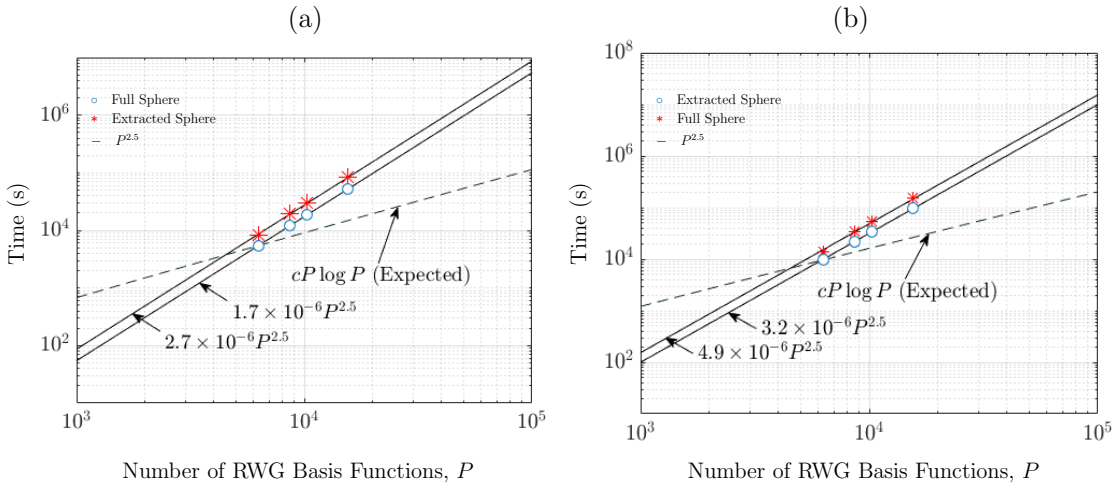


Figure 5.24: Time complexity plot for (a) forming the Z_{SVS} supermatrix (b) total time taken to obtain a solution.

5.3 Descriptive Analysis and Tests of Hypothesis

It is observed from Tables 5.1, 5.4, 5.7, 5.10 and 5.13 that the skin-depth δ decreases with increase in frequency. The skin-depth δ then dictates the density of

the mesh elements. These tables also show that the extraction greatly reduces the number of volumetric elements (tetrahedral) generated with increase in frequency - courtesy of the skin-depth δ.

Similarly, from Tables 5.2, 5.5, 5.8, 5.11 and 5.14, it can be observed that the relative error in the Mie series computed SAR for full sphere with respect to the SVS-EFIE computed SAR for the same full sphere stays roughly at 3%. The relative error in the Mie series computed SAR for the full sphere with respect to the SVS-EFIE computed SAR for the extracted sphere stays roughly at 10%. However, considering SVS-EFIE computed SAR for extracted sphere relative to that of SVS-EFIE computed SAR for the full sphere reveal relative error of roughly 7%.

Furthermore, Figures (5.1), (5.5), (5.9), (5.13) and (5.17) depict the effect of the range truncation on the structure of the supermatrices where the red blocks represent the $fullmatrix$ blocks, the green blocks denote the $rkmatrix$ blocks and white blocks are the empty blocks representing the skipped blocks that fall beyond the truncation distant $\mathcal{D}_T$. The effect of the white blocks is the reduction in the amount of computational tasks and time because those blocks are never computed in the first place.

On the same note, Figures (5.3), (5.7), (5.11), (5.15) and (5.19) show that the volumetric current distribution of the extracted spheres match the corresponding full spheres. Figures (5.4), (5.8), (5.12), (5.16) and (5.20) indicate that the solutions obtained with the new techniques on the extracted sphere compared to that of the

full sphere have the same range of relative error when these two results are compared with the Mie series solution. The maximum relative error in the solutions remains roughly at 20%. Likewise, Figures (5.2), (5.6), (5.10), (5.14) and (5.18) also confirm the accuracy of the solution because, the electric current computed from the extracted spheres match the equivalent Mie series counterparts.

Moreover, Tables 5.3, 5.6, 5.9, 5.12 and 5.15 show that the memory and time requirements in the cases involving the extracted sphere is much less compared to the corresponding full sphere. However, the time complexities of each of the supermatrices are shown in Figures (5.21) to (5.23). Figure (5.21)(a) clearly shows that the memory complexity of Surface-to-Surface supermatrix scales as $\mathcal{O}(kPlogP)$ for full sphere while it scales as $\mathcal{O}(qPlogP)$ for the extracted sphere where constant $q < k$. Figure (5.21)(b) shows that the time complexity for filling Surface-to-Surface supermatrix scales as $\mathcal{O}(kP^2)$ for full sphere while for the extracted sphere it scales as $\mathcal{O}(qP^2)$ where constant $q < k$. Figure (5.22)(a) clearly shows that the memory complexity of Surface-to-Volume supermatrix scales as $\mathcal{O}(kP^{1.5}logP)$ for full sphere while it scales as $\mathcal{O}(qP^{1.5}logP)$ for the extracted sphere where constant $q < k$. Figure (5.22)(b) shows that the time complexity for filling Surface-to-Volume supermatrix scales as $\mathcal{O}(kP^{2.5})$ for full sphere while for the extracted sphere it scales as $\mathcal{O}(qP^{2.5})$ where constant $q < k$. Consequently, the same behaviour in memory and time complexity can be observed for Volume-to-Surface supermatrix as plotted in Figure (5.23). Observing Figure (5.24) reveals the time complexity scaling for forming Z_{SVS} and the

total time to obtain a solution as $kP^{2.5}$ for the full sphere and $qP^{2.5}$ for the extracted sphere where $q < k$. It is important to note that the choice of SVD compression is responsible for the time complexities obtained in the presented cases. However, it is expected that the time complexities for filling the Surface-Surface supermatrix, Surface-Volume supermatrix, Volume-Surface supermatrix, time to form the Z_{SVS} supermatrix, and the total time to obtain a solution scale as $PlogP$ (if the ACA compression algorithm is used) as illustrated in Figures 5.21(b), 5.22(b), 5.23(b), and 5.24.

From the foregoing, it can be observed that the average relative error in the computed SAR for the test case showed that only 7% of the volumetric field has been affected as a result of volumetric mesh elements extraction. This answers the research question 1. The number of the computational steps involved in obtaining a solution have been shown to reduce due to the fact that some of the computational steps are skipped and a certain portion of the mesh volume was removed. The reduction in the number of computational tasks and memory requirements answers the research question 2. Despite the application of the interaction range truncation and volumetric extraction the maximum relative error in the solution remain the same for both the full sphere case and the extracted sphere equivalent.

Chapter 6

Conclusions

6.1 Overview and Summary

This research work proposed reduction techniques that could be employed to reduce the computational complexities associated with obtaining the solution of a scattering problem on 3-D metal object via the SVS-EFIE formulation. It, explicitly, demonstrates new techniques that could be employed in dealing with the bottlenecks associated with the computation of the SVS-EFIE operators. It also validates the solutions obtained with Mie series analytical solutions. The new techniques exploit the skin-effect of PECs to reduce the complexity through volumetric mesh elements extraction and interactions range truncation governed by the metal's Green's function while the results obtained remained within a given error range. In this regard, the memory and time complexity has been shown to be greatly reduced with these new

methods.

6.2 Conclusions

This work has demonstrated the time and the memory reductions in solution of scattering problem obtained from $\mathcal{H}-$Matrices accelerated SVS-EFIE formulation. The time complexity of filling Surface-to-Surface has been shown to scale at $\mathcal{O}(P^2)$ (because of the choice of SVD over ACA) and memory complexity scales as $\mathcal{O}(PlogP)$ in line with the suggested theoretical estimate. Time and memory complexity of Surface-to-Volume scale as $\mathcal{O}(P^{2.5})$ and $\mathcal{O}(P^{1.5}logP)$ respectively. The same behaviour in time and memory complexities hold for the Volume-to-Surface operator. The error estimates also show that error level is the same for the solutions obtained from the application of these new methods and the ones without the new techniques.

6.3 Implications

The new techniques, employed in this work, reduce both the time and the memory requirements of the whole scattering problem. However, the effects of these new methods are the introduction of additional parameters which must be considered in order to obtain accurate solution. Ordinarily, in order to change the accuracy of the solution of $\mathcal{H}-$Matrices based solver, only the (ACA or SVD) compression tolerance

need to be adjusted. In contrast, these new methods introduce two (2) additional dependants - the truncation distance $\mathcal{D}_T$ and the interior radius r which dictates the extent of the extraction. This then implies that, keeping the ACA or SVD tolerance constant, the accuracy of the solution still depends on the $\mathcal{D}_T$ and r.

6.4 Suggestions for Future Research

All the test cases presented in this research work use SVD compression, though both the ACA and SVD compressions were implemented. ACA algorithm being a heuristic method is not guaranteed to compress all kind of matrices. This drawback of ACA manifests in this work as the ACA failed to compressed the majority of the admissible blocks and this facilitated the choice of SVD in this work. As a result of this, it is suggested that further research should be conducted on this in order to be able to obtain almost linear complexity in filling times of the supermatrices.

Bibliography

[1] Kush Agarwal and Yong-Xin Guo. Interaction of electromagnetic waves with humans in wearable and biomedical implant antennas. In *Electromagnetic Compatibility (APEMC), 2015 Asia-Pacific Symposium on*, pages 154–157. IEEE, 2015.

[2] Mario Bebendorf. Approximation of boundary element matrices. *Numerische Mathematik*, 86(4):565–589, 2000.

[3] Mario Bebendorf. Hierarchical matrices: a means to efficiently solve elliptic boundary value problems. 2007.

[4] Mario Bebendorf and Sergej Rjasanow. Adaptive low-rank approximation of collocation matrices. *Computing*, 70(1):1–24, 2003.

[5] Steffen Börm, Lars Grasedyck, and Wolfgang Hackbusch. Introduction to hierarchical matrices with applications. *Engineering analysis with boundary elements*, 27(5):405–422, 2003.

[6] Yaniv Brick, Vitaliy Lomakin, and Amir Boag. Fast direct solver for essentially convex scatterers using multilevel non-uniform grids. *IEEE Transactions on Antennas and Propagation*, 62(8):4314–4324, 2014.

[7] Weng Cho Chew. *Waves and fields in inhomogeneous media*. IEEE press, 1995.

[8] Raul Enriquez-Shibayama, Benjamin Lopez-Garcia, Luis Nathan Perez-Acosta, and Ana Sanchez-Granados. Design of very low distortion dense routings for skew compensation in high speed interconnects. *IEEE Electromagnetic Compatibility Magazine*, 5(4):100–103, 2016.

[9] Lars Ole Fichte. Interaction of biological tissue with electromagnetic waves in the rf range. In *Environmental Electromagnetics (CEEM), 2015 7th Asia-Pacific Conference on*, pages 10–10. IEEE, 2015.

[10] Ch Geuzaine and JF Remacle. A three-dimensional finite element mesh generator with built-in preand post-processing facilities. *URL: gmsh. info. Google Scholar*.

[11] Marie-Christine Gosselin, Günter Vermeeren, Sven Kuhn, Valpré Kellerman, Stefan Benkler, Tero MI Uusitupa, Wout Joseph, Azeddine Gati, Joe Wiart, Frans JC Meyer, et al. Estimation formulas for the specific absorption rate in humans exposed to base-station antennas. *IEEE transactions on electromagnetic compatibility*, 53(4):909–922, 2011.

[12] Lars Grasedyck and Wolfgang Hackbusch. Construction and arithmetics of h-matrices. *Computing*, 70(4):295–334, 2003.

[13] Steffen Börm; Lars Grasedyck; Wolfgang Hackbusch. Hierarchical matrices. Preprint, 2003.

[14] Wolfgang Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. part i: Introduction to $\mathcal{H}$-matrices. *Computing*, 62(2):89–108, 1999.

[15] Roger F Harrington. *Field computation by moment methods*. Wiley-IEEE Press, 1993.

[16] Mohammad Izadi. *Hierarchical matrix techniques on massively parallel computers*. PhD thesis, Leipzig University, 2012.

[17] Jian-Ming Jin. *Theory and computation of electromagnetic fields*. John Wiley & Sons, 2011.

[18] Barışcan Karaosmanoğlu and Özgür Ergül. Modified combined tangential formulation for stable and accurate analysis of plasmonic structures. In *Applied Computational Electromagnetics Society Symposium-Italy (ACES), 2017 International*, pages 1–2. IEEE, 2017.

[19] A Kishk and Lotfollah Shafai. Different formulations for numerical solution of single or multibodies of revolution with mixed boundary conditions. *IEEE Transactions on Antennas and Propagation*, 34(5):666–673, 1986.

[20] Farhad Sheikh Hosseini Lori, Anton Menshov, Reza Gholami, Jamiu Babatunde Mojolagbe, and Vladimir I Okhmatovski. Novel single-source surface integral equation for scattering problems by 3-d dielectric objects. *IEEE Transactions on Antennas and Propagation*, 66(2):797–807, 2018.

[21] Farhad Sheikh Hosseini Lori, Anton Menshov, and Vladimir I Okhmatovski. New vector single-source surface integral equation for scattering problems on dielectric objects in 2-d. *IEEE Transactions on Antennas and Propagation*, 65(7):3794–3799, 2017.

[22] Anton Menshov and Vladimir Okhmatovski. Method of moment solution of surface-volume-surface electric field integral equation for two-dimensional transmission lines of complex cross-sections. In *Signal and Power Integrity (SPI), 2012 IEEE 16th Workshop on*, pages 31–34. IEEE, 2012.

[23] Anton Menshov and Vladimir Okhmatovski. Novel surface integral equation formulation for accurate broadband rl extraction in transmission lines of arbitrary cross-section. In *Microwave Symposium Digest (MTT), 2012 IEEE MTT-S International*, pages 1–3. IEEE, 2012.

[24] Anton Menshov and Vladimir Okhmatovski. New single-source surface integral equations for scattering on penetrable cylinders and current flow modeling in 2-d conductors. *IEEE Transactions on Microwave Theory and Techniques*, 61(1):341–350, 2013.

[25] Anton Menshov and Vladimir I Okhmatovski. Surface–volume–surface electric field integral equation for magneto-quasi-static analysis of complex 3-d interconnects. *IEEE Transactions on Microwave Theory and Techniques*, 62(11):2563–2573, 2014.

[26] Eric Michielssen and Amir Boag. A multilevel matrix decomposition algorithm for analyzing scattering from large structures. *IEEE Transactions on Antennas and Propagation*, 44(8):1086–1093, 1996.

[27] Carl Müller. *Foundations of the mathematical theory of electromagnetic waves*, volume 155. Springer Science & Business Media, 2013.

[28] Utkarsh R Patel, Sean V Hum, and Piero Triverio. A magneto-quasi-static surface formulation to calculate the impedance of 3d interconnects with arbitrary cross-section. In *Signal and Power Integrity (SPI), 2017 IEEE 21st Workshop on*, pages 1–4. IEEE, 2017.

[29] Andrew F Peterson, Scott L Ray, and Raj Mittra. *Computational methods for electromagnetics*, volume 2. 1998.

[30] Zhi Guo Qian, Weng Cho Chew, and Roberto Suaya. Generalized impedance boundary condition for conductor modeling in surface integral equation. *IEEE Transactions on Microwave Theory and Techniques*, 55(11):2354–2364, 2007.

[31] Sadasiva Rao, D Wilton, and Allen Glisson. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Transactions on antennas and propagation*, 30(3):409–418, 1982.

[32] Amr ME Safwat, Islam A Eshrah, Tamer M Abuelfadl, and Hadia El-Hennawy. University research on antenna design and scattering problems in egypt. In

Microwave Conference (EuMC), 2014 44th European, pages 1924–1927. IEEE, 2014.

[33] John Shaeffer. Direct solve of electrically large integral equations for problem sizes to 1 m unknowns. *IEEE Transactions on Antennas and Propagation*, 56(8):2306–2313, 2008.

[34] Farhad Sheikh Hosseini Lori. *Novel single source integral equation for analysis of electromagnetic scattering by penetrable objects*. PhD thesis, University of Manitoba, 2017.

[35] Horst D Simon and Hongyuan Zha. Low-rank matrix approximation using the lanczos bidiagonalization process with applications. *SIAM Journal on Scientific Computing*, 21(6):2257–2274, 2000.

[36] David R Swatek. Investigation of single-source surface integral equations for electromagnetic wave scattering by dielectric bodies. 1999.

[37] Kezhong Zhao, Marinos N Vouvakis, and Jin-Fa Lee. The adaptive cross approximation algorithm for accelerated method of moments computations of emc problems. *IEEE transactions on electromagnetic compatibility*, 47(4):763–773, 2005.

[38] Yu Zhao and Junfa Mao. Equivalent surface impedance-based mixed potential integral equation accelerated by optimized h-matrix for 3-d interconnects. *IEEE Transactions on Microwave Theory and Techniques*, 2017.

[39] Yu Zhao, Min Tang, Shang Xiang, and Junfa Mao. $\mathcal{H}$-matrix accelerated contour integral method for modeling multiconductor transmission lines. *IEEE Transactions on Electromagnetic Compatibility*, 60(2):552–555, 2018.

[40] Shucheng Zheng, Anton Menshov, and Vladimir I Okhmatovski. New single-source surface integral equation for magneto-quasi-static characterization of transmission lines situated in multilayered media. *IEEE Transactions on Microwave Theory and Techniques*, 64(12):4341–4351, 2016.

A

Sphere Model

```
/* Sphere with option for volumetric elements extraction
   Parameters:
  sd => skin depth of the material
  s  => characteristics length of the outer sphere;
  f  => factor*sd determines where the removal will start, that is,
   inner radius
  rad => radius of the sphere
  extract => 0 | 1 value; It means 0 => No and 1 => Yes
   Author:
  Jamiu Babatunde Mojolagbe
  Electrical and Computer Engineering
  University of Manitoba, Winnipeg, MB, Canada
*/

DefineConstant[
  rad = {10.0e-06, Name "Sphere Parameters/Basic/Radius"}
  sd  = {2.4e-06, Name "Sphere Parameters/Basic/Skin Depth (SD)"}
  num = {2, Name "Sphere Parameters/Meshing/Elements per SD"}
  f   = {2.0, Name "Sphere Parameters/Extraction/Factor (Factor*SD Away)
   "}
  extract = {0, Choices{0="No", 1="Yes"}, Name "Sphere Parameters/
   Extraction/Extract", Highlight "Blue"}
  v_name = {6, Name "Sphere Parameters/Meshing/Physical Volume Name"}
  s_name = {1, Name "Sphere Parameters/Meshing/Physical Surface Name"}
];


Mesh.CharacteristicLengthFromCurvature=0.009;
s   = sd/num;
ss = s;
If (extract)
  rad_in = rad - (sd*f); // calculate inner radius
Else
  rad_in = 0.;
```

```
EndIf

Point(1) = {0, 0, 0, s};
Point(2) = {-rad, 0, 0, s};
Point(3) = {rad, 0, 0, s};
Point(4) = {0, -rad, 0, s};
Point(5) = {0, rad, 0, s};
Point(6) = {0, 0, -rad, s};
Point(7) = {0, 0, rad, s};

Circle(1) = {3, 1, 5};
Circle(2) = {5, 1, 2};
Circle(3) = {2, 1, 4};
Circle(4) = {4, 1, 3};
Circle(5) = {6, 1, 5};
Circle(6) = {5, 1, 7};
Circle(7) = {7, 1, 4};
Circle(8) = {4, 1, 6};
Circle(9) = {6, 1, 2};
Circle(10) = {2, 1, 7};
Circle(11) = {7, 1, 3};
Circle(12) = {3, 1, 6};

Line Loop(25) = {11, 1, 6};
Ruled Surface(26) = {25};
Line Loop(27) = {10, -6, 2};
Ruled Surface(28) = {27};
Line Loop(29) = {3, -7, -10};
Ruled Surface(30) = {29};
Line Loop(31) = {7, 4, -11};
Ruled Surface(32) = {31};
Line Loop(33) = {12, 5, -1};
Ruled Surface(34) = {33};
Line Loop(35) = {-5, -2, 9};
Ruled Surface(36) = {35};
Line Loop(37) = {-9, -3, -8};
Ruled Surface(38) = {37};
Line Loop(39) = {8, -12, -4};
Ruled Surface(40) = {39};


If(rad_in > 0)
  Point(8) = {-rad_in , 0, 0, ss};
  Point(9) = {rad_in , 0, 0, ss};
  Point(10) = {0, -rad_in , 0, ss};
  Point(11) = {0, rad_in , 0, ss};
  Point(12) = {0, 0, -rad_in , ss};
  Point(13) = {0, 0, rad_in , ss};

  Circle(13) = {9, 1, 11};
```

```
  Circle(14) = {11, 1, 8};
  Circle(15) = {8, 1, 10};
  Circle(16) = {10, 1, 9};
  Circle(17) = {12, 1, 11};
  Circle(18) = {11, 1, 13};
  Circle(19) = {13, 1, 10};
  Circle(20) = {10, 1, 12};
  Circle(21) = {12, 1, 8};
  Circle(22) = {8, 1, 13};
  Circle(23) = {13, 1, 9};
  Circle(24) = {9, 1, 12};

  Line Loop(41) = {23, 13, 18};
  Ruled Surface(42) = {41};
  Line Loop(43) = {22, -18, 14};
  Ruled Surface(44) = {43};
  Line Loop(45) = {15, -19, -22};
  Ruled Surface(46) = {45};
  Line Loop(47) = {19, 16, -23};
  Ruled Surface(48) = {47};
  Line Loop(49) = {24, 17, -13};
  Ruled Surface(50) = {49};
  Line Loop(51) = {-17, -14, 21};
  Ruled Surface(52) = {51};
  Line Loop(53) = {-21, -15, -20};
  Ruled Surface(54) = {53};
  Line Loop(55) = {20, -24, -16};
  Ruled Surface(56) = {55};

  Surface Loop(57)= {26,28,30,32,34,36,38,40,  42,44,46,48,50,52,54,56};
Else
  Surface Loop(57)= {26,28,30,32,34,36,38,40};
EndIf

Physical Surface(s_name) = {28, 26, 34, 36, 30, 38, 40, 32};
Volume(58) = {57};
Physical Volume(v_name) = {58};
```

B

Method getDistance()

```
/**
getDistance - Computes the distance between two clusters or sons of a supermatrix

@param A  Cluster of pcluster data type
@param B  Cluster of pcluster data type

@return distance Distance between the bounding boxes of A and B
*/
double getDistance(const pcluster &A, const pcluster &B) {
  double sigma = 0.;              // initialize sigma to hold sums
  unsigned d = A->d;              // Dimension of the bounding boxes

  for (unsigned i = 0; i < d; i++) {
    const double& aMin = A->bmin[i];     // A's bounding box minimum
    const double& aMax = A->bmax[i];     // A's bounding box maximum
    const double& bMin = B->bmin[i];     // B's bounding box minimum
    const double& bMax = B->bmax[i];     // B's bounding box maximum

    double abMax = std::abs(aMax - bMax);

    // check the spatial locations of the bounding box
    // and compute the difference
    if (aMin > bMin) {
      sigma += pow((bMax > aMin) ? std::min(abMax, bMax - aMin) : aMin - bMax, 2);
    }
    else {
      sigma += pow((aMax > bMin) ? std::min(abMax, aMax - bMin) : bMin - aMax, 2);
    }
  }
  return sqrt(sigma);
}
```