University of Kentucky
UKnowledge

University of Kentucky Doctoral Dissertations | Graduate School

2008

MODULAR FAST DIRECT ANALYSIS USING NON-RADIATING LOCAL-GLOBAL SOLUTION MODES

Xin Xu
University of Kentucky

Right click to open a feedback form in a new tab to let us know how this document benefits you.

Recommended Citation
Xu, Xin, "MODULAR FAST DIRECT ANALYSIS USING NON-RADIATING LOCAL-GLOBAL SOLUTION MODES" (2008). *University of Kentucky Doctoral Dissertations*. 690.
https://uknowledge.uky.edu/gradschool_diss/690

This Dissertation is brought to you for free and open access by the Graduate School at UKnowledge. It has been accepted for inclusion in University of Kentucky Doctoral Dissertations by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

ABSTRACT OF DISSERTATION

Xin Xu

The Graduate School

University of Kentucky

2008

MODULAR FAST DIRECT ANALYSIS USING NON-RADIATING LOCAL-GLOBAL SOLUTION MODES

ABSTRACT OF DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Engineering
at the University of Kentucky

By
Xin Xu
Lexington, Kentucky

Director: Dr. Robert J. Adams, Associate Professor of
Electrical and Computer Engineering
Lexington, Kentucky

2008
Copyright © Xin Xu 2008

ABSTRACT OF DISSERTATION

MODULAR FAST DIRECT ANALYSIS USING NON-RADIATING LOCAL-GLOBAL SOLUTION MODES

This dissertation proposes a modular fast direct (MFD) analysis method for a class of problems involving a large fixed platform region and a smaller, variable design region. A modular solution algorithm is obtained by first decomposing the problem geometry into platform and design regions. The two regions are effectively detached from one another using basic equivalence concepts. Equivalence principles allow the total system model to be constructed in terms of independent interaction modules associated with the platform and design regions. These modules include interactions with the equivalent surface that bounds the design region. This dissertation discusses how to analyze (fill and factor) each of these modules separately and how to subsequently compose the solution to the original system using the separately analyzed modules.

The focus of this effort is on surface integral equation formulations of electromagnetic scattering from conductors and dielectrics. In order to treat large problems, it is necessary to work with sparse representations of the underlying system matrix and other, related matrices. Fortunately, a number of such representations are available. In the following, we will primarily use the adaptive cross approximation (ACA) to fill the multilevel simply sparse method (MLSSM) representation of the system matrix. The MLSSM provides a sparse representation that is similar to the multilevel fast multipole method.

Solutions to the linear systems obtained using the modular analysis strategies described above are obtained using direct methods based on the local-global solution (LOGOS) method. In particular, the LOGOS factorization provides a data sparse factorization of the MLSSM representation of the system matrix. In addition, the LOGOS solver also provides an approximate sparse factorization of the inverse of the system matrix. The availability of the inverse eases the development of the MFD method. Because the behavior of the LOGOS factorization is critical to the development of the proposed MFD method, a significant part of this dissertation is devoted to providing additional analyses, improvements, and characterizations of LOGOS-based direct solution methods. These further developments of the LOGOS factorization algorithms and their application to the development of the MFD method comprise the most significant contributions of this dissertation.

Key Words: modular design; local-global solution; direct solver; reduced order model; electromagnetic

Xin Xu

Student's Signature

Jan 7, 2009

Date

MODULAR FAST DIRECT ANALYSIS USING NON-RADIATING LOCAL-GLOBAL SOLUTION MODES

By

Xin Xu

Robert J. Adams

Director of Dissertation

Yu Ming Zhang

Director of Graduate Studies

Jan 7, 2009

Date

RULES FOR THE USE OF DISSERTATIONS

Unpublished dissertations submitted for the Doctor's degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgments.

Extensive copying or publication of the dissertation in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

A library that borrows this dissertation for use by its patrons is expected to secure the signature of each user.

Name Date

_____

_____

_____

_____

_____

_____

_____

_____

DISSERTATION

Xin Xu

The Graduate School

University of Kentucky

2008

MODULAR FAST DIRECT ANALYSIS USING NON-RADIATING LOCAL-GLOBAL SOLUTION MODES

DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Engineering
at the University of Kentucky

By
Xin Xu
Lexington, Kentucky

Director: Dr. Robert J. Adams, Associate Professor of
Electrical and Computer Engineering
Lexington, Kentucky

2008

To Ching and Yan Zhe

ACKNOWLEDGMENTS

I would like to thank my advisor, Professor Robert J. Adams, for providing me the opportunity to study and work in his group. He is always supportive and positive in guiding me for solving problems during the process of research. He is a very enthusiastic researcher. He never reserves his opinions. His ability to produce inspiring new ideas in solving problems is impressive. We have frequent discussions on the progress of and problems in our research work. I learned a lot from the discussions. I am also inspired a lot by the discussions.

I am grateful to Professor Stephen D. Gedney and Professor Cai-Cheng Lu who generously allowed me to use their original Fortran codes in my works. They also provided patient and timely assistance in understanding the codes and the theories.

The members of my dissertation committee, Robert, J, Adams, Stephen D. Gedney, Cai-Cheng Lu, William T. Smith, Qiang Ye and D. Manivannan, have generously given their time and expertise to better my work. I thank for their effort and contribution.

I must acknowledge my friends and colleagues who assisted, advised and supported my research and writing over the years. Especially, I need to express my gratitude to Dr. Yuan Xu with whom I have worked closely on the LOGOS factorization. He introduced and taught me many details of LOGOS related formulation and programming. I thank Dr. Aiming Zhu, Dr. Chong Luo, Dr. Zhiyong Zeng, Ms. Jin Cheng, Mr. Jun Shik Choi, Mr. Stephen Maloney, Mr. Bo Zhao for numerous discussions and exchange of codes.

Table of Contents

List of Figures

List of Files

MFD_Dissertation_Xin.pdf (1.8MB)

Chapter 1. Introduction

1.1. Motivation

In many critical electromagnetic (EM) applications, it is important to design relatively small subsystems (such as an antenna) for optimal operation on a much larger existing platform (such as an aircraft). It is also sometimes desired to modify a small region of a large object in order to improve its electromagnetic performance as a whole. In most of these scenarios, only those structures in a small, well-defined design region are modified during each design cycle; the remainder of the object (platform region) does not change from one design iteration to another. The purpose of this dissertation is to provide a fast method for analyzing these types of problems.

1.2. Overview

Frequency domain computational EM (CEM) methods are very important and are widely used for simulating the EM properties of a wide range of systems and devices. Such methods typically involve solving linear matrix equations with N unknowns of the form

$$\mathbf{Z}\mathbf{x} = \mathbf{F}, \tag{1.1}$$

where $\mathbf{Z}$ is an $N \times N$ matrix, which is referred to as the system matrix; $\mathbf{F}$ contains the forcing vectors or excitation vectors, which is also referred to as the right hand side (RHS); $\mathbf{x}$ contains the unknowns to be solved for each forcing vector. Equation (1.1) can be obtained using either differentially based formulations, or integral equation (IE) based methods. In the following we will focus on surface integral equation (SIE) formulations of electromagnetic phenomena. A significant aspect of SIEs is that the resulting system matrix is full. For large systems, this is a significant factor in determining appropriate strategies for implementing and solving (1.1).

Equation (1.1) can be solved either iteratively or directly. Iterative methods [1] check the equality of (1.1) by providing a trial value of $\mathbf{x}$. Different iterative methods provide different ways to calculate the successive trial values. The efficiency of most iterative methods is governed by the efficiency of the matrix-vector multiply, $\mathbf{Z}\mathbf{x}$. Fast iterative

methods [2-14] provide efficient strategies for calculating the matrix-vector multiply which save both CPU time and memory. However, these methods often suffer from a lack of robustness. For many problems with ill-conditioned system matrices, iterative methods converge very slowly or even do not converge at all. Preconditioning can improve the condition of a system. However, to find a suitable preconditioner is non-trivial and, in many cases, expensive. Another drawback of fast iterative methods is the need to perform a full solution for each new RHS even though the system matrix is not changed. For these reasons, considerable effort has recently been devoted to developing efficient direct solvers. Direct methods can potentially provide significant advantages relative to iterative methods, including the ability to more rapidly solve multiple-RHS problems and the ability to efficiently combine with other techniques, such as the Schur complement method. Of course, iterative and direct methods are not mutually exclusive, and another advantage of efficient direct methods is their potential use as robust preconditioners for fast iterative solvers.

Direct methods obtain the inverse or a representation of the inverse of $\mathbf{Z}$. One of the basic ways to obtain the inverse of $\mathbf{Z}$ is to compute the LU factorization [15] of $\mathbf{Z}$, such that

$$\mathbf{Z} = \mathbf{L}\mathbf{U}, \tag{1.2}$$

$$\mathbf{Z}^{-1} = \mathbf{U}^{-1}\mathbf{L}^{-1}. \tag{1.3}$$

Generally, this process has a CPU-time complexity of $O\left(N^3\right)$ and a memory complexity of $O\left(N^2\right)$. If the system matrix is sparse, such as those obtained from the finite element method (FEM), various methods can be used to speed up the LU factorization [16, 17] by controlling the fill-in of the sparse matrices during factorization. We will refer to these methods as sparse LU factorization methods. Some of the latter methods have been included in publically available software libraries, such as SuperLU [18] and UMFPACK [19]. The sparsity pattern of the system matrix is an important factor that affects the efficiency of the algorithms used by these codes.

Some system matrices, such as those obtained from integral equations, are inherently dense. Sparse LU factorization cannot be applied directly. Fortunately, sparse approximate representations of the dense system matrices exist. Canning and Rogovin

(1998) [20] proposed an algebraic way of building a sparse representation of dense system matrices. Sparse LU factorization on top of these sparse representations of the system matrix can be used to develop fast direct solvers [20, 21]. Beside the sparse LU factorization, additional fast direct solvers that are based on sparse representations of system matrices have been developed recently [21-25].

The direct solver developed by Adams et. al. [25] is based on the concept of local-global solution (LOGOS) modes. In this case, a particular sparse representation of the system matrix is factored using the so-called LOGOS modes, which are derived from the system matrix. By specifying whether the LOGOS source modes have overlapping or non-overlapping support, LOGOS modes can be classified as either overlapping or non-overlapping. By specifying whether the LOGOS field modes radiate or do not radiate on to the structures outside a specified region, the LOGOS modes can be classified as radiating or non-radiating. The factorization using overlapping LOGOS modes has been shown to be asymptotically more efficient than that using the non-overlapping LOGOS modes for solving large problems. The non-radiating LOGOS modes are suitable for low frequency problems and the radiating LOGOS modes are suitable for high frequency problems.

The availability of the inverse of a system matrix makes the solution process versatile. Schur complement method [26] is an important method that can be used to accelerate the simulation of problems that involve combinations of variable structures and fixed structures [27, 28]. The acceleration is achieved because the system matrix for the fixed structure can be inverted once and reused for all combinations.

The work in this dissertation is based on the novel combination of these ideas. In particular, equivalence concepts are used to develop a modular analysis method, and the Schur complement is used to factor the resulting system for variable designs in the presence of a larger, fixed platform. LOGOS-based sparse direct solution algorithms are used in order to improve the efficiency of the underlying modular solver. In order to accomplish this, several significant improvements and analyses of the LOGOS solver are developed as part of this work.

1.3. Basic Idea

Most large systems are composed of smaller modules. Each module has its interface to connect to other modules. In practice, it is convenient and efficient to design and manufacture the modules separately with various constraints including those specified on the interfaces.

The development of an EM solver that provides similar modularity in an efficient manner is an attractive possibility because the unchanged modules can be reused in a variety of varying environments. At the time of this writing, it appears that no modular method has yet been developed which is based on fast iterative solvers. In contrast, the combination of the Schur complement method with standard direct solvers, such as LU factorization, provides a basic modular simulation method. The work presented in this dissertation consists of combining a LOGOS-based sparse factorization method with the Schur complement method to produce a modular fast direct analysis method, which will be referred to as the MFD method. Though the MFD method presented in this dissertation is a self-contained solution method, it is ultimately intended to be embedded in a more general MFD scheme which is illustrated by the flow chart pictured in Figure 1.

In Figure 1, the "baseline system" represents a sparse representation of the system matrix for a given problem (e.g., a ship). The steps indicated in the figure are:

S1) Fill the sparse representation of the system matrix for the baseline system. Factor the sparse representation using the LOGOS factorization. The results of the factorization are saved to hard-disk for later use.

S2) Later, when an engineer wishes to design a device to operate within/on the previously characterized platform (i.e., the baseline system), the first step in the proposed design procedure will consist in specifying a spatial region, which is referred to as the design region, that will contain all possible designs. The closed surface that bounds this region will subsequently be referred to as a "Green box."

S3) Once the design region has been defined, the next step in Figure 1 consists of updating the LOGOS factorization of the baseline problem to include any changes that are required by the introduction of the fictitious box. (The ability to rapidly update the factorization is a unique feature of the LOGOS factorization. Such is not generally possible using other sparse factorization strategies.)

S4) Specify a trial design structure, and fill and factor its system matrix, including all interactions with the Green box.

S5) Use the Schur complement to combine the separately factored representations of the platform and design regions.

S6) Repeat S4 and S5 until a satisfactory design is obtained.

S7) Update the baseline factorization to include the new design structure and replace the baseline factorization in the hard-disk.

While the modular design strategy indicated by Figure 1 is the principle motivation of this work, the remainder of the dissertation will focus on how to use the LOGOS factorization to realize modular EM design at a reduced cost relative to existing Schur-based methods. The extension of the work in this dissertation to the framework indicated in Figure 2 will be considered in detail elsewhere.

The flow chart for the MFD method described in this thesis is given in Figure 2. S3 of Figure 2 produces the same results as those produced in S3 of Figure 1. Other steps in Figure 2 are the same as those in Figure 1.

1.4. Objective and Scope

The purpose of this work is to develop a fast modular analysis and design method that analyzes different parts of an object in a modular way so that the solution to the whole system can be assembled with the existing analyzed modules and the newly added modules. Because the LOGOS-based fast direct solution method is used in the development of the method, it is referred to as a Modular Fast Direct, i.e. MFD, method.

Chapter 2 reviews some of the general technical details that are used throughout the thesis to analyze electromagnetic problems. In particular, this chapter provides an essential summary of integral equation formulations for time harmonic electromagnetic problems. While the methods discussed herein are not necessarily limited to integral equations, such formulations are the sole focus of this work. Chapter 2 also provides a review of the Nyström method for discretizing surface integral equations.

Chapter 3 discusses the sparse methods that will be used throughout to fill and store the system matrix. The adaptive cross approximation (ACA) is used to fill blocks of the

system matrix, which are subsequently converted into the more efficient (i.e., sparser) multilevel simply sparse method (MLSSM) representation.

Chapter 4 introduces the LOGOS concept and reviews the LOGOS factorization algorithm. Different types of LOGOS modes are briefly discussed and the focus is put on the non-overlapping, non-radiating LOGOS modes. The ΘR factorization, which helps to find the LOGOS modes efficiently, is discussed in detail in this chapter. The error control theory for the LOGOS factorization is formally developed, and numerical examples are provided to demonstrate the validity of the error control for a number of geometries. The multi-range localization method is also discussed to improve the efficiency of the LOGOS factorization.

Chapter 5 presents the MFD method for analyzing 3D EM design problems. It applies the equivalence principle so that the problems can be formulated in a modular way. The LOGOS factorization is applied to the platform system matrix and the design system matrix consecutively. The factorization on the design system matrix and the solution processes can reuse the platform factorization results. Therefore, the design cycle is greatly reduced as compared to the time used to resolve the whole problem. Further reduction of the design cycle is possible by introducing reduced order models (ROMs) for the design task. In some cases, the use of ROMs renders the design cycle independent of the platform size. Numerical examples are presented to show the performance of the MFD method for different kinds of problems.

Chapter 6 concludes the work and indicates significant directions for additional work based on the methods developed here.

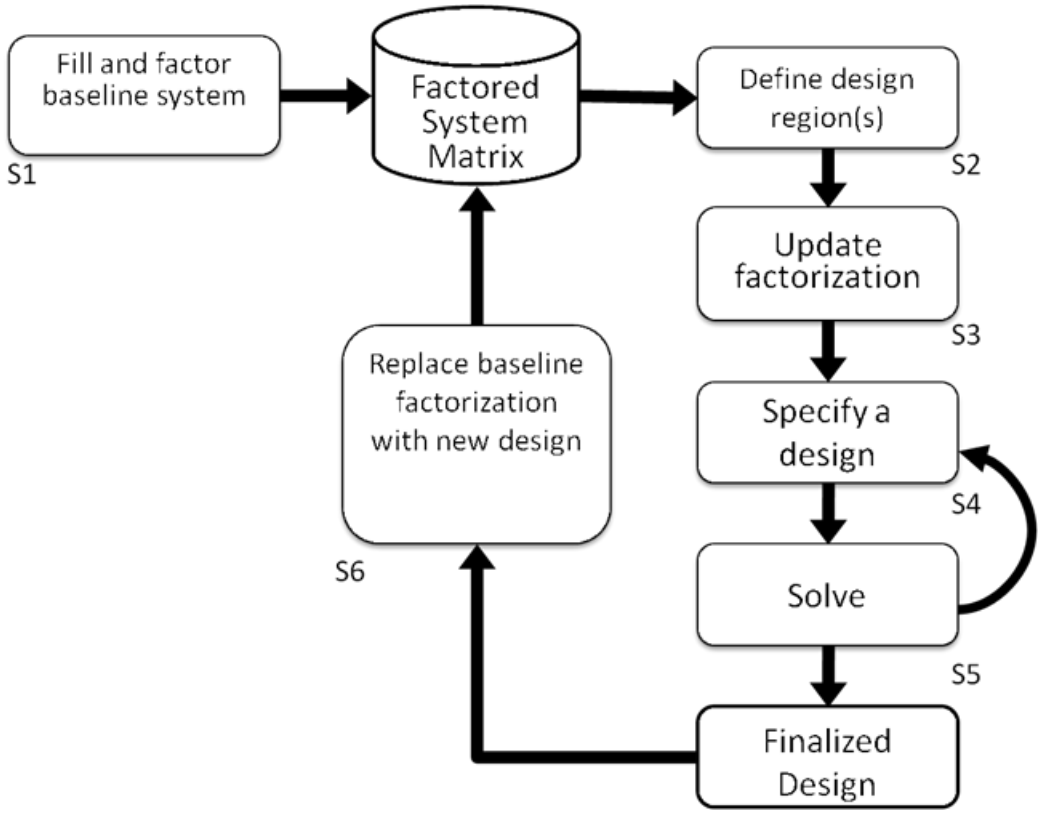


Figure 1. Advanced MFD method. See Section 1.3 for a description of the elements of this flowchart.

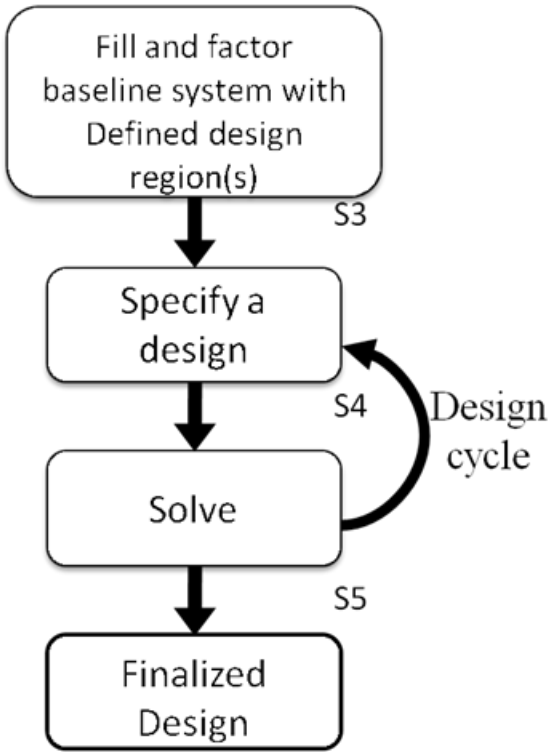


Figure 2. Flow chart of the MFD method developed in this dissertation. See Section 1.3 for a description of the elements of this flowchart.

Chapter 2. Matrix Equations for EM Problems

This work focuses on solving time-harmonic electromagnetic problems formulated with integral equations (IEs). While it is expected that the MFD method developed in this work will also be applicable to problem formulations based on both the finite element method (FEM) and hybrid FEM/IE methods, these possibilities are not considered here.

As discussed in Chapter 1, working with integral equation formulations necessitates the use of compressed representations of the underlying integral operators in order to treat large problems. Such compression is possible due to the characteristics of electromagnetic field interactions and the consequent properties of the system matrix. These characteristics are highlighted below.

For a given EM scattering/radiation problem, a range of integral equation formulations are possible. This is significant as different formulations lead to system matrices with different properties. These differences can have several consequences. In the first place, the formulation chosen can affect the accuracy of the resulting solution. Furthermore, when iterative solvers are used, the choice of formulation can dramatically impact the convergence of the underlying iterative solver, as well as the ability to develop efficient preconditioners. In the context of sparse direct methods, the choice of formulation impacts both the degree of compression that can be obtained in the system matrix as well as the efficiency of the LOGOS factorization algorithm.

This chapter reviews some of the fundamental details in electromagnetic theory and simulation related to the integral equation formulation and discretization strategies used in the remainder of the dissertation.

1.1. Basic Equations for Electromagnetic Fields

The starting point of all electromagnetic field analysis is the set of Maxwell's equations [29]. Macroscopic Maxwell's equations are convenient to work with when dealing with problems where the wavelength of the field is large enough such that the material medium can be characterized by constitutive parameters. When the medium is linear, the Maxwell's equations in the space-frequency domain are given by (with suppressed time variation $e^{j\omega t}$)

$$\nabla \times \mathbf{E} = -j\omega\mu\mathbf{H} - \mathbf{M}, \tag{2.1}$$

$$\nabla \times \mathbf{H} = j\omega\varepsilon\mathbf{E} + \mathbf{J}, \tag{2.2}$$

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon}, \tag{2.3}$$

$$\nabla \cdot \mathbf{H} = \frac{m}{\mu}, \tag{2.4}$$

with the charge conservation relations

$$\nabla \cdot \mathbf{J} = -j\omega\rho, \tag{2.5}$$

$$\nabla \cdot \mathbf{M} = -j\omega m. \tag{2.6}$$

$\mathbf{E}$, $\mathbf{H}$, $\mathbf{J}$, $\mathbf{K}$, ρ, m, ε and μ are the electric field, magnetic field, electric current, magnetic current, electric charge density, magnetic charge density, permittivity and permeability, respectively.

For a linear, homogeneous and isotropic medium, both ε and μ are scalar and not a function of position, so that the vector wave equations can be written as

$$\nabla \times \nabla \times \mathbf{E} - k^2\mathbf{E} = -j\omega\mu\mathbf{J} - \nabla \times \mathbf{M}, \tag{2.7}$$

$$\nabla \times \nabla \times \mathbf{H} - k^2\mathbf{H} = -j\omega\varepsilon\mathbf{M} + \nabla \times \mathbf{J}. \tag{2.8}$$

Using $\nabla \times \nabla \times \mathbf{A} = \nabla\nabla \cdot \mathbf{A} - \nabla^2\mathbf{A}$, (2.7) and (2.8) becomes

$$\nabla^2\mathbf{E} + k^2\mathbf{E} = j\omega\mu\mathbf{J} + \nabla \times \mathbf{M} + \nabla\rho/\varepsilon, \tag{2.9}$$

$$\nabla^2\mathbf{H} + k^2\mathbf{H} = j\omega\varepsilon\mathbf{M} - \nabla \times \mathbf{J} + \nabla m/\mu. \tag{2.10}$$

In the following discussions, unless pointed out explicitly, all mediums are linear and isotropic.

Specific EM problems specify the boundary conditions that define the relationship between the fields on the two sides of a boundary, S, between two mediums. The fundamental boundary conditions are

$$-\hat{n} \times \left(\mathbf{E}_2 - \mathbf{E}_1\right)\Big|_S = \mathbf{M}, \tag{2.11}$$

$$\hat{n} \times \left(\mathbf{H}_2 - \mathbf{H}_1\right)\Big|_S = \mathbf{J}, \tag{2.12}$$

$$\hat{n} \cdot \left(\mathbf{D}_2 - \mathbf{D}_1\right)\Big|_S = \rho, \tag{2.13}$$

$$\hat{n} \cdot \left(\mathbf{B}_2 - \mathbf{B}_1\right)\Big|_S = m, \tag{2.14}$$

where the subscripts 1 and 2 refer to the medium 1 and 2, respectively; $\hat{n}$ is the unit normal vector to the boundary pointing from medium 1 to medium 2. **M** and **J** are the surface magnetic and electric currents on the boundary. Due to the constraints of Maxwell's equations, (2.11) and (2.14) are not independent, similarly with (2.12) and (2.13).

A special boundary condition is that at the infinity. Any Cartesian component ϕ of the EM field must satisfy Sommerfeld's radiation condition [29]

$$\lim_{r\to\infty} r\left(\frac{\partial\phi}{\partial r} + jk\phi\right) = 0. \tag{2.15}$$

To derive integral representation of the fields and the associated integral equations, it is easier to work with auxiliary quantities, such as the vector and scalar potentials. Due to linearity, Maxwell's equations can be separated into two sets of equations that are dual to each other. One set is due to electric sources only. It can be written as

$$\nabla\times\mathbf{E} = -j\omega\mu\mathbf{H}, \tag{2.16}$$

$$\nabla\times\mathbf{H} = j\omega\varepsilon\mathbf{E} + \mathbf{J}, \tag{2.17}$$

$$\nabla\cdot\mathbf{E} = \frac{\rho}{\varepsilon}, \tag{2.18}$$

$$\nabla\cdot\mathbf{H} = 0. \tag{2.19}$$

The fields derived from this set of equations are called fields of electric type. The other set of equations are due to magnetic sources only, which can be obtained from (2.16)-(2.19) using the duality relations

$$\begin{aligned} &\mathbf{E}\to\mathbf{H},\ \ \mathbf{H}\to-\mathbf{E},\ \ \varepsilon\to\mu,\ \ \mu\to\varepsilon, \\ &\mathbf{J}\to\mathbf{M},\ \ \rho\to m. \end{aligned} \tag{2.20}$$

The corresponding fields are called fields of magnetic type. The total fields due to electric and magnetic sources are the summation of the fields of electric and magnetic types. The vector potentials and scalar potentials can be derived from the two sets of equations.

Equation (2.19) indicates that **H** is solenoidal and can be written as the curl of a vector. Let

$$\mathbf{H} = \frac{1}{\mu}\nabla \times \mathbf{A} = \frac{1}{\mu}\nabla \times \left(\mathbf{A} + \frac{1}{j\omega}\nabla \phi_0 \right), \tag{2.21}$$

where ϕ_0 is an arbitrary scalar function. Substituting (2.21) into (2.16), we have

$$\nabla \times \left(\mathbf{E} + j\omega \mathbf{A} + \nabla \phi_0 \right) = 0, \tag{2.22}$$

Therefore, $\mathbf{E} + j\omega \mathbf{A} + \nabla \cdot \phi_0$ is irrotational and can be written as

$$\mathbf{E} + j\omega \mathbf{A} + \nabla \phi_0 = -\nabla \phi, \tag{2.23}$$

Eliminating **E** and **H** from (2.17) using (2.21) and (2.23), and using $\nabla \times \nabla \times \mathbf{A} = \nabla \nabla \cdot \mathbf{A} - \nabla^2 \mathbf{A}$, we have

$$\nabla^2 \mathbf{A} + k^2 \mathbf{A} = \nabla \left(\nabla \cdot \mathbf{A} + j\omega \varepsilon \mu \left(\phi + \phi_0 \right) \right) - \mu \mathbf{J}\ . \tag{2.24}$$

ϕ_0 and, therefore, $\phi + \phi_0$ are arbitrary functions. Write $\phi \leftarrow \phi + \phi_0$ and let

$$\nabla \cdot \mathbf{A} + j\omega \varepsilon \mu \phi = 0. \tag{2.25}$$

Equation (2.24) becomes

$$\nabla^2 \mathbf{A} + k^2 \mathbf{A} = -\mu \mathbf{J}. \tag{2.26}$$

Taking the divergence of (2.23), we have

$$\nabla^2 \phi + k^2 \phi = -\frac{\rho}{\varepsilon}. \tag{2.27}$$

Equations (2.26) and (2.27) are the equations that govern the vector and scalar potentials produced by the electric sources. $\mathbf{A}$ is the magnetic vector potential and ϕ is the electric scalar potential. The field of electric type is given by

$$\mathbf{E} = -j\omega \mathbf{A} - \nabla \phi, \tag{2.28}$$

$$\mathbf{H} = \frac{1}{\mu}\nabla \times \mathbf{A}. \tag{2.29}$$

The potentials due to magnetic sources and the fields of magnetic type can be obtained using the duality relations (2.20) as

$$\nabla^2 \mathbf{F} + k^2 \mathbf{F} = -\varepsilon \mathbf{M}, \tag{2.30}$$

$$\nabla^2 \varphi + k^2 \varphi = -\frac{m}{\mu}, \tag{2.31}$$

$$\mathbf{H} = -j\omega \mathbf{F} - \nabla \varphi, \tag{2.32}$$

$$\mathbf{E} = -\frac{1}{\varepsilon}\nabla \times \mathbf{F}, \tag{2.33}$$

where $\mathbf{F}$ and φ are the electric vector potential and the magnetic scalar potential, respectively.

2.1. Green's Functions

The vector wave equations, (2.9) and (2.10), and the equations governing the potentials, (2.26), (2.27), (2.30) and (2.31), are all Helmholtz equations. At zero frequency (or $k \to 0$), these equations reduce to Laplace's Equations. These linear equations can be written in a general form as

$$L\left(\phi(\mathbf{r})\right) = -f(\mathbf{r}), \tag{2.34}$$

where $\mathbf{r}$ is the 3D position vector, $\phi(\mathbf{r})$ the unknown function, $f(\mathbf{r})$ the source (forcing) function and L a linear operator. (2.34) is written for scalar functions only. It can be used to represent the Cartesian components of the vector Helmholtz equation, (2.9) and (2.10). The solution to this linear equation can be easily written as a superposition of the Green's function of it, which satisfies the same boundary conditions that need to be satisfied by the unknown function $\phi(\mathbf{r})$.

The Green's function $G(\mathbf{r},\mathbf{r}')$ is the solution of

$$L\left(G(\mathbf{r},\mathbf{r}')\right) = -\delta(\mathbf{r}-\mathbf{r}'), \tag{2.35}$$

where $\delta(\mathbf{r}-\mathbf{r}')$ is the 3D Dirac delta function, which satisfies

$$\int_V f(\mathbf{r})\delta(\mathbf{r}-\mathbf{r}')dv = \int_V f(\mathbf{r})\delta(\mathbf{r}'-\mathbf{r})dv = \begin{cases} f(\mathbf{r}'), & \mathbf{r}' \in V \\ 0, & \mathbf{r}' \notin V \end{cases}, \tag{2.36}$$

where f is an arbitrary continuous function. The primed quantities are related to the source and the corresponding unprimed quantities are related to the field. If the Green's function satisfies the same boundary condition as that for the unknown function $\phi(\mathbf{r})$, the function $\phi(\mathbf{r})$ can be expressed using the Green's function as

$$f(\mathbf{r}) = \int_V g(\mathbf{r}')G(\mathbf{r},\mathbf{r}')dv'. \tag{2.37}$$

Specifically, the Green's function that satisfies the 3D scalar Helmholtz equation

$$\nabla^2 G + k^2 G = -\delta(\mathbf{r} - \mathbf{r}') \tag{2.38}$$

in free-space is given by

$$G(\mathbf{r}, \mathbf{r}') = \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|}. \tag{2.39}$$

The free-space Green's function of the two-dimensional (2-D) scalar Helmholtz equation is

$$G(\boldsymbol{\rho}, \boldsymbol{\rho}') = \frac{1}{4j} H_0^{(2)}\left(k|\boldsymbol{\rho} - \boldsymbol{\rho}'|\right). \tag{2.40}$$

At the limit $k \to 0$, the Helmholtz equation (2.38) becomes Laplace's equation

$$\nabla^2 G = -\delta(\mathbf{r} - \mathbf{r}'). \tag{2.41}$$

The Green's functions for Laplace's equation in three and two dimensions are given by

$$G(\mathbf{r}, \mathbf{r}') = \frac{1}{4\pi|\mathbf{r}-\mathbf{r}'|}, \tag{2.42}$$

and

$$G(\boldsymbol{\rho}, \boldsymbol{\rho}') = -\frac{1}{2\pi}\ln|\boldsymbol{\rho} - \boldsymbol{\rho}'|, \tag{2.43}$$

respectively.

2.2. Integral Equations

The integral representation of the fields in free-space due to electric and magnetic sources can be obtained with the aid of potentials and Green's functions. In analogy with (2.37) , the vector potential and scalar potential in free-space due to electric sources in (2.26), (2.27) can be written as

$$\mathbf{A}(\mathbf{r}) = \mu \int_V \mathbf{J}(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') dv', \tag{2.44}$$

$$\phi(\mathbf{r}) = \frac{1}{\varepsilon} \int_V \rho(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') dv'. \tag{2.45}$$

The fields of electric type can be obtained using (2.28) and (2.29) as

$$\mathbf{E} = -j\omega\mu \int_V \mathbf{J}(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') dv' - \frac{1}{\varepsilon} \int_V \rho(\mathbf{r}') \nabla G(\mathbf{r}, \mathbf{r}') dv', \tag{2.46}$$

$$\mathbf{H} = -\int_V \mathbf{J}(\mathbf{r}') \times \nabla G(\mathbf{r},\mathbf{r}')\, dv' . \tag{2.47}$$

The fields of magnetic type can be obtained using the duality relations (2.20) as

$$\mathbf{F}(\mathbf{r}) = \varepsilon \int_V \mathbf{M}(\mathbf{r}') G(\mathbf{r},\mathbf{r}')\, dv' , \tag{2.48}$$

$$\varphi(\mathbf{r}) = \frac{1}{\mu} \int_V m(\mathbf{r}') G(\mathbf{r},\mathbf{r}')\, dv' . \tag{2.49}$$

$$\mathbf{H} = -j\omega\varepsilon \int_V \mathbf{M}(\mathbf{r}') G(\mathbf{r},\mathbf{r}')\, dv' - \frac{1}{\mu} \int_V m(\mathbf{r}') \nabla G(\mathbf{r},\mathbf{r}')\, dv' , \tag{2.50}$$

$$\mathbf{E} = \int_V \mathbf{M}(\mathbf{r}') \times \nabla G(\mathbf{r},\mathbf{r}')\, dv' . \tag{2.51}$$

Equations (2.46), (2.47), (2.50) and (2.51) define the source-field relations in free-space or unbounded homogenous space.

In the following, the integral representation of the fields in the presence of piecewise homogenous material regions are derived. Consider a homogeneous region R bounded by the surface ∂R and probably the surface at infinity as shown by the two cases in Figure 3. Let the unit normal vector on the bounding surface pointing to region R be denoted by $\hat{n}$. The neighbor regions of region R, denoted by R_i in Figure 3, are those that share the boundary ∂R or part of it with region R.

Any fields $\mathbf{\Phi}$ and $\mathbf{\Psi}$ in region R satisfy the vector analogous of the Green's identity:

$$\int_R \left(\mathbf{\Phi} \cdot \nabla \times \nabla \times \mathbf{\Psi} - \mathbf{\Psi} \cdot \nabla \times \nabla \times \mathbf{\Phi} \right) dv = -\int_{\partial R} \left(\mathbf{\Psi} \times \nabla \times \mathbf{\Phi} - \mathbf{\Phi} \times \nabla \times \mathbf{\Psi} \right) \cdot \hat{n} ds . \tag{2.52}$$

Let $\mathbf{\Phi} = \mathbf{E}$ and $\mathbf{\Psi} = \mathbf{a}G$, where $\mathbf{E}$ is the total electric field in region R, $\mathbf{a}$ is an arbitrary constant vector and G is the 3D free-space Green's function. Substitute $\mathbf{\Phi}$ and $\mathbf{\Psi}$ into (2.52) and make use of the following results[1]

---

[1] The following vector identities are used to derive the resutls:

$$\nabla \times (\mathbf{A}\phi) = \nabla\phi \times \mathbf{A} + \phi(\nabla \times \mathbf{A}),$$

$$\nabla \times (\mathbf{A} \times \mathbf{B}) = \mathbf{A}(\nabla \cdot \mathbf{B}) - \mathbf{B}(\nabla \cdot \mathbf{A}) + (\mathbf{B} \cdot \nabla)\mathbf{A} - (\mathbf{A} \cdot \nabla)\mathbf{B},$$

$$\nabla(\mathbf{A} \cdot \mathbf{B}) = (\mathbf{A} \cdot \nabla)\mathbf{B} + (\mathbf{B} \cdot \nabla)\mathbf{A} + \mathbf{A} \times (\nabla \times \mathbf{B}) + \mathbf{B} \times (\nabla \times \mathbf{A}),$$

$$\begin{aligned}\nabla\times\nabla\times(\mathbf{a}G) &= \nabla\times(\nabla G\times\mathbf{a}) = -\mathbf{a}\nabla^2 G + (\mathbf{a}\cdot\nabla)\nabla G \\ &= -\mathbf{a}\nabla^2 G + \nabla(\mathbf{a}\cdot\nabla G)\end{aligned} \tag{2.53}$$

$$\begin{aligned}\int_R \mathbf{E}\cdot\nabla(\mathbf{a}\cdot\nabla G)\,dv &= \int_R \left\{\nabla\cdot\left[(\mathbf{a}\cdot\nabla G)\mathbf{E}\right] - (\mathbf{a}\cdot\nabla G)\nabla\cdot\mathbf{E}\right\} dv \\ &= \mathbf{a}\cdot\int_{\partial R}(\nabla G)\mathbf{E}\cdot\hat{n}\,ds - \mathbf{a}\cdot\int_R \nabla G(\nabla\cdot\mathbf{E})\,dv\end{aligned} \tag{2.54}$$

$$\begin{aligned}\int_{\partial R}(G\mathbf{a}\times\mathbf{M})\cdot\hat{n}\,ds &= \int_R \nabla\cdot(\mathbf{a}G\times\mathbf{M})\,dv \\ &= \mathbf{a}\cdot\int_R (\mathbf{M}\times\nabla G - G\nabla\times\mathbf{M})\,dv\end{aligned} \tag{2.55}$$

and the Helmholtz equations, (2.9) and (2.38), satisfied by $\mathbf{E}$ and G. Then, (2.52) becomes

$$\begin{aligned}&\mathbf{a}\cdot\int_R\left(j\omega\mu\mathbf{J}G + \mathbf{E}\delta(\mathbf{r}-\mathbf{r}') - \frac{\rho}{\varepsilon}\nabla G + \mathbf{M}\times\nabla G\right)dv \\ &= \mathbf{a}\cdot\int_{\partial R}\left[(\hat{n}\cdot\mathbf{E})\nabla G + (\hat{n}\times\mathbf{E})\times\nabla G + j\omega\mu G\mathbf{H}\times\hat{n}\right]ds\end{aligned}. \tag{2.56}$$

Since $\mathbf{a}$ is arbitrary, (2.56) should hold without $\mathbf{a}$. Using the reciprocity property of the Green's functions, we have

$$\begin{aligned}&\int_R\left(-j\omega\mu\mathbf{J}G + \frac{\rho}{\varepsilon}\nabla' G - \mathbf{M}\times\nabla' G\right)dv' \\ &+\int_{\partial R}\left[-j\omega\mu G(\hat{n}'\times\mathbf{H}) + (\hat{n}'\cdot\mathbf{E})\nabla' G + (\hat{n}'\times\mathbf{E})\times\nabla' G\right]ds'. \\ &= \begin{cases}\mathbf{E} & \mathbf{r}\in V \\ 0 & \mathbf{r}\in\bar{V}\end{cases}\end{aligned} \tag{2.57}$$

Using the duality relation (2.20), the magnetic field can be written as

$$\begin{aligned}&\int_R\left(-j\omega\varepsilon\mathbf{M}G + \frac{m}{\mu}\nabla' G + \mathbf{J}\times\nabla' G\right)dv' \\ &+\int_{\partial R}\left[j\omega\varepsilon G(\hat{n}'\times\mathbf{E}) + (\hat{n}'\cdot\mathbf{H})\nabla' G + (\hat{n}'\times\mathbf{H})\times\nabla' G\right]ds'. \\ &= \begin{cases}\mathbf{H} & \mathbf{r}\in V \\ 0 & \mathbf{r}\in\bar{V}\end{cases}\end{aligned} \tag{2.58}$$

Equations (2.57) and (2.58) give the expression of the fields inside region R. The volume integrals on the LHS's of (2.57) and (2.58) are the contribution to the fields from

---

$$\nabla\cdot(\mathbf{A}\times\mathbf{B}) = \mathbf{B}\cdot(\nabla\times\mathbf{A}) - \mathbf{A}\cdot(\nabla\times\mathbf{B}).$$

the sources existing in region R. The surface integrals on the LHS's of (2.57) and (2.58) are the contribution to the fields due to the fields on the boundary ∂R. Comparing the expressions in the volume and surface integrals, it is apparent that the source field relations given by (2.46), (2.47), (2.50) and (2.51) can be maintained with the following definition of equivalent surface currents and charges on the boundary ∂R:

$$\begin{aligned} \mathbf{J}_e &= \hat{n}' \times \mathbf{H}, \quad \mathbf{M}_e = -\hat{n}' \times \mathbf{E}, \\ \rho_e &= \varepsilon \hat{n}' \cdot \mathbf{H}, \quad m_e = -\mu \hat{n}' \cdot \mathbf{E}. \end{aligned} \tag{2.59}$$

The charge conservation relations also hold for the surface equivalent sources,

$$\nabla_s \cdot \mathbf{J}_e = -j\omega\rho_e; \quad \nabla_s \cdot \mathbf{M}_e = -j\omega m_e, \tag{2.60}$$

where

$$\nabla_s = \nabla - \hat{n}\frac{\partial}{\partial n}. \tag{2.61}$$

On a closed surface, the following equations always hold:

$$\int_{\partial R} \left(\hat{n}' \cdot \mathbf{E}\right) \nabla' G ds' = -\frac{j}{\omega\varepsilon} \int_{\partial R} \left[\left(\hat{n}' \times \mathbf{H}\right) \cdot \nabla'\right] \nabla' G ds', \tag{2.62}$$

$$\int_{\partial R} \left(\hat{n}' \cdot \mathbf{H}\right) \nabla' G ds' = \frac{j}{\omega\mu} \int_{\partial R} \left[\left(\hat{n}' \times \mathbf{E}\right) \cdot \nabla'\right] \nabla' G ds'. \tag{2.63}$$

Equations (2.62) and (2.63) relate the equivalent charge contributions to the fields with the equivalent current contribution to the fields.

In a source free region R, the field exists as if the equivalent sources given by (2.59) existing on the boundary ∂R radiate in free-space or homogeneous unbounded space with medium characterized by that in region R. These sources radiate the exact fields inside region R and null field outside region R. This is one of the fundamental concepts in EM theory: the equivalence theorem [30]. The equivalence theorem is used extensively in the subsequent development of the MFD method.

Rewrite (2.57) and (2.58) for a source free region R as

$$\int_{\partial R} \left[-j\omega\mu G\left(\hat{n}' \times \mathbf{H}\right) + \left(\hat{n}' \cdot \mathbf{E}\right) \nabla' G + \left(\hat{n}' \times \mathbf{E}\right) \times \nabla' G\right] ds' = \begin{cases} \mathbf{E} & \mathbf{r} \in R \\ 0 & \mathbf{r} \in \bar{R} \end{cases}, \tag{2.64}$$

$$\int_{\partial R} \left[j\omega\varepsilon G\left(\hat{n}' \times \mathbf{E}\right) + \left(\hat{n}' \cdot \mathbf{H}\right) \nabla' G + \left(\hat{n}' \times \mathbf{H}\right) \times \nabla' G\right] ds' = \begin{cases} \mathbf{H} & \mathbf{r} \in R \\ 0 & \mathbf{r} \in \bar{R} \end{cases}. \tag{2.65}$$

The field given by (2.64) and (2.65) are the secondary fields caused by some source illumination. These fields are called the scattered fields, which can also be written as

$$\int_{\partial R}\left[-j\omega\mu G\left(\hat{n}'\times\mathbf{H}\right)+\frac{1}{j\omega\varepsilon}\left[\left(\hat{n}'\times\mathbf{H}\right)\cdot\nabla'\right]\nabla'G+\left(\hat{n}'\times\mathbf{E}\right)\times\nabla'G\right]ds' = \begin{cases}\mathbf{E} & \mathbf{r}\in R\\ 0 & \mathbf{r}\in\bar{R}\end{cases}, \tag{2.66}$$

$$\int_{\partial R}\left[j\omega\varepsilon G\left(\hat{n}'\times\mathbf{E}\right)+\frac{1}{j\omega\mu}\left[\left(\hat{n}'\times\mathbf{E}\right)\cdot\nabla'\right]\nabla'G+\left(\hat{n}'\times\mathbf{H}\right)\times\nabla'G\right]ds' = \begin{cases}\mathbf{H} & \mathbf{r}\in R\\ 0 & \mathbf{r}\in\bar{R}\end{cases}. \tag{2.67}$$

Define linear integral-differential operators $\mathbf{L}$ and $\mathbf{K}$ as

$$\begin{aligned}\mathbf{L}\left(\mathbf{F}\right)&=-jk\int_{\partial R}\left(\mathbf{F}G-\frac{1}{k^2}\nabla'\cdot\mathbf{F}\nabla'G\right)ds'\\&=-jk\int_{\partial R}\left(\mathbf{F}+\frac{1}{k^2}\nabla\nabla'\cdot\mathbf{F}\right)Gds'\\&=-jk\int_{\partial R}\left(G+\frac{1}{k^2}\nabla\nabla G\cdot\right)\mathbf{F}ds',\\\mathbf{K}\left(\vec{F}\right)&=\int_{\partial R}\mathbf{F}\left(\vec{r}'\right)\times\nabla'Gds'=-\int_{\partial R}\mathbf{F}\left(\vec{r}'\right)\times\nabla Gds'.\end{aligned} \tag{2.68}$$

The scattered field can be written as

$$\mathbf{E}^{sca}\left(\mathbf{r}\right)=\eta\mathbf{L}\left(\mathbf{J}\right)-\mathbf{K}\left(\mathbf{M}\right), \tag{2.69}$$

$$\mathbf{H}^{sca}\left(\mathbf{r}\right)=\frac{1}{\eta}\mathbf{L}\left(\mathbf{M}\right)+\mathbf{K}\left(\mathbf{J}\right), \tag{2.70}$$

where the definition of the equivalent currents in (2.59) has been used, and the subscript, 'e', for the equivalent currents are dropped for simplicity. The total field in region R is given by $\mathbf{E}=\mathbf{E}^{inc}+\mathbf{E}^{sca}$ and $\mathbf{H}=\mathbf{H}^{inc}+\mathbf{H}^{sca}$. Equations (2.69) and (2.70) can be written out for every homogenous region indicated in Figure 3 to obtain the fields in that region as a function of the equivalent currents existing on its boundaries. When the fields are written for a specific region that is indexed by i, the quantities belonging to this region, such as $\mathbf{E}$, $\mathbf{H}$, ε, μ, $\hat{n}$ and k, etc., are attached with a subscript i.

The integral equations can be setup by enforcing the boundary conditions (2.11)-(2.14) that connect the total fields on the two sides of a boundary. The boundary

conditions (2.11) and (2.12) are more convenient to work with because they are related to the surface equivalent currents that are used in the formulation. Once (2.11) and (2.12) are enforced, (2.13) and (2.14) are automatically satisfied because of the constraints of Maxwell's equations. Therefore, only the tangential component of the fields on the boundary is required to setup the boundary condition. In a general format, the boundary conditions on the boundary ∂R which is shared by region R and its neighbors R_i can be written as

$$\mathbf{M} + \mathbf{M}_i = \mathbf{E} \times \hat{n}\big|_{\partial R^+} + \mathbf{E}_i \times \hat{n}_i\big|_{\partial R^-} = \mathbf{M}_B, \tag{2.71}$$

$$\mathbf{J} + \mathbf{J}_i = \hat{n} \times \mathbf{H}\big|_{\partial R^+} + \hat{n}_i \times \mathbf{H}_i\big|_{\partial R^-} = \mathbf{J}_B. \tag{2.72}$$

Note that the unit normal vector $\hat{n}$ points into region R, and $\hat{n}_i$ points into region R_i. ∂R^+ represents the region R side of boundary ∂R. ∂R^- represents the neighbor side of boundary ∂R. $\mathbf{J}_B$ and $\mathbf{M}_B$ are the physical surface currents on the boundary. $\mathbf{J}_B$ is non-zero on a boundary involving a perfect electric conductor (PEC), and $\mathbf{M}_B$ is non-zero on a boundary involving a perfect magnetic conductor (PMC). $\mathbf{J}_B$ and $\mathbf{M}_B$ are zero on boundaries between materials with finite electric and magnetic conductance. It is possible to re-define $\mathbf{J}_B$ and $\mathbf{M}_B$ to some non-zero values in order to model an impedance boundary condition, which is outside the scope of this work. Two commonly used boundary conditions are discussed here: boundaries involving PEC and boundaries involving only dielectrics.

A PEC enforces zero internal fields and supports no magnetic currents. The boundary conditions for a PEC material interface are

$$\mathbf{M} = \mathbf{E} \times \hat{n}\big|_{\partial R^+} = 0, \tag{2.73}$$

and

$$\mathbf{J} = \hat{n} \times \mathbf{H}\big|_{\partial R^+}. \tag{2.74}$$

(2.73) is equivalent to (2.71) with $\mathbf{M}_B = 0$ and $\mathbf{M}_i = 0$. (2.74) is equivalent to (2.72) with $\mathbf{J}_B = \mathbf{J}$ and $\mathbf{J}_i = 0$. i is used to index the PEC region. Equations (2.73) and (2.74) are not independent and either one can be used to setup the integral equations. Equation (2.73) results in the electric field integral equation (EFIE), and (2.74) leads to the

magnetic field integral equation (MFIE). Equation (2.75) and, therefore, the EFIE can be used for either closed or open PEC boundaries, because the conditions $\mathbf{M}_B = 0$ and $\mathbf{M}_i = 0$ can be satisfied in either case. However, equation (2.74) and, therefore, the MFIE can only be used for closed PEC boundaries because $\mathbf{J}_i = 0$ is not true for open PEC boundaries.

Both the EFIE and the MFIE suffer from the non-uniqueness of their solutions at the internal resonance frequency of a PEC body [31]. It has been shown that the combined field integral equation (CFIE), which is given by

$$\text{CFIE} = \frac{\alpha}{\eta}\text{EFIE} + (1-\alpha)\text{MFIE}, \tag{2.76}$$

where $0 < \alpha < 1$ produces unique solutions even at the resonances of the EFIE and MFIE formulations [31]. This is possible because the internal resonance frequencies of the EFIE and MFIE do not coincide.

For a boundary between two non-PEC media, the boundary conditions are

$$\mathbf{M} + \mathbf{M}_i = \mathbf{E} \times \hat{n}\big|_{\partial R^+} + \mathbf{E}_i \times \hat{n}_i\big|_{\partial R^-} = \mathbf{M}_B = 0, \tag{2.77}$$

$$\mathbf{J} + \mathbf{J}_i = \hat{n} \times \mathbf{H}\big|_{\partial R^+} + \hat{n}_i \times \mathbf{H}_i\big|_{\partial R^-} = \mathbf{J}_B = 0. \tag{2.78}$$

(2.77) and (2.78) are the PMCHWT [31, 32] formulation for a dielectric boundary. An alternative boundary condition for a dielectric boundary is known as the Müller formulation [33], which can be written as

$$2\mathbf{M} = \mathbf{M} - \mathbf{M}_i = \mathbf{E} \times \hat{n}\big|_{\partial R^+} - \mathbf{E}_i \times \hat{n}_i\big|_{\partial R^-}, \tag{2.79}$$

$$2\mathbf{J} = \mathbf{J} - \mathbf{J}_i = \hat{n} \times \mathbf{H}\big|_{\partial R^+} - \hat{n}_i \times \mathbf{H}_i\big|_{\partial R^-}. \tag{2.80}$$

The Müller formulation has some advantages over the PMCHWT formulation for low contrast materials (i.e., $\varepsilon_r < 20$) [34] because it behaves as a second-kind integral equation and the static terms of the **L**-operator in (2.68) cancel in the limit as $|\mathbf{r} - \mathbf{r}'| \to 0$. The Müller formulation has a lower condition number than the PMCHWT formulation for a Nyström discretization for moderate to low contrast materials [34].

2.3. Fields on the boundary

Equations (2.64) and (2.65) hold except when $\mathbf{r}$ is on ∂R. In the latter situation, the source and field points can coincide, and the singularity of the Green's function and its derivatives must be handled appropriately. This problem can be worked out by splitting the closed surface integral into two surface integrals: one of them shrinks to the field point in a defined limiting process, the other one expands to the original surface integral. Different limiting processes can be defined and the same results can be obtained [36-39]. Without presenting the details, the results are given here:

$$\begin{aligned}&\int_{\partial R}\left[j\omega\mu G\left(\hat{n}'\times\mathbf{H}\right)-\left(\hat{n}'\cdot\mathbf{E}\right)\nabla' G-\left(\hat{n}'\times\mathbf{E}\right)\times\nabla' G\right]ds'\\&=\frac{\mathbf{E}}{2}+PVI\int_{\partial R}\left[j\omega\mu G\left(\hat{n}'\times\mathbf{H}\right)-\left(\hat{n}'\cdot\mathbf{E}\right)\nabla' G-\left(\hat{n}'\times\mathbf{E}\right)\times\nabla' G\right]ds'\end{aligned}, \tag{2.81}$$

$$\begin{aligned}&\int_{\partial R}\left[-j\omega\varepsilon G\left(\hat{n}'\times\mathbf{E}\right)-\left(\hat{n}'\cdot\mathbf{H}\right)\nabla' G-\left(\hat{n}'\times\mathbf{H}\right)\times\nabla' G\right]ds'\\&=\frac{\mathbf{H}}{2}+PVI\int_{\partial R}\left[-j\omega\varepsilon G\left(\hat{n}'\times\mathbf{E}\right)-\left(\hat{n}'\cdot\mathbf{H}\right)\nabla' G-\left(\hat{n}'\times\mathbf{H}\right)\times\nabla' G\right]ds'\end{aligned}, \tag{2.82}$$

where $PVI\int_{\partial R}\bullet$ means the principle value integral. Equations (2.81) and (2.82) are applicable to planar boundaries. When the boundary has sharp edges and the field on the sharp edge is evaluated, the constant $1/2$ should be replaced with a value that includes the solid angle spanned by the sharp edge. The scattered field can be written as

$$\mathbf{E}^{sca}=\theta\left(\mathbf{r}\right)\mathbf{E}+\int_{\partial R}\left[j\omega\mu G\left(\hat{n}'\times\mathbf{H}\right)-\left(\hat{n}'\cdot\mathbf{E}\right)\nabla' G-\left(\hat{n}'\times\mathbf{E}\right)\times\nabla' G\right]ds', \tag{2.83}$$

$$\mathbf{H}^{sca}=\theta\left(\mathbf{r}\right)\mathbf{H}+\int_{\partial R}\left[-j\omega\varepsilon G\left(\hat{n}'\times\mathbf{E}\right)-\left(\hat{n}'\cdot\mathbf{H}\right)\nabla' G-\left(\hat{n}'\times\mathbf{H}\right)\times\nabla' G\right]ds', \tag{2.84}$$

where

$$\theta\left(\mathbf{r}\right)=\begin{cases}0 & \mathbf{r}\in R\\ 1/2 & \mathbf{r}\in\partial R\end{cases}, \tag{2.85}$$

and the integrals in (2.83) and (2.84) should be principle value integrals when $\mathbf{r}\in\partial R$.

2.4. Locally Corrected Nyström Method

The matrix equations used to develop the MFD method in this work are obtained using the locally corrected Nystr öm (LCN) method [35, 40-46]. The formulation and

notation used in the following introduction to the LCN method closely follows the discussions of [35] and [43].

Consider the integral equation used to solve for a surface current density $\mathbf{J}(\mathbf{r}')$

$$\phi^{inc}(\mathbf{r}) = \int_S K(\mathbf{r},\mathbf{r}')\mathbf{J}(\mathbf{r}')ds', \tag{2.86}$$

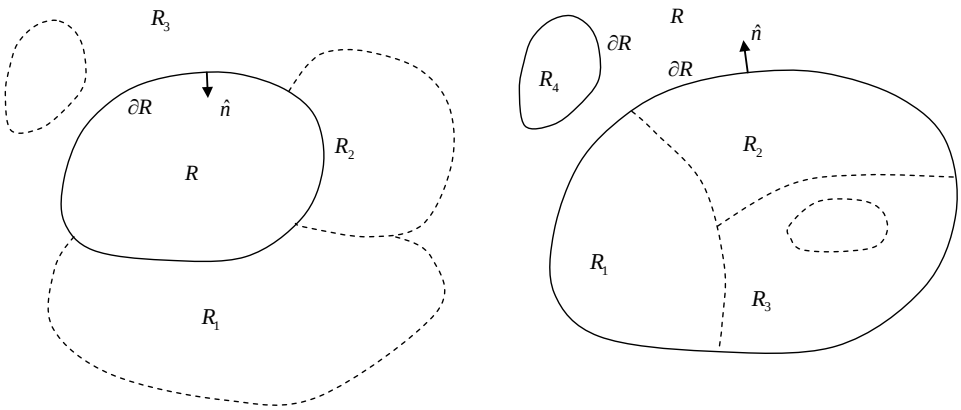
where S is a smooth surface, $\phi^{inc}(\mathbf{r})$ is the known forcing function, and $K(\mathbf{r},\mathbf{r}')$ is the kernel. The surface S is discretized into N_p curvilinear patches that represent the surface contour to high order. The integral over each patch is then approximated using a suitable quadrature rule. The observation field is sampled at the quadrature points, leading to a square linear system of equations, with the q_m th row defined as

$$\phi^{inc}(\mathbf{r}_{q_m}) = \sum_{p=1}^{N_p}\sum_{q=1}^{N_q}\omega_{q_p}K(\mathbf{r}_{q_m},\mathbf{r}_{q_p})\mathbf{J}(\mathbf{r}_{q_p}), \tag{2.87}$$

where $\mathbf{r}_{q_p}$ and ω_{q_p} are the qth abscissas and weights on the pth patch. For electromagnetic applications, the kernel is undefined at vanishing separation of the source and field points. Therefore, (2.87) is ill posed. This issue can be circumvented by the so-called local correction method. Rewrite (2.87) as

$$\phi^{inc}(\mathbf{r}_{q_m}) = \sum_{p\in far}\sum_{q=1}^{N_q}\omega_{q_p}K(\mathbf{r}_{q_m},\mathbf{r}_{q_p})\mathbf{J}(\mathbf{r}_{q_p}) + \sum_{p\in near}\sum_{q=1}^{N_q}\tilde{\omega}_{q_{p(m)}}\mathbf{J}(\mathbf{r}_{q_p}), \tag{2.88}$$

where $\tilde{\omega}_{q_{p(m)}}$ are the weights of the specialized local quadrature rule for integrating $\int_S K(\mathbf{r},\mathbf{r}')\mathbf{J}(\mathbf{r}')ds'$ when the underlying quadrature scheme is not enough to account for the singular behavior of the kernel $K(\mathbf{r},\mathbf{r}')$. The abscissa points for the specialized, or local, quadrature rule are chosen to be the same as the underlying quadrature rule of the Nyström scheme. The weights $\tilde{\omega}_{q_{p(m)}}$ are solved by requiring that the local quadrature rule integrates (2.86) to the desired accuracy when the currents are replaced by the set of functions used to define the underlying quadrature rule. The LCN method described above can be shown to be equivalent to a quadrature sampled high order moment method [43].



(a) (b)

Figure 3 Arbitrary 3-D scatterers analyzed with the present method. (a) A finite homogenous region R is bounded by ∂R with inward normal $\hat{n}$. Its neighbors, R_1, R_2 and R_3, share the boundary ∂R with it. (b) An infinite homogeneous region R is bounded by ∂R with inward normal $\hat{n}$ and the surface at infinity. Its neighbors, R_1, R_2, R_3 and R_4, share the boundary ∂R with it.

Chapter 3. Sparse Representations of System Matrix

Fast direct solvers rely on a sparse representation of the system matrix. When the system matrix is inherently sparse, such as when the finite element method (FEM) is used to discretize the underlying differential equations, fast direct solvers that exploit the sparse structure of the matrix can be developed and applied to the standard system matrix. On the other hand, when integral equation formulations such as those discussed in Chapter 2 are used, the resulting matrix equation is dense and general, exact direct solution methods will have a complexity of $O(N^3)$, which is generally considered to be excessive for large values of N. Fortunately, the matrices derived from integral equation formulations of electromagnetic phenomena can be approximated by sparse representations that have controllable error. Direct solvers can subsequently be used to invert the sparse approximation to the original full matrix.

In this chapter, the approaches to efficiently fill and store the full system matrix in a sparse format are reviewed. Some modifications are introduced to the original approaches to adapt to the needs of our applications, i.e., the development of LOGOS factorization and the MFD method.

3.1. Review on Sparse Representations

A sparse representation of a full matrix is obtained by approximating the full matrix to a certain error. The purpose is to save memory, both the final memory used to save the sparse representation and the peak memory during the process to obtain the sparse representation. Because the matrix is sparse, i.e. not all elements need to be saved, any operations defined on it are presumably faster than those defined on the corresponding full matrix. To improve the efficiency of the whole algorithm, all operations, including that used to obtain the sparse representation of the full matrix from the mathematical representation, should operate only on sparse representations of the system matrix. Some available techniques for obtaining a sparse matrix representation from mathematical formulations include panel clustering [47, 48], wavelet methods [49], multipole expansions [3, 50, 51], interpolation [52] and the adaptive cross approximation (ACA) and related H-matrix methods [53-58].

Both panel clustering and multipole expansion methods require analytical expansion of the kernel function which is either inconvenient or difficult to achieve for general kernels. The interpolation method can be used instead of analytical expansion but the price paid is performance. Wavelet methods are (thus far) limited to problems where the underlying geometry can be described by a small number of smooth maps.

The ACA method is an algebraic method where no expansion of the kernel function is needed. The proof of convergence exists for Nyström discretizations of single layer potential [58]. An improved version of ACA, called hybrid ACA, combines the ACA algorithm with an interpolation-based separation of the kernel function. It has been proved that the hybrid ACA converges for both single and double layer potentials of asymptotically smooth kernels as well as Nyström, collocation or Galerkin boundary element formulations [58].

The ACA method is normally used to fill far interaction blocks in the system matrix. Using multilevel tree structures to decompose the underlying geometry in the problem is an easy way to differentiate near and far interaction blocks. The hierarchical matrix (H-matrix) is a type of representation obtained using ACA. It has almost linear memory complexity [59] at low frequency but higher complexity at high frequency. Both H^2-matrix [60, 61] and MLSSM [62-64] provide more memory savings at high frequency due to their nested structures, which is similar to the FMM representation [65].

This dissertation utilizes the ACA method to obtain a sparse representation of the system matrix. The MLSSM representation is used to achieve further compression of the system matrix. The two methods are discussed in the following sections.

3.2. Multilevel Spatial Decomposition of a Geometry

All sparse representations presented in this dissertation rely on a multilevel spatial decomposition of the underlying geometry. This is accomplished using an oct-tree for three-dimensional applications. The resulting spatial groupings are the same as those associated with the multilevel fast multipole method (MLFMM) [66]. However, the minimal group size is not limited by the wavelength in our case.

The number of levels in the oct-tree will be denoted by L. The individual levels are indexed by l, $l = 1, \ldots, L$. The level $l = 1$ is the root level of the tree. The root level

consists of a single group containing all spatial samples. The number of nonempty groups at the lth level of the tree is $M(l)$, and the number of spatial samples in each group is approximately $N/M(l)$. The total number of levels, L, is chosen so that the smallest spatial group in each branch of the multilevel tree contains approximately twenty DOF. At a given level (level-l) of the tree, those groups sharing the boundaries of the i th group are referred to as the level-l near-neighbors of the i th group. (The i th group is also considered to be a near-neighbor to itself.) The remaining groups at level-l are referred to as far (or non near-neighbor) groups. The groups that have the same parent group are called sibling groups. Sibling groups are all near-neighbor groups, but not all near-neighbor groups are sibling groups. In the following discussion it will be convenient to indicate the i th group at level-l using the notation $i(l)$. Similarly, the notation $\mathbf{z}_{i(l)}$ will be used to indicate the submatrix of a level-l matrix, $\mathbf{Z}_l$, associated with source group $i(l)$.

As an example, consider a PEC strip which is fitted into a 4-level oct-tree. The non-empty groups at each level are indexed by the integers as indicated in Figure 4. The horizontal braces in Figure 4 denote the parent-children relation between groups in two consecutive levels of the tree. For this simple structure, any two groups at a level indexed by consecutive integers are near-neighbor groups.

3.3. Adaptive Cross Approximation

We implemented the standard ACA procedure discussed in [57] to fill sparse representations of the system matrix obtained from the EFIE, MFIE and CFIE formulations. However, the standard ACA does not work for matrices like

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \tag{3.1}$$

or more complicated checkerboard-like structures. The reason is that the sampling of rows and columns passing $\mathbf{A}$ can never be pivoted to sample those rows and columns passing $\mathbf{B}$ because of the off-diagonal zero blocks. The hybrid ACA is one way to solve this problem but it involves many changes to an existing code to implement either the

interpolation or the cross approximation [58]. In lieu of such modifications, we have developed an alternative strategy to address this type of problem.

In the following subsections, the standard ACA and the modified version used to treat matrices of the type indicated by (3.1) are discussed.

3.3.1. Standard ACA method

ACA is suitable to fill the matrix blocks that represent interactions between well separated groups. Once the underlying geometry is decomposed by an oct-tree (assuming a three-dimensional geometry), the system matrix, $\mathbf{Z}$, can be written as:

$$\mathbf{Z} = \sum_{l=2}^{L} \mathbf{Z}_l^{near} \tag{3.2}$$

where $\mathbf{Z}_l^{near}$ is the near-neighbor interaction blocks of the system matrix at level-*l*. Figure 5 illustrates the partition of the system matrix according to (3.2). Except $\mathbf{Z}_4^{near}$, the near interaction blocks at level-*l*, $\mathbf{Z}_l^{near}$, in (3.2) can be considered as the non-near-neighbor interaction blocks at level-(*l*+1), i.e., $\mathbf{Z}_l^{near} = \mathbf{Z}_{l+1}^{non-near}$. Equation (3.2) can be rewritten as

$$\mathbf{Z} = \mathbf{Z}_L^{near} + \sum_{l=3}^{L} \mathbf{Z}_l^{non-near} \tag{3.3}$$

The matrix $\mathbf{Z}_l^{non-near}$ can be filled using the ACA. To this end, rewrite $\mathbf{Z}_l^{non-near}$ as

$$\mathbf{Z}_l^{non-near} = \left[\mathbf{Z}_{1(l)}^{non-near}, \cdots, \mathbf{Z}_{i(l)}^{non-near}, \cdots, \mathbf{Z}_{M(l)}^{non-near} \right]. \tag{3.4}$$

For each column block in (3.4), the ACA procedure given in [57] is used to obtain the outer product representation:

$$\mathbf{Z}_{i(l)}^{non-near} \approx \tilde{\mathbf{Z}}_{i(l)}^{(K)} = \tilde{\mathbf{U}}_{i(l)} \tilde{\mathbf{V}}_{i(i)}^{H} = \sum_{k=1}^{K} \mathbf{u}_k \mathbf{v}_k^H , \tag{3.5}$$

where K is the effective rank of the matrix $\mathbf{Z}_l^{non-near}$, $\tilde{\mathbf{U}}_{i(l)}$ and $\tilde{\mathbf{V}}_{i(l)}^H$ are rectangular full matrices with columns and rows given by $\mathbf{u}_k$ and $\mathbf{v}_k^H$, respectively. The superscript, H, means Hermitian conjugate. The ACA procedure is to find $\mathbf{u}_k$ and $\mathbf{v}_k^H$ for $k = 1, \cdots, K$ adaptively. After each $\mathbf{u}_k$ and $\mathbf{v}_k^H$ are found, the convergence of the collective

approximation to the original matrix is checked by calculating how much modification the last $\mathbf{u}_k$ and $\mathbf{v}_k^H$ introduced to the approximation (3.5). The criteria of convergence used in [57] is $\|\mathbf{u}_k\|\|\mathbf{v}_k^H\| \le \varepsilon_{ACA}\left\|\tilde{\mathbf{Z}}_{i(l)}^{(k)}\right\|$, where ε_{ACA} is the tolerance for ACA fill. This criteria controls the relative error in the matrix block, $\mathbf{Z}_{i(l)}^{non-near}$, which is the non-near interaction part of the matrix block associated with the source group $i(l)$. For our application, it is sufficient to control the relative error in the full matrix block associated with the source group $i(l)$, which includes the near interaction part, because the LOGOS factorization discussed in the later chapters effectively compresses the full matrix block associated with the source group $i(l)$. In the remainder of this section we develop an alternative ACA termination condition.

Consider a simple case where the mesh contains very small density variation. We want to obtain a representation (3.5) for every block of the system matrix. Using (3.5), (3.3) can be written as

$$\mathbf{Z} = \mathbf{Z}_L^{near} + \sum_{l=3}^{L}\left(\tilde{\mathbf{Z}}_l^{(K)} + \mathbf{Z}_{l,err}\right) \tag{3.6}$$

where $\mathbf{Z}_{l,err}$ is the error matrix caused by incomplete sampling of the original system matrix blocks. The total error satisfies

$$\mathbf{Z}_{err} = \sum_{l=3}^{L}\mathbf{Z}_{l,err}\ . \tag{3.7}$$

Assume the error is distributed across all blocks homogeneously. The target of error control is

$$\frac{\|\mathbf{Z}_{err}\|_F}{\|\mathbf{Z}\|_F} \le \varepsilon_{ACA}\,. \tag{3.8}$$

Or,

$$\frac{\|\mathbf{Z}_{err}\|_F^2}{\|\mathbf{Z}\|_F^2} = \frac{\sum_{l=3}^{L}\|\mathbf{Z}_{l,err}\|_F^2}{\|\mathbf{Z}\|_F^2} = \frac{\sum_{l=3}^{L}\sum_{i=i(l)}^{M(l)}\|\mathbf{Z}_{i(l),err}\|_F^2}{\|\mathbf{Z}\|_F^2} \le \varepsilon_{ACA}^2\,. \tag{3.9}$$

Since we have assumed that the error is homogeneously distributed, (3.9) can be satisfied by requiring that

$$\left\|\mathbf{Z}_{i(l),err}\right\|_F^2 \le \varepsilon_{ACA}^2 \left\|\mathbf{Z}\right\|_F^2 \frac{m \times n}{N \times N}. \tag{3.10}$$

where we assume $\mathbf{Z}_{i(l),err}$ is a $m \times n$ rectangular matrix and the total number of unknown in the problem is N. The term $\frac{m \times n}{N \times N}$ arises because the error allowed for a matrix block is proportional to the number of elements in the block. The following two approximations are used in the ACA tolerance control:

$$\left\|\mathbf{Z}_{i(l),err}\right\|_F^2 \approx \left\|\mathbf{u}_k\right\| \left\|\mathbf{v}_k^H\right\|, \tag{3.11}$$

$$\left\|\mathbf{Z}\right\|_F^2 \approx \left\|\mathbf{Z}_L^{near}\right\|_F^2. \tag{3.12}$$

Substituting (3.11) and (3.12) into (3.10), we have

$$\left\|\mathbf{u}_k\right\| \left\|\mathbf{v}_k^H\right\| \le \varepsilon_{ACA} \left\|\mathbf{Z}_L^{near}\right\|_F \sqrt{\frac{m \times n}{N \times N}}. \tag{3.13}$$

(3.13) is the tolerance control criteria we used in our implementation of ACA.

3.3.2. Modified ACA method

ACA cannot fill a matrix having a structure like (3.1) because any rows/columns intersecting the block $\mathbf{A}$ will always have zero elements in the columns/rows where $\mathbf{B}$ is saved. Because the ACA algorithm always chooses the next row/column by looking for the maximum absolute value in the previously sampled column/row, the matrix $\mathbf{B}$ can never be sampled without special requirements. This deficiency of the ACA is not surprising, since the algorithm was developed to treat smooth kernels, and the matrix indicated in (3.1) has abrupt changes at the edges of $\mathbf{A}$ and $\mathbf{B}$.

Fortunately, this deficiency of the ACA can be resolved by introducing some additional logic into the algorithm in order to force it to cover $\mathbf{B}$. Before any columns are sampled, all row indices of the matrix block are collected as a list of overlooked rows. Once a column is sampled, the position of the non-zero elements are removed from the list of overlooked rows. The same list is built for the overlooked columns. After the convergence of the ACA is reached, the list of overlooked rows and columns are

checked. If there are still elements inside, the next row or column sample should be chosen in the list of overlooked rows or columns. The ACA continues until another convergence is reached. The check for the overlooked rows and columns are carried out again until there are no more overlooked rows and columns.

This modified ACA is implemented in the development of the MFD method where the checker-board structured matrix blocks are possible because the Green-box contains a planar material surface and both EFIE and MFIE kernels are present.

3.4. Multilevel Simply Sparse Method

The MLSSM [62-64, 67] representation provides a representation of the system matrix which is asymptotically more efficient than the sparse outer product representation obtained from ACA. The MLSSM representation is similar to both the multilevel FMM and the H^2 H-matrix representation [59, 60]. In this section, the structure of the MLSSM representation is discussed, and the procedure used to obtain the MLSSM representation from blocks of the ACA representation is described.

3.4.1. Structure of MLSSM representation

The structure of the MLSSM is indicated by the following multilevel recursion formula:

$$\mathbf{Z}_l = \hat{\mathbf{Z}}_l + \mathbf{U}_l \, \mathbf{Z}_{l-1} \, \mathbf{V}_l^H, \ \left(l = 2,\ldots,L\right), \tag{3.14}$$

In (3.14), $\hat{\mathbf{Z}}_l$ is the sparse matrix containing all near-neighbor interactions at level-l of the Oct-tree which were not represented at a finer level of the tree. The matrices $\mathbf{U}_l$ and $\mathbf{V}_l$ are rectangular, orthonormal, block diagonal matrices which effectively compress interactions between far groups at level-l of the Oct-tree. The original impedance matrix is recovered from (3.14) when $l = L$ (i.e., $\mathrm{Z} = \mathrm{Z}_L$). When $l = 2$, equation (3.14) reduces to

$$\mathbf{Z}_2 = \hat{\mathbf{Z}}_2 . \tag{3.15}$$

The matrices $\mathbf{U}_2$ and $\mathbf{V}_2$ are not defined in this case because all interactions at level-2 are between touching (i.e., near-neighbor) groups. For the same reason, the recursion indicated by (3.14) and (3.15) is not continued to level-1.

Taking the PEC strip shown in Figure 4 as an example, the system matrix can be written as

$$\mathbf{Z} = \hat{\mathbf{Z}}_4 + \mathbf{U}_4\left(\hat{\mathbf{Z}}_3 + \mathbf{U}_3\mathbf{Z}_2\mathbf{V}_3^H\right)\mathbf{V}_4^H, \tag{3.16}$$

or

$$\mathbf{Z} = \hat{\mathbf{Z}}_4 + \mathbf{U}_4\hat{\mathbf{Z}}_3\mathbf{V}_4^H + \mathbf{U}_4\mathbf{U}_3\mathbf{Z}_2\mathbf{V}_3^H\mathbf{V}_4^H. \tag{3.17}$$

The three terms on the RHS of (3.17) represent the near-neighbor interaction blocks at levels 4, 3 and 2, respectively, that have not been represented by the finer level near-neighbor interactions. They fit into different portions of the system matrix as shown by Figure 6. $\hat{\mathbf{Z}}_4$ saves only the shaded blocks in Figure 6-(a). $\mathbf{U}_4\hat{\mathbf{Z}}_3\mathbf{V}_4^H$ recovers the shaded blocks in Figure 6-(b), with the block diagonal matrices $\mathbf{U}_4$ and $\mathbf{V}_4^H$, and the compressed version of $\mathbf{U}_4\hat{\mathbf{Z}}_3\mathbf{V}_4^H$ saved in $\hat{\mathbf{Z}}_3$. The structures of $\mathbf{U}_4$, $\mathbf{V}_4^H$ and $\hat{\mathbf{Z}}_3$ are shown in Figure 7. $\mathbf{U}_4\mathbf{U}_3\mathbf{Z}_2\mathbf{V}_3^H\mathbf{V}_4^H$ recovers the shaded blocks in Figure 6-(c). The structures of $\mathbf{U}_3$, $\mathbf{V}_3^H$ and $\mathbf{Z}_2$ are shown in Figure 8.

In the following discussion, we will make use of the fact that although the matrices $\mathbf{Z}_{l-1}$ appearing in (3.14) have a level-$(l-1)$ subscript, they are actually comprised of blocks which represent interactions between non-near-neighbor groups at level-l. (This follows from the fact that the matrices $\mathbf{U}_l$ and $\mathbf{V}_l^H$ in (3.14) are level-l block-diagonal matrices.) This is also true of $\mathbf{Z}_2$ in (3.15). The matrix $\mathbf{Z}_2$ is comprised non-near-neighbor interactions between level-3 groups. The interactions between groups that are neither near-neighbors nor non-near-neighbors are referred to as far interactions.

3.4.2. How to Build MLSSM

The basic procedure for building MLSSM from ACA representation is:

1) Save the near interaction blocks at the finest level (level-L) to $\hat{\mathbf{Z}}_L$. Let $l = L$.

2) Find the minimal source basis for the far interaction blocks, $\mathbf{Z}_{i(l)}^{far}$, of each source group, $i(l)$, at level-*l*.

3) Find the minimal field basis for the far interaction blocks of each field group at level-*l*. This step is similar to the step-2 but the far interaction blocks are row blocks associated with every field group instead of column blocks associated with every source group.

4) With these minimal basis, find the compressed non-near neighbor interaction blocks at level-*l*. These are the near neighbor interactions at level-(*l*-1). Therefore, they are represented symbolically by $\hat{\mathbf{Z}}_{l-1}$ according to (3.14).

5) Use the minimal basis obtained above to find the projected ACA representation of the rest of the matrix, which are the level-(*l*-1) far interaction matrices.

6) Decrease the level index by one, such that $l \Leftarrow l-1$. Repeat steps 2-5 until level-3.

The basic procedure described above follows the recursive formula (3.14) nicely. However, this procedure requires that the entire ACA representation of the full system matrix be present in memory all at once, which creates a memory peak usage in the algorithm. A modified version of this procedure has been developed which lowers the peak memory required to build the MLSSM representation [63].

The method discussed in [63] is used hereinafter to build the MLSSM structure without requiring the presence of the entire ACA structure in physical memory at any point in time. The idea of this method is to incorporate the non-near neighbor interaction blocks represented using ACA into the MLSSM representation of the far interaction blocks at each level. Therefore, this method works from level-3 until the finest level.

3.4.3. Efficiency and Accuracy of MLSSM

The CPU Time for obtaining the MLSSM representation and the memory used for the representation have been reported in [63]. More examples are quoted here to demonstrate the efficiency and accuracy of MLSSM [68]. The examples are formulated using surface IE's and discretized using the locally corrected Nyström method [35] and [43].

The first example contains a PEC sphere with radius 2λ. The EFIE formulation is used. The density of the mesh on the sphere is increased from 300 DOF/λ^2 onwards in order to pack more unknowns into the system. The time used to obtain the MLSSM representation of the system matrix is shown in Figure 9 for three tolerances, 10^{-3}, 10^{-4} and 10^{-5}. It is observed that the CPU cost to build the MLSSM representation is about linear as a function of the number of unknowns. The memory usage of this example is plotted in Figure 10. A better than linear memory usage can be observed.

The second example contains a vias like structure as shown in Figure 11. The detailed geometric specification of this structure is given in Section 4.2.6. The density of the mesh on this structure is increased from 300 DOF/λ^2 onwards. Figure 12 and Figure 13 show the time used to obtain the MLSSM representation and the memory usage for the representation, respectively. The trends of the curves are similar to those of the first example.

The third example contains a series of spheres with radii of 2λ, 2.83λ, 4λ and 5.66λ. The CFIE formulation is used for this example. The density of the mesh provides an average of 300 DOFs/λ^2. Figure 14 shows the time used to build the MLSSM representation. The time to build the representation scales more rapidly in this case, approximately as $\left(O\left(N^{1.8}\right)\right)$. Figure 15 shows the memory usage of this example. The memory usage also scales higher than the last two examples, approximately as $O\left(N^{1.2}\right)$. These data indicate that the MLSSM representation is not as efficient for the high frequency case as for the low frequency cases shown in the last two examples.

Figure 16 to Figure 18 show the relative RMS error of the MLSSM representation for the three cases shown above. Three tolerances are used for each case. It can be seen that the relative RMS error of the MLSSM representation scales roughly by the same factor when the tolerance is scaled. Besides the scalability of the error control, Figure 18 shows that the relative RMS error increases as the number of unknowns increases. This is the case where the mesh density is kept constant and the radius of the sphere increases. It is still not clear why the error increases for this high frequency case. More work is being done to obtain better error control.

In the remainder of the dissertation, we will assume the existence of an MLSSM representation of the system matrix having a root-mean-square (RMS) error controlled by ε_{ssm}. The LOGOS factorization will begin with this sparse MLSSM representation of the system matrix.

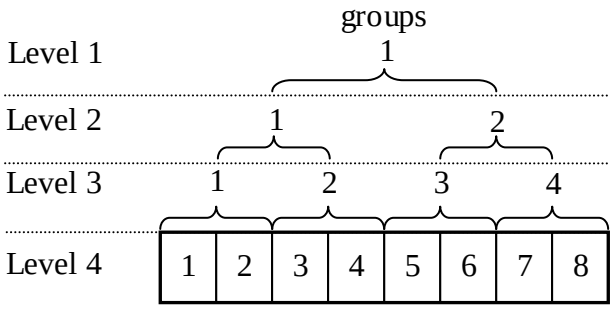


Figure 4. A PEC strip is fitted into a 4 level oct-tree. Due to the simple structure of the strip, many groups in the oct-tree are empty. The resulting tree has eight non-empty groups at level 4. There are four, two and one groups at levels 3, 2 and 1, respectively. Each group at levels 3, 2 and 1 has two children as indicated by the horizontal braces.

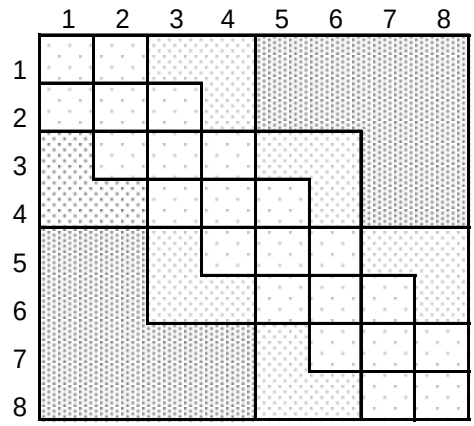


Figure 5. The system matrix for a PEC strip. The strip is decomposed in a 4-level oct-tree as shown in Figure 4.

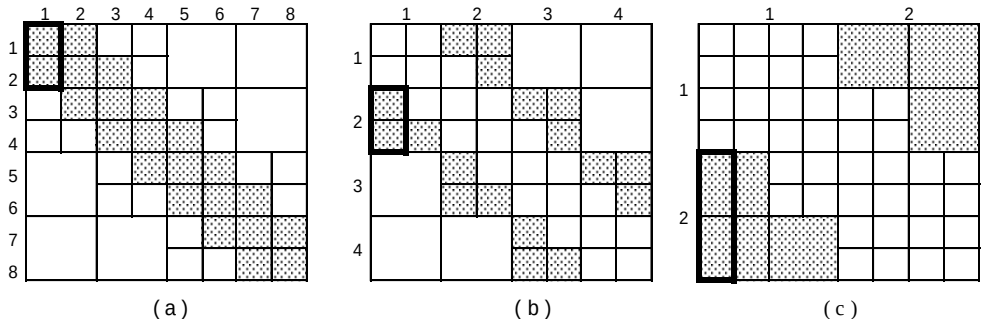


Figure 6. Shaded blocks indicate the pieces of $\mathbf{Z}$ that are represented by the different terms in the MLSSM representation: (a) $\hat{\mathbf{Z}}_4$, (b) $\mathbf{U}_4\hat{\mathbf{Z}}_3\mathbf{V}_4^H$, (c) $\mathbf{U}_4\mathbf{U}_3\mathbf{Z}_2\mathbf{V}_3^H\mathbf{V}_4^H$. Lines are used to indicate blocks of $\mathbf{Z}$ corresponding to interactions at levels three (large squares) and four (smaller squares). The blocks bordered with bold lines are different portions of $\mathbf{Z}$ associated with source group 1 at level-4. These blocks are represented by the MLSSM representation as: (a) $\hat{\mathbf{Z}}_{1(4)}$, (b) $\mathbf{U}_4\hat{\mathbf{Z}}_3\mathbf{V}_{1(4)}^H$, (c) $\mathbf{U}_4\mathbf{U}_3\mathbf{Z}_2\mathbf{V}_3^H\mathbf{V}_{1(4)}^H$.

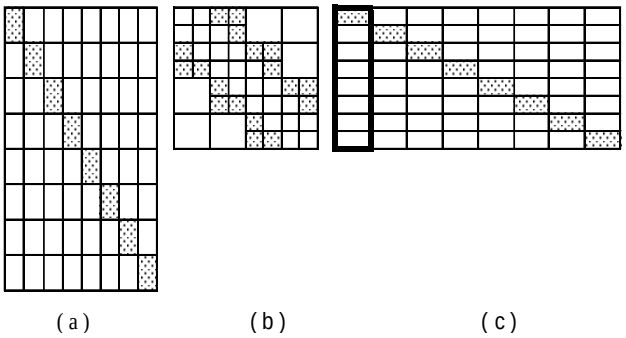


Figure 7. Level 4 SSM matrices (a): $\mathbf{U}_4$; (b): $\hat{\mathbf{Z}}_3$; (c): $\mathbf{V}_4^H$.

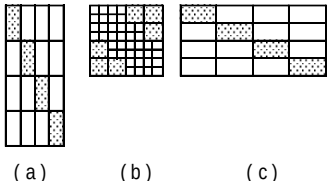


Figure 8. Level 3 SSM matrices (a): $\mathbf{U}_3$; (b): $\hat{\mathbf{Z}}_2$; (c): $\mathbf{V}_3^H$.

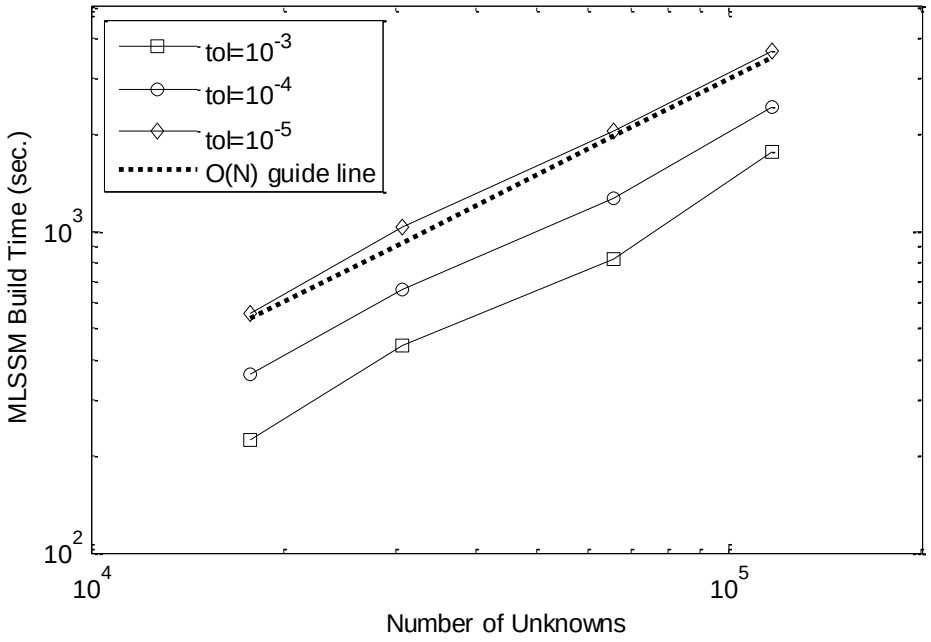


Figure 9. The CPU time used to fill the MLSSM representation of the system matrix for a PEC sphere with radius 2λ. The EFIE formulation with Nyström discretization is used to obtain the matrix equation. The coarsest mesh has a density of 300 DOF/λ^2.

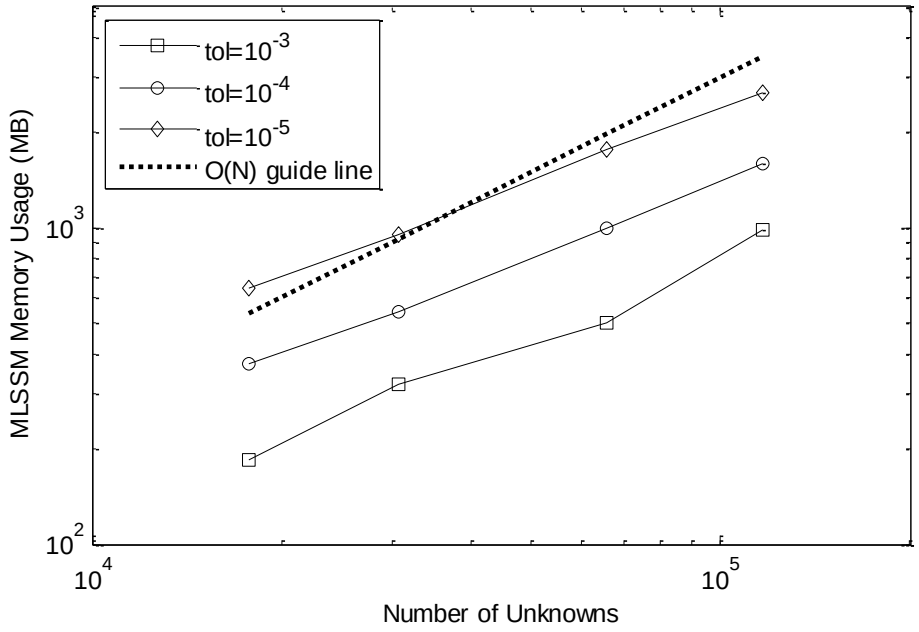


Figure 10. The memory usage of the MLSSM representation of the system matrix for a PEC sphere with radius 2λ. The EFIE formulation with Nyström discretization is used to obtain the matrix equation. The coarsest mesh has a density of 300 DOF/λ^2.

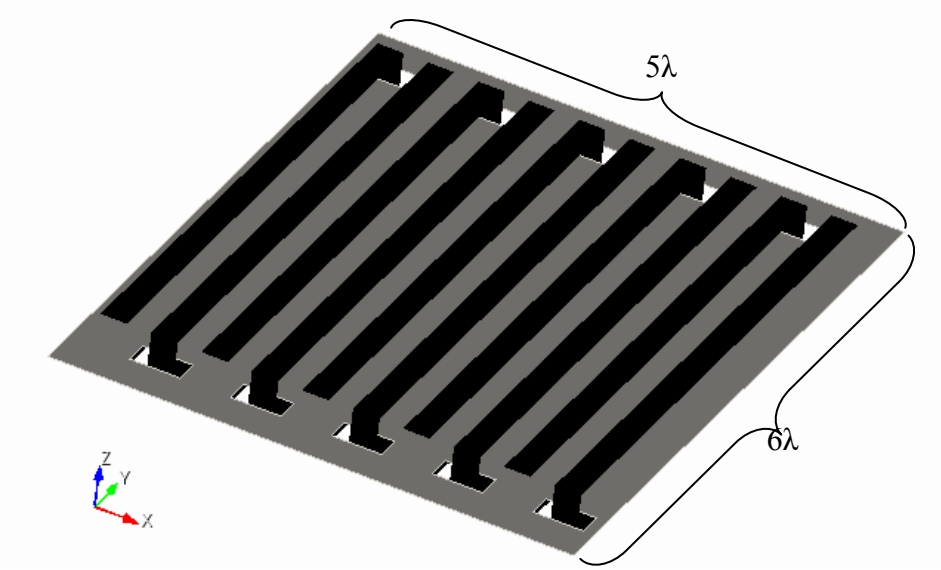


Figure 11. A vias-like geometry. It consists of U-shaped PEC strips that pass through rectangular holes on a PEC plate. The strips and plate are assumed to have zero thickness.

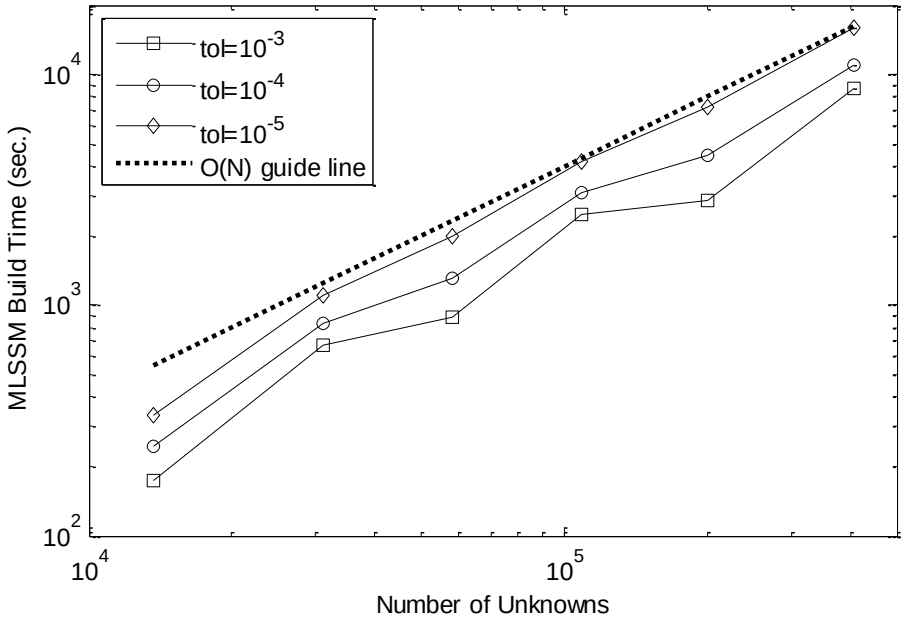


Figure 12. The CPU time used to fill the MLSSM representation of the system matrix for a vias-like structure (see Figure 11). The EFIE formulation with Nystr öm discretization is used to obtain the matrix equation.

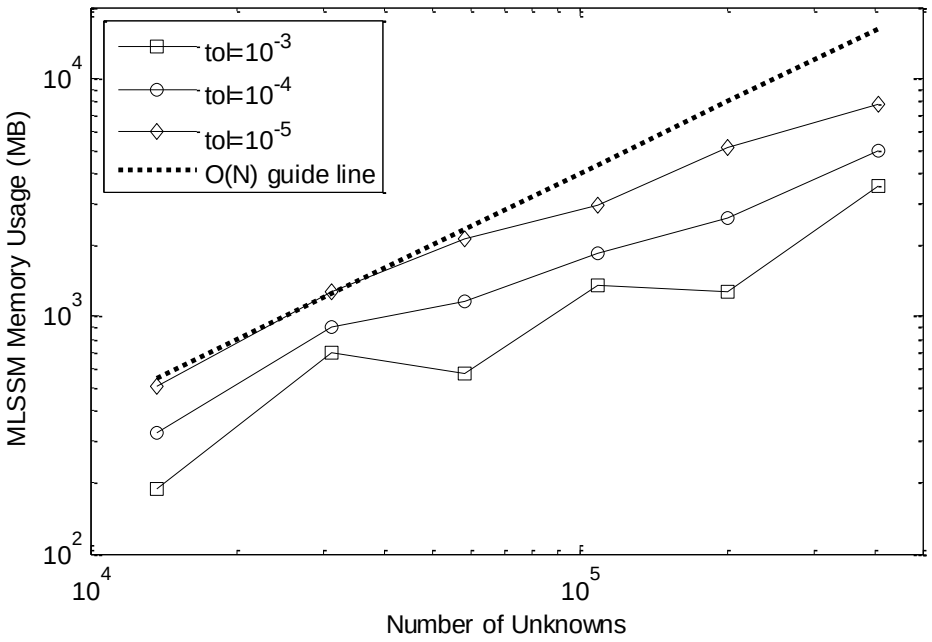


Figure 13. The memory usage of the MLSSM representation of the system matrix for a vias-like structure (see Figure 11). The EFIE formulation with Nyström discretization is used to obtain the matrix equation.

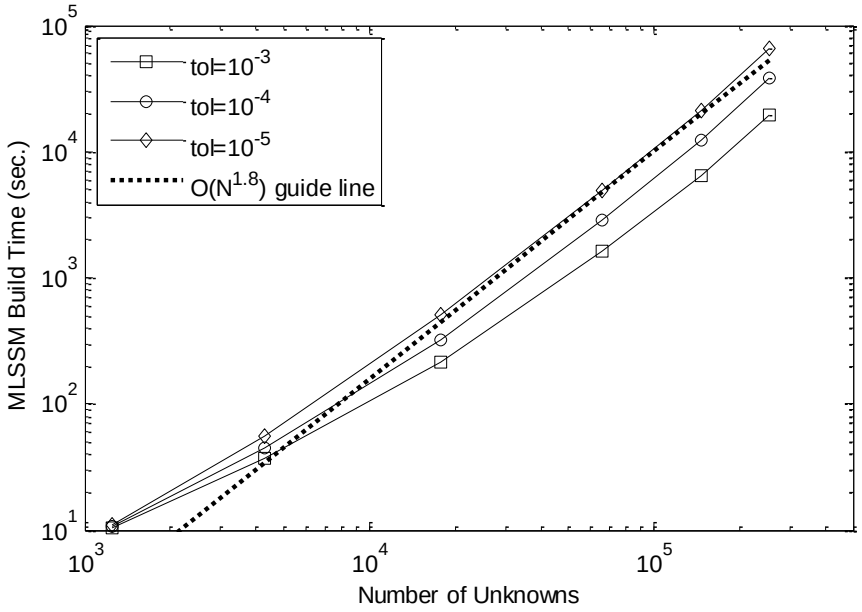


Figure 14. The CPU time used to fill the MLSSM representation of the system matrix for a series of spheres with radii of 2λ, 2.83λ, 4λ and 5.66λ. The density of the mesh provides an average of 300 DOF/λ^2.

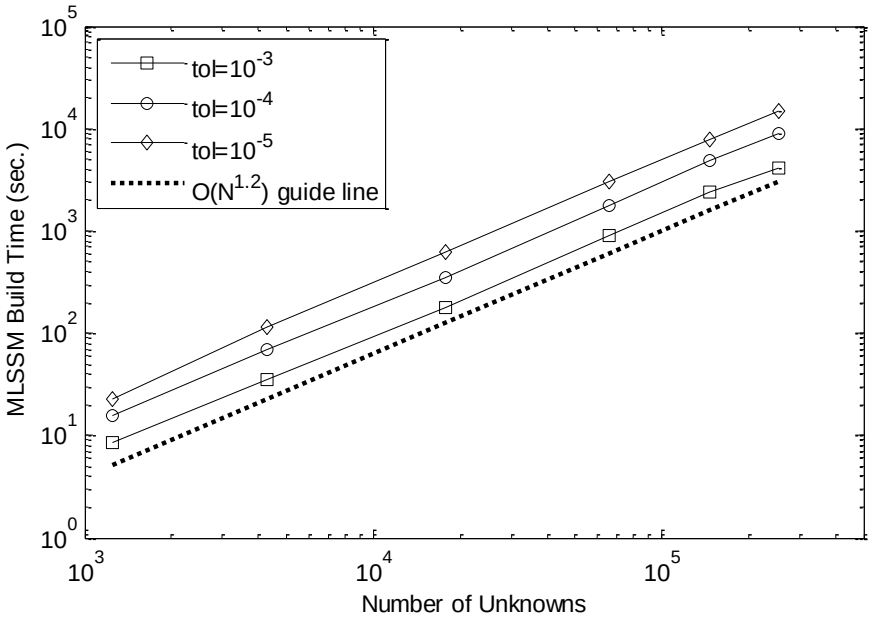


Figure 15. The memory usage of the MLSSM representation of the system matrices for a series of spheres with radii of 2λ, 2.83λ, 4λ and 5.66λ. The density of the mesh provides an average of 300 DOF/λ^2.

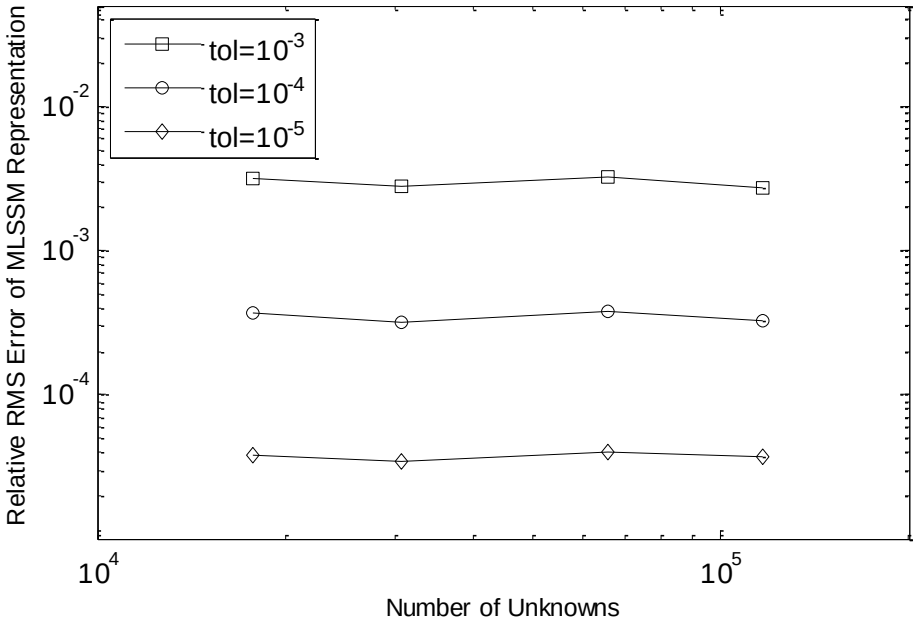


Figure 16. The relative RMS error the MLSSM representation of the system matrix for a PEC sphere with radius 2λ. The EFIE formulation with Nyström discretization is used to obtain the matrix equation. The coarsest mesh has a density of 300 DOF/λ^2.

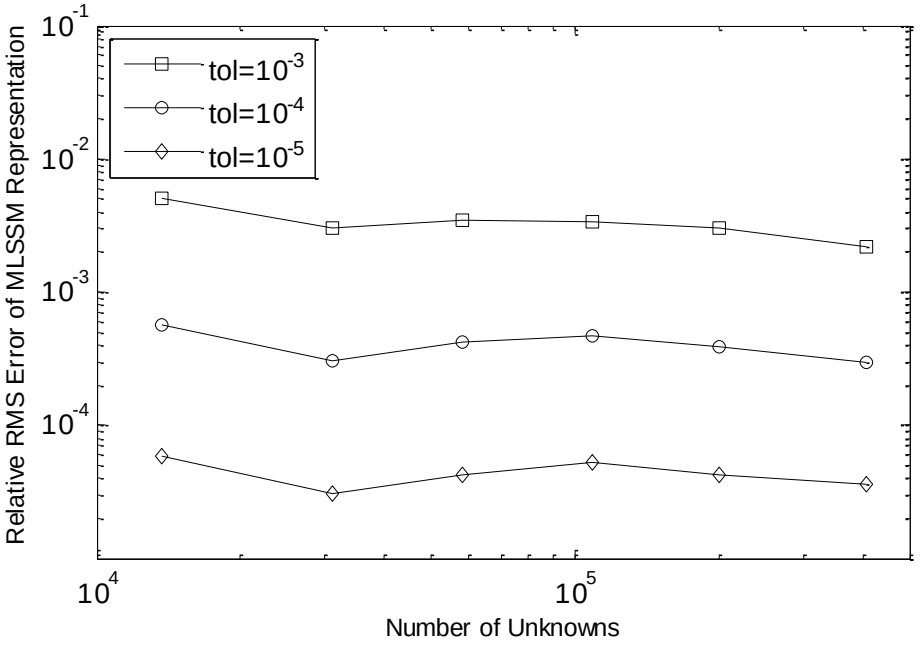


Figure 17. The relative RMS error of the MLSSM representation of the system matrix for a vias-like structure (see Figure 11). The EFIE formulation with Nystr öm discretization is used to obtain the matrix equation.

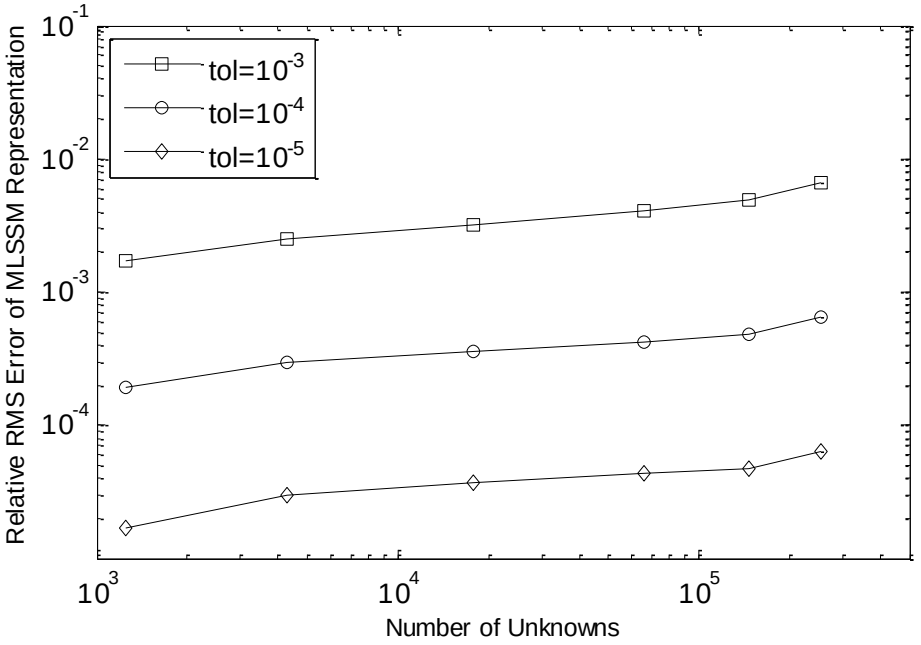


Figure 18. The relative RMS error of the MLSSM representation of the system matrices for a series of spheres with radii of 2λ, 2.83λ, 4λ and 5.66λ. The density of the mesh provides an average of 300 DOF/λ^2.

Chapter 4. LOGOS Factorization

Integral equation-based numerical models provide an important tool for the analysis and design in a variety of application areas. Traditional implementations of surface integral equation (IE) based models lead to dense matrix equations of the form,

$$\mathbf{Z}\mathbf{x} = \mathbf{F}^i \tag{4.1}$$

where the vector $\mathbf{x}$ contains the coefficients of the sub-domain basis functions used to represent equivalent surface currents, and the vector $\mathbf{F}^i$ is determined by samples of the source fields at points on the surface of the underlying geometry. To treat complex problems, it is necessary to use sophisticated methods which rely on compressed representations of the system matrix [3, 69]. Solution methods that rely on these compressed representations can be roughly categorized as either fast iterative solution methods [3] or fast direct solution methods [20, 21, 24, 25, 70-79].

This dissertation works with LOGOS based fast direct solvers. The basic idea of LOGOS modes and the process to obtain these modes are presented in Section 4.1. The LOGOS modes based factorization is also presented. Section 4.2 presents an efficient algorithm, named ΘR factorization, which is a critical step in obtaining LOGOS modes. Section 4.3 discusses how the error of LOGOS factorization can be controlled. Section 4.4 illustrate how to obtain the solution after LOGOS factorization. Finally, a multi-range localization procedure is introduced in Section 4.5 to improve the efficiency of the NL-LOGOS factorization.

4.1. Determination of LOGOS Modes

Let the simulation domain be denoted as S . Decompose S into two non-overlapping pieces,

$$S = S_1 + S_2 \tag{4.2}$$

For convenience, we will refer to S_1 and S_2 as "Region 1" and "Region 2," respectively. This decomposition of S leads to an associated decomposition of system equation satisfied by LOGOS modes as,

$$\begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} \\ \mathbf{Z}_{21} & \mathbf{Z}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{1,m} \\ \mathbf{x}_{2,m} \end{bmatrix} = \begin{bmatrix} \mathbf{F}^i_{1,m} \\ \mathbf{F}^i_{2,m} \end{bmatrix}, \tag{4.3}$$

where the integer subscript m on $\mathbf{x}_m$ and $\mathbf{F}^i_m$ is used to index the LOGOS modes. A single LOGOS mode is defined by an excitation/solution pairing $\left(\mathbf{F}^i_m, \mathbf{x}_m\right)$. The subvector $\mathbf{x}_{1,m}$ is the part of $\mathbf{x}_m$ associated with Region 1, and $\mathbf{x}_{2,m}$ is associated with Region 2. Matrix $\mathbf{Z}_{12}$ corresponds to interactions from Region 2 to 1, etc.

Consider the determination of LOGOS modes $\left(\mathbf{F}^i_m, \mathbf{x}_m\right)$ which have nonzero support only in Region 1 (i.e., $\mathbf{x}_{1,m} \neq 0,\ \mathbf{x}_{2,m} = 0$). The *local* condition associated with these modes is

$$\mathbf{Z}_{11}\mathbf{x}_{1,m} = \mathbf{F}^i_{1,m}\,. \tag{4.4}$$

The *global* condition is

$$\mathbf{Z}_{21}\,\mathbf{x}_{1,m} = \mathbf{F}^i_{2,m}\,. \tag{4.5}$$

Combining (4.4) and (4.5) provides a local-global condition

$$\mathbf{Z}_{21}\,\mathbf{Z}_{11}^{-1}\,\mathbf{F}^i_{1,m} = \mathbf{F}^i_{2,m}\,, \tag{4.6}$$

satisfied by all LOGOS modes. The condition (4.6) can be used to determine LOGOS modes that, to $O(\varepsilon)$, have sources $\mathbf{x}_m$ that are confined to Region 1.

It is useful to introduce two classifications for LOGOS modes: localizing vs. non-localizing, and overlapping vs. non-overlapping. Localizing LOGOS modes are obtained by requiring $\mathbf{F}^i_{2,m} = 0$ in (4.6), which requires that the associated source functions radiate no energy into Region 2. The additional modes obtained in the more general case $\left(\mathbf{F}^i_{2,m} \neq 0\right)$ are referred to as non-localizing LOGOS modes.

The second classification (overlapping vs. non-overlapping) is used to indicate the domain of support of the $\mathbf{x}_{1,m}$. In the case where the support of $\mathbf{x}_{1,m}$ is somewhat larger than Region 1, the support of source functions defined in Region 1 will overlap with the support of sources defined in other spatial regions. Such modes are referred to as overlapping LOGOS modes. The non-overlapping case is obtained when the support is restricted to distinct spatial regions.

Combining these two classifications, we can have 4 types of LOGOS modes:

1) NL-LOGOS modes: non-overlapping localizing LOGOS modes.

2) OL-LOGOS modes: overlapping localizing LOGOS modes.

3) NN-LOGOS modes: non-overlapping non-localizing LOGOS modes.

4) ON-LOGOS modes: overlapping non-localizing LOGOS modes.

This dissertation considers factorizations based only on the NL-LOGOS modes. NL-LOGOS modes in region 1 are obtained by imposing (to order-ε) (4.6) for the case $\mathbf{F}_{2,m}^{i}=0$,

$$\mathbf{Z}_{21}\mathbf{Z}_{11}^{-1}\mathbf{F}_{1,m}^{i}=0. \tag{4.7}$$

These LOGOS modes are designated as "localizing" LOGOS modes because the field scattered from Region 1 to Region 2 is zero to order-ε ($\mathbf{Z}_{21}\mathbf{x}_{1,m}\approx 0$).

To compute localizing modes satisfying (4.7) to order-ε we follow [78] and perform a QR decomposition on the matrix block associated with sources in Region 1,

$$\mathbf{Z}_1=\begin{bmatrix}\mathbf{Z}_{11}\\ \mathbf{Z}_{21}\end{bmatrix}=\begin{bmatrix}\mathbf{Q}_{11}\\ \mathbf{Q}_{21}\end{bmatrix}\mathbf{R}_1, \tag{4.8}$$

where $\mathbf{Q}_{11}$ ($\mathbf{Q}_{21}$) is of the same size as $\mathbf{Z}_{11}$ ($\mathbf{Z}_{21}$), and $\mathbf{R}_1$ is a square upper triangular matrix. A singular value decomposition (SVD) of $\mathbf{Q}_{11}$ provides

$$\mathbf{Q}_{11}=\mathbf{u}_1\mathbf{s}_1\mathbf{v}_1^{H}. \tag{4.9}$$

Let s_i $(i=1,\cdots,n)$ indicate the singular values in $\mathbf{s}_1$ which have been sorted in non-increasing order. The magnitude of these singular values serves as the field strength multiplier by which the corresponding left singular vectors receive field radiated by the corresponding right singular vectors. Because $\mathbf{Q}_{11}$ is part of an orthonormal matrix, $s_i\geq 1$. Thus, the desired, order-ε localizing modes are obtained from those singular vectors corresponding to s_i that are near unity. Let N_L indicate the number of s_i $\left(i=1,\cdots,N_L\right)$ that are sufficiently close to unity to qualify as order-ε localizing modes (the ε-dependent condition used to determine N_L will be discussed below). If the corresponding right singular vector is denoted by $\mathbf{v}_1^{(L)}$ and the rest of the singular vectors

$\mathbf{v}_1^{(N)}$, we have $\mathbf{v}_1 = \left[\mathbf{v}_1^{(L)} \quad \mathbf{v}_1^{(N)}\right]$. The matrix $\mathbf{s}_1$ has a similar representation, $\mathbf{s}_1 = \left[\mathbf{s}_1^{(L)} \quad \mathbf{s}_1^{(N)}\right]$. Therefore,

$$\mathbf{Z}_1 = \begin{bmatrix} \mathbf{u}_1\mathbf{s}_1 \\ \mathbf{Q}_{21}\mathbf{v}_1 \end{bmatrix} \mathbf{v}_1^H \mathbf{R}_1, \tag{4.10}$$

or

$$\mathbf{Z}_1\mathbf{R}_1^{-1}\left[\mathbf{v}_1^{(L)} \quad \mathbf{v}_1^{(N)}\right] = \begin{bmatrix} \mathbf{u}_1\mathbf{s}_1 \\ \mathbf{Q}_{21}\mathbf{v}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1\mathbf{s}_1^{(L)} & \mathbf{u}_1\mathbf{s}_1^{(N)} \\ \mathbf{Q}_{21}\mathbf{v}_1^{(L)} & \mathbf{Q}_{21}\mathbf{v}_1^{(N)} \end{bmatrix}, \tag{4.11}$$

Because $\mathbf{v}_1$ is unitary, the column vectors of

$$\begin{bmatrix} \mathbf{u}_1\mathbf{s}_1 \\ \mathbf{Q}_{21}\mathbf{v}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{11} \\ \mathbf{Q}_{21} \end{bmatrix} \mathbf{v}_1$$

are orthonormal. The large singular values (approaching unity) in $\mathbf{s}_1^{(L)}$ indicate small values in $\mathbf{Q}_{21}\mathbf{v}_1^{(L)}$. The value N_L is determined such that the following approximation can be made

$$\mathbf{Q}_{21}\mathbf{v}_1^{(L)} \approx \mathbf{0}, \tag{4.12}$$

and

$$\mathbf{Z}_1\mathbf{R}_1^{-1}\left[\mathbf{v}_1^{(L)} \quad \mathbf{v}_1^{(N)}\right] = \mathbf{Z}_1\left[\mathbf{\Lambda}_1^{(L)} \quad \mathbf{R}_1^{-1}\mathbf{v}_1^{(N)}\right] \approx \begin{bmatrix} \mathbf{u}_1\mathbf{s}_1^{(L)} & \mathbf{u}_1\mathbf{s}_1^{(N)} \\ \mathbf{0} & \mathbf{Q}_{21}\mathbf{v}_1^{(N)} \end{bmatrix}. \tag{4.13}$$

In (4.13), $\mathbf{\Lambda}_1^{(L)} = \mathbf{R}_1^{-1}\mathbf{v}^{(L)}$ contains the localizing LOGOS source modes, which radiate approximately no field into Region 2. The criteria for determining N_L, which controls the error in the LOGOS factorization, is discussed in [80] and the details will be provided in Section 4.3. The same procedure can be used to find localizing source modes, $\mathbf{\Lambda}_2^{(L)}$, in Region 2 which radiate approximately no field into Region 1. $\mathbf{\Lambda}_1^{(N)}$ and $\mathbf{\Lambda}_2^{(N)}$ are assigned as the orthonormal complements of $\mathbf{\Lambda}_1^{(L)}$ and $\mathbf{\Lambda}_2^{(L)}$, respectively. Therefore, we have

$$\begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} \\ \mathbf{Z}_{21} & \mathbf{Z}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_1^{(L)} & \mathbf{0} & \mathbf{\Lambda}_1^{(N)} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_2^{(L)} & \mathbf{0} & \mathbf{\Lambda}_2^{(N)} \end{bmatrix} \approx \begin{bmatrix} \mathbf{u}_1\mathbf{s}_1^{(L)} & \mathbf{0} & \mathbf{Z}_{11}^{(N)} & \mathbf{Z}_{12}^{(N)} \\ \mathbf{0} & \mathbf{u}_2\mathbf{s}_2^{(L)} & \mathbf{Z}_{21}^{(N)} & \mathbf{Z}_{22}^{(N)} \end{bmatrix}. \tag{4.14}$$

The matrix blocks on the RHS of (4.14) have been defined by matching their positions with the matrix multiplies on the LHS. Let $\mathbf{P}_1^{(L)} = \mathbf{u}_1$ and $\mathbf{P}_2^{(L)} = \mathbf{u}_2$ span the localized field space (they are orthonormal). The projection matrix is composed of $\mathbf{P}_1^{(L)}$, $\mathbf{P}_2^{(L)}$ and their orthonormal complements $\mathbf{P}_1^{(N)}$ and $\mathbf{P}_2^{(N)}$. The complete one level factorization is

$$\begin{bmatrix} \left(\mathbf{P}^{(L)}\right)^H \\ \left(\mathbf{P}^{(N)}\right)^H \end{bmatrix} \begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} \\ \mathbf{Z}_{21} & \mathbf{Z}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}^{(L)} & \mathbf{\Lambda}^{(N)} \end{bmatrix} \approx \begin{bmatrix} \tilde{\mathbf{I}} & \mathbf{Z}^{(LN)} \\ \mathbf{0} & \mathbf{Z}^{(NN)} \end{bmatrix}, \tag{4.15}$$

where $\mathbf{\Lambda}^{(L)} = \begin{bmatrix} \mathbf{\Lambda}_1^{(L)} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_2^{(L)} \end{bmatrix}$, and similar definitions are used for $\mathbf{\Lambda}^{(N)}$, $\mathbf{P}^{(L)}$ and $\mathbf{P}^{(N)}$. $\tilde{\mathbf{I}} = \begin{bmatrix} \mathbf{s}_1^{(L)} & \mathbf{0} \\ \mathbf{0} & \mathbf{s}_2^{(L)} \end{bmatrix}$ is a matrix with only approximately unity diagonal elements. The factorization can be written as

$$\begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} \\ \mathbf{Z}_{21} & \mathbf{Z}_{22} \end{bmatrix} \approx \begin{bmatrix} \mathbf{P}^{(L)} & \mathbf{P}^{(N)} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{I}} & \mathbf{Z}^{(LN)} \\ \mathbf{0} & \mathbf{Z}^{(NN)} \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}^{(L)} & \mathbf{\Lambda}^{(N)} \end{bmatrix}^{-1}. \tag{4.16}$$

If the underlying geometry is decomposed by a multilevel tree, the groups in each level of the tree define the regions. At the finest level (level-L) of the tree the factorization is applied to the original system matrix, $\mathbf{Z} = \mathbf{Z}_L = \mathbf{Z}_{L+1}^{(NN)}$, and (4.16) becomes

$$\mathbf{Z}_{L+1}^{(NN)} \approx \begin{bmatrix} \mathbf{P}_L^{(L)} & \mathbf{P}_L^{(N)} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{I}} & \mathbf{Z}_L^{(LN)} \\ \mathbf{0} & \mathbf{Z}_L^{(NN)} \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_L^{(L)} & \mathbf{\Lambda}_L^{(N)} \end{bmatrix}^{-1}. \tag{4.17}$$

At a coarser level of the tree, (4.17) looks the same. Therefore, the same factorization can be carried out recursively on $\mathbf{Z}_l^{(NN)}$ for $l = L, \cdots, 2$.

4.2. Sparse ΘR Factorization of Z

The most time consuming operation in the foregoing discussion is the QR factorization in (4.8) and the SVD in (4.9). The QR factorization in (4.8) dominates the cost because the matrix block $\mathbf{Z}_1$ is an N-by-m block, where N is the number of degrees of freedom (DOF) in the problem and m indicates the number of DOF in Region 1.

To avoid the large costs of this QR decomposition in computing the NL-LOGOS modes, it is necessary to define a more efficient strategy for obtaining the same information by working with a sparse representation of $\mathbf{Z}$. In the following, this is accomplished using the ΘR factorization of the sparse MLSSM representation of $\mathbf{Z}$.

4.2.1. Definition of ΘR Factorization

Let $\mathbf{Z}_{i(l)}$ denote the columns of $\mathbf{Z}$ associated with sources in the level-l group, $i(l)$. The matrix $\mathbf{Z}$ can then be represented as the horizontal concatenation of these column blocks,

$$\mathbf{Z} = \left[\mathbf{Z}_{1(l)}, \ldots, \mathbf{Z}_{i(l)}, \ldots, \mathbf{Z}_{M(l)} \right]. \tag{4.18}$$

Performing a QR factorization [15] of each column block on the right side of (4.18),

$$\mathbf{Z}_{i(l)} = \mathbf{q}_{i(l)} \mathbf{r}_{i(l)}. \tag{4.19}$$

Assembling the results yields the representation

$$\mathbf{Z} = \left[\mathbf{q}_{1(l)}, \ldots, \mathbf{q}_{i(l)}, \ldots, \mathbf{q}_{M(l)} \right] \mathrm{diag}\left(\mathbf{r}_{i(l)} \right). \tag{4.20}$$

In (4.20), the notation $\mathrm{diag}\left(\mathbf{r}_{i(l)} \right)$ is used to indicate the block diagonal matrix having the square matrices $\mathrm{r}_{i(l)}$ along its diagonal.

With the definitions

$$\mathbf{\Theta}_l = \left[\mathbf{q}_{1(l)}, \ldots, \mathbf{q}_{i(l)}, \ldots, \mathbf{q}_{M(l)} \right], \tag{4.21}$$

$$\mathbf{R}_l = \mathrm{diag}\left(\mathbf{r}_{i(l)} \right), \tag{4.22}$$

Equation (4.20) can be written as

$$\mathbf{Z} = \mathbf{\Theta}_l \mathbf{R}_l. \tag{4.23}$$

The factorization given by (4.23) defines the so called ΘR factorization of $\mathbf{Z}$ according to the level-l group partitioning. Here, it is simply referred to as ΘR factorization of $\mathbf{Z}$ at level-l or level-l ΘR factorization of $\mathbf{Z}$.

4.2.2. Basic Idea

Suppose we want to find the ΘR Factorization of (3.17). First, partition the system matrix according to level-4 source groups as

$$\mathbf{Z} = \left[\mathbf{Z}_{1(4)}, \cdots, \mathbf{Z}_{i(4)}, \cdots, \mathbf{Z}_{M(4)}\right]. \tag{4.24}$$

According to (3.17), the different terms in the MLSSM representation represent different portions of the matrix as shown by Figure 6. The portions that make up $\mathbf{Z}_{1(4)}$ can be written as

$$\mathbf{Z}_{1(4)} = \hat{\mathbf{Z}}_{1(4)} + \mathbf{U}_4 \hat{\mathbf{Z}}_3 \mathbf{V}_{1(4)}^H + \mathbf{U}_4 \mathbf{U}_3 \mathbf{Z}_2 \mathbf{V}_3^H \mathbf{V}_{1(4)}^H, \tag{4.25}$$

where each term is indicated in Figure 6 using thickened lines. $\mathbf{V}_{1(4)}^H$ is indicated using thickened lines in Figure 7. Equation (4.25) can also be written as the vertical concatenation of the three matrix terms on its RHS:

$$\mathbf{Z}_{1(4)} = \left[\hat{\mathbf{Z}}_{1(4)}; \mathbf{U}_4 \hat{\mathbf{Z}}_3 \mathbf{V}_{1(4)}^H; \mathbf{U}_4 \mathbf{U}_3 \mathbf{Z}_2 \mathbf{V}_3^H \mathbf{V}_{1(4)}^H\right], \tag{4.26}$$

where ";" indicates vertical concatenation. In (4.26) and the following discussions, the three matrix blocks are used to represent their non-zero parts only (the shaded parts with thick boxes in Figure 6). A standard QR factorization of $\mathbf{Z}_{1(4)}$ would provide the desired factor, $\mathbf{R}_{1(4)}$. However, this is not efficient because it operates on a full matrix. Alternatively, we can find the **Q** and **R** factors for each of the blocks in (4.26), so that

$$\begin{aligned} \mathbf{Z}_{1(4)} &= \left[\mathbf{Q}_a \mathbf{R}_a; \mathbf{Q}_b \mathbf{R}_b; \mathbf{Q}_c \mathbf{R}_c\right] \\ &= \begin{bmatrix} \mathbf{Q}_a & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_b & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_c \end{bmatrix} \begin{bmatrix} \mathbf{R}_a \\ \mathbf{R}_b \\ \mathbf{R}_c \end{bmatrix}, \end{aligned} \tag{4.27}$$

Another QR factorization on the matrix containing only **R** factors in (4.27) gives

$$\mathbf{Z}_{1(4)} = \begin{bmatrix} \mathbf{Q}_a & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_b & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_c \end{bmatrix} \mathbf{Q}' \mathbf{R}_{1(4)} = \mathbf{Q}_{1(4)} \mathbf{R}_{1(4)}. \tag{4.28}$$

Equation (4.28) defines the QR factorization of $\mathbf{Z}_{1(4)}$ because

$$\mathbf{Q}_{1(4)} = \begin{bmatrix} \mathbf{Q}_a & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_b & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_c \end{bmatrix} \mathbf{Q}' \tag{4.29}$$

is column orthonormal. We have used the fact that the product of two column orthonormal matrices is column orthonormal. Therefore, if only the factor **R** is needed,

which can be found using $\mathbf{R}_a$, $\mathbf{R}_b$ and $\mathbf{R}_c$ only, the matrices $\mathbf{Q}_a$, $\mathbf{Q}_b$, $\mathbf{Q}_c$ and $\mathbf{Q}'$ can be discarded in the process of finding $\mathbf{R}_{1(4)}$. Based on the same arguments, it is sufficient to find $\mathbf{R}_b$ and $\mathbf{R}_c$ from $\hat{\mathbf{Z}}_3\mathbf{V}_{1(4)}^H$ and $\mathbf{Z}_2\mathbf{V}_3^H\mathbf{V}_{1(4)}^H$, respectively, since the $\mathbf{U}$ matrices in the MLSSM is column orthonormal. The matrices $\hat{\mathbf{Z}}_3\mathbf{V}_{1(4)}^H$ and $\mathbf{Z}_2\mathbf{V}_3^H\mathbf{V}_{1(4)}^H$ can be significantly smaller than $\mathbf{U}_4\hat{\mathbf{Z}}_3\mathbf{V}_{1(4)}^H$ and $\mathbf{U}_4\mathbf{U}_3\mathbf{Z}_2\mathbf{V}_3^H\mathbf{V}_{1(4)}^H$ due to the compression used in the MLSSM representation. Therefore, the efficiency of finding the factor $\mathbf{R}_{1(4)}$ is improved.

The efficiency of finding $\mathbf{R}_b$ from $\hat{\mathbf{Z}}_3\mathbf{V}_{1(4)}^H$ can also be improved. Assume level-4 ΘR factorization of $\hat{\mathbf{Z}}_3$ is $\hat{\mathbf{Z}}_3 = \hat{\mathbf{\Theta}}_4\hat{\mathbf{R}}_4$. $\mathbf{R}_b$ can be found by apply QR factorization on $\hat{\mathbf{R}}_{1(4)}\mathbf{V}_{1(4)}^H$. Similar techniques can be used to find $\mathbf{R}_c$ from $\mathbf{Z}_2\mathbf{V}_3^H\mathbf{V}_{1(4)}^H$.

This strategy for finding $\mathbf{R}_{1(4)}$ from the MLSSM representation of $\mathbf{Z}_{1(4)}$ always operates on compressed data and has two important properties. First, no matrix blocks need to be extracted/decompressed from the sparse MLSSM representation. Furthermore, no additional approximation to the data structure is required to extract the required $\mathbf{R}_{1(4)}$ factors.

The fact that the product of two column orthonormal matrices is column orthonormal will be used extensively in the following derivation of the sparse ΘR factorization algorithm. This allows the column orthonormal matrices on the left of the factorization to be discarded if only the R factors are needed. In the following, the ΘR factorization algorithm is introduced in further detail. This discussion primarily consists of the systematic, multilevel application of the ideas indicated by (4.24) to (4.29).

4.2.3. ΘR Factorization Algorithm

As shown in Section 4.2.2, given the $\mathbf{R}_l$ matrix of (4.23) and the level-l MLSSM matrix $\hat{\mathbf{Z}}_l$, NL-LOGOS modes can be rapidly computed (The full matrix $\mathbf{\Theta}_l$ is not needed to compute the NL-LOGOS modes). Thus, the ΘR factorization described here only explicitly computes the block-diagonal R factor, and the matrix $\mathbf{\Theta}_l$ is never

explicitly formed. Our discussion of the algorithm will primarily focus on the level-L implementation since the multilevel generalization is obvious.

Finally, in anticipation of additional applications of the proposed algorithms, we consider the slightly more general problem of determining the ΘR factorization of a projected version of the MLSSM representation of the impedance matrix. In particular, let $\mathbf{P}_L$ indicate an arbitrary level-L block diagonal matrix (i.e., the diagonal blocks span only a single group at level-L). The ΘR factorization of $\mathbf{P}_L^H\mathbf{Z}$ at level-L is given by

$$\mathbf{P}_L^H\mathbf{Z} = \mathbf{\Theta}_L\mathbf{R}_L . \tag{4.30}$$

If $\mathbf{P}_L = \mathbf{I}$, then (4.30) indicates a ΘR factorization of $\mathbf{Z}$.

The proposed sparse ΘR factorization of $\mathbf{P}_L^H\mathbf{Z}$ requires traversing the multilevel tree twice, first from level-L to level-2 (upward pass), and then from level-2 back to level-L (downward pass). In some cases, only the downward pass is required. However, for completeness, we discuss both the upward and downward pass in Sections 4.2.3.1 and 4.2.3.2, respectively.

4.2.3.1. ΘR – Upward Pass

The upward pass is necessary only when there is a non-column-orthonormal matrix multiplying from the left onto the MLSSM representation of the system matrix. Beginning at the finest level, equation (3.14) allows $\mathbf{P}_L^H\mathbf{Z}$ to be expressed as

$$\mathbf{P}_L^H\mathbf{Z} = \mathbf{P}_L^H\left(\hat{\mathbf{Z}}_L + \mathbf{U}_L\mathbf{Z}_{L-1}\mathbf{V}_L^H\right). \tag{4.31}$$

Due to the nested structure of MLSSM, (4.31) can also be written as

$$\begin{aligned}\mathbf{P}_L^H\mathbf{Z} = {} & \mathbf{P}_L^H\hat{\mathbf{Z}}_L \\ & + \mathbf{P}_L^H\mathbf{U}_L\hat{\mathbf{Z}}_{L-1}\mathbf{V}_L^H \\ & + \mathbf{P}_L^H\mathbf{U}_L\mathbf{U}_{L-1}\hat{\mathbf{Z}}_{L-2}\mathbf{V}_{L-1}^H\mathbf{V}_L^H \\ & + \cdots \\ & + \mathbf{P}_L^H\mathbf{U}_L\mathbf{U}_{L-1}\cdots\mathbf{U}_3\hat{\mathbf{Z}}_2\mathbf{V}_3^H\cdots\mathbf{V}_{L-1}^H\mathbf{V}_L^H\end{aligned} . \tag{4.32}$$

The upward pass works with a part of each term in the RHS of (4.32) and factors it such that

$$\mathbf{P}_L^H = \mathbf{I}\times\tilde{\mathbf{P}}_L^H , \tag{4.33}$$

$$\mathbf{P}_L^H \mathbf{U}_L = \tilde{\mathbf{U}}_L \tilde{\mathbf{P}}_{L-1}^H, \tag{4.34}$$

$$\mathbf{P}_L^H \mathbf{U}_L \mathbf{U}_{L-1} = \tilde{\mathbf{U}}_L \tilde{\mathbf{U}}_{L-1} \tilde{\mathbf{P}}_{L-2}^H, \tag{4.35}$$

$$\mathbf{P}_L^H \mathbf{U}_L \mathbf{U}_{L-1} \cdots \mathbf{U}_3 = \tilde{\mathbf{U}}_L \tilde{\mathbf{U}}_{L-1} \cdots \tilde{\mathbf{U}}_3 \tilde{\mathbf{P}}_2^H, \tag{4.36}$$

where $\tilde{\mathbf{U}}_l$ is block diagonal and column orthonormal and $\tilde{\mathbf{P}}_l^H$ is block diagonal. The purpose here is to factor out matrices that are column orthonormal and leave the non-orthonormal matrices to the right of the factorization, as indicated by (4.33) to (4.36).

Equation (4.33) is trivial. (4.34) can be obtained by applying the ΘR factorization at level-L to the matrix $\mathbf{P}_L^H \mathbf{U}_L$. The $\mathbf{\Theta}_L$ factor is saved in $\tilde{\mathbf{U}}_L$ and the $\mathbf{R}_L$ factor is saved in $\tilde{\mathbf{P}}_{L-1}^H$. While it is block-diagonal at level-L, the matrix $\tilde{\mathbf{P}}_{L-1}^H$ is labeled with the subscript $L-1$ in anticipation of its use below.

With (4.34), the LHS of (4.35) can now be written as

$$\mathbf{P}_L^H \mathbf{U}_L \mathbf{U}_{L-1} = \tilde{\mathbf{U}}_L \tilde{\mathbf{P}}_{L-1}^H \times \mathbf{U}_{L-1}. \tag{4.37}$$

The ΘR factorization at level-$(L-1)$ can be applied to $\tilde{\mathbf{P}}_{L-1}^H \mathbf{U}_{L-1}$, with the $\mathbf{\Theta}_{L-1}$ factor saved in $\tilde{\mathbf{U}}_{L-1}$ and $\mathbf{R}_{L-1}$ saved in $\tilde{\mathbf{P}}_{L-2}^H$. Then (4.37) becomes

$$\mathbf{P}_L^H \mathbf{U}_L \mathbf{U}_{L-1} = \tilde{\mathbf{U}}_L \tilde{\mathbf{U}}_{L-1} \tilde{\mathbf{P}}_{L-2}^H. \tag{4.38}$$

Following the same procedure as given by (4.37) and (4.38), the upward pass can be carried out until level-3.

4.2.3.2. ΘR – Downward Pass

After the preceding upward pass, we have

$$\begin{aligned} \mathbf{P}_L^H \mathbf{Z} = & \tilde{\mathbf{P}}_L^H \hat{\mathbf{Z}}_L \\ & + \tilde{\mathbf{U}}_L \tilde{\mathbf{P}}_{L-1}^H \hat{\mathbf{Z}}_{L-1} \mathbf{V}_L^H \\ & + \tilde{\mathbf{U}}_L \tilde{\mathbf{U}}_{L-1} \tilde{\mathbf{P}}_{L-2}^H \hat{\mathbf{Z}}_{L-2} \mathbf{V}_{L-1}^H \mathbf{V}_L^H \\ & + \cdots \\ & + \tilde{\mathbf{U}}_L \tilde{\mathbf{U}}_{L-1} \cdots \tilde{\mathbf{U}}_3 \tilde{\mathbf{P}}_2^H \hat{\mathbf{Z}}_2 \mathbf{V}_3^H \cdots \mathbf{V}_{L-1}^H \mathbf{V}_L^H \end{aligned} . \tag{4.39}$$

Equation (4.39) can be written in the same format as the MLSSM representation of $\mathbf{Z}$ if $\tilde{\mathbf{P}}_l^H \hat{\mathbf{Z}}_l$ is taken as $\hat{\mathbf{Z}}_l$. A recursive relation similar to (3.14) can be written as

$$\mathbf{P}_l^H \mathbf{Z}_l = \tilde{\mathbf{P}}_l^H \hat{\mathbf{Z}}_l + \tilde{\mathbf{U}}_l \tilde{\mathbf{P}}_{l-1}^H \mathbf{Z}_{l-1} \mathbf{V}_l^H. \tag{4.40}$$

The downward pass of sparse ΘR factorization works on (4.40) and produces results (4.30) without explicitly forming and saving $\mathbf{\Theta}_L$.

The following procedures are divided into two steps: Step-I is to find the following ΘR factorization

$$\mathbf{P}_l^H \mathbf{Z}_l = \bar{\mathbf{\Theta}}_{l+1} \bar{\mathbf{R}}_{l+1}, \tag{4.41}$$

which is the factorization of level-l matrix $\mathbf{P}_l^H \mathbf{Z}_l$ at level-$(l+1)$; Step-II is to find the ΘR factorization

$$\mathbf{P}_L^H \mathbf{Z}_L = \mathbf{\Theta}_L \mathbf{R}_L, \tag{4.42}$$

which is the ΘR factorization of the finest level (level-L) matrix $\mathbf{P}_L^H \mathbf{Z}_L$ at the finest level. Step-I is used to find the $\mathbf{R}$ factors recursively from the level-2 to the second finest level. Step-II is used to find the $\mathbf{R}$ factors at the finest level.

a) Step-I of Downward Pass

Performing ΘR factorization on the matrix $\tilde{\mathbf{P}}_2^H \mathbf{Z}_2$ at level-3, we have

$$\tilde{\mathbf{P}}_2^H \mathbf{Z}_2 = \bar{\mathbf{\Theta}}_3 \bar{\mathbf{R}}_3. \tag{4.43}$$

At level-3,

$$\tilde{\mathbf{P}}_3^H \mathbf{Z}_3 = \tilde{\mathbf{P}}_3^H \hat{\mathbf{Z}}_3 + \tilde{\mathbf{U}}_3 \tilde{\mathbf{P}}_2^H \mathbf{Z}_2 \mathbf{V}_3^H. \tag{4.44}$$

Substituting (4.43) into (4.44), it becomes

$$\tilde{\mathbf{P}}_3^H \mathbf{Z}_3 = \tilde{\mathbf{P}}_3^H \hat{\mathbf{Z}}_3 + \tilde{\mathbf{U}}_3 \bar{\mathbf{\Theta}}_3 \bar{\mathbf{R}}_3 \mathbf{V}_3^H. \tag{4.45}$$

Applying ΘR factorization at level-4 to $\tilde{\mathbf{P}}_3^H \hat{\mathbf{Z}}_3$ and $\bar{\mathbf{R}}_3 \mathbf{V}_3^H$, we have

$$\tilde{\mathbf{P}}_3^H \hat{\mathbf{Z}}_3 = \hat{\mathbf{\Theta}}_4 \hat{\mathbf{R}}_4, \tag{4.46}$$

$$\bar{\mathbf{R}}_3 \mathbf{V}_3^H = \tilde{\mathbf{V}}_4^H \tilde{\mathbf{R}}_4. \tag{4.47}$$

where

$$\hat{\mathbf{\Theta}}_4 \hat{\mathbf{R}}_4 = \left[\hat{\mathrm{q}}_{1(4)}, \ldots, \hat{\mathrm{q}}_{M(4)} \right] \begin{bmatrix} \hat{\mathbf{r}}_{1(4)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{r}}_{M(4)} \end{bmatrix}. \tag{4.48}$$

With (4.46) and (4.47), (4.45) becomes

$$\tilde{\mathbf{P}}_3^H \mathbf{Z}_3 = \hat{\mathbf{\Theta}}_4 \hat{\mathbf{R}}_4 + \tilde{\mathbf{U}}_3 \bar{\mathbf{\Theta}}_3 \tilde{\mathbf{V}}_4^H \tilde{\mathbf{R}}_4 . \tag{4.49}$$

Let $\tilde{\mathbf{\Theta}}_4 = \tilde{\mathbf{U}}_3 \bar{\mathbf{\Theta}}_3 \tilde{\mathbf{V}}_4^H$. It can be shown that $\tilde{\mathbf{\Theta}}_4$ has the structure

$$\tilde{\mathbf{\Theta}}_4 = \left[\tilde{\mathbf{q}}_{1(4)}, \ldots, \tilde{\mathbf{q}}_{i(4)}, \ldots, \tilde{\mathbf{q}}_{M(4)} \right], \tag{4.50}$$

where the column blocks associated with the level-4 groups are orthonormal. Equation (4.49) can be simplified as

$$\tilde{\mathbf{P}}_3^H \mathbf{Z}_3 = \hat{\mathbf{\Theta}}_4 \hat{\mathbf{R}}_4 + \tilde{\mathbf{\Theta}}_4 \tilde{\mathbf{R}}_4 . \tag{4.51}$$

For a single level-4 source group, $i(4)$, the right side of (4.51) can be represented as

$$\hat{\mathbf{\Theta}}_4 \hat{\mathbf{R}}_4 + \tilde{\mathbf{\Theta}}_4 \tilde{\mathbf{R}}_4 \xrightarrow{\text{group } i(4)} \hat{\mathbf{q}}_{i(4)} \hat{\mathbf{r}}_{i(4)} + \tilde{\mathbf{q}}_{i(4)} \tilde{\mathbf{r}}_{i(4)} . \tag{4.52}$$

Again, $\hat{\mathbf{q}}_{i(4)}$ and $\tilde{\mathbf{q}}_{i(4)}$ are column orthonormal matrices. Furthermore, due to arguments similar to those used to obtain (4.28) from (4.25), the right side of (4.52) can be factored as

$$\hat{\mathbf{q}}_{i(4)} \hat{\mathbf{r}}_{i(4)} + \tilde{\mathbf{q}}_{i(4)} \tilde{\mathbf{r}}_{i(4)} = [\hat{\mathbf{q}}_{i(4)}, \tilde{\mathbf{q}}_{i(4)}] \begin{bmatrix} \hat{\mathbf{r}}_{i(4)} \\ \tilde{\mathbf{r}}_{i(4)} \end{bmatrix} = [\hat{\mathbf{q}}_{i(4)}, \tilde{\mathbf{q}}_{i(4)}] \mathbf{q}_{i(4)} \mathbf{r}_{i(4)} = \bar{\mathbf{q}}_{i(4)} \bar{\mathbf{r}}_{i(4)} , \tag{4.53}$$

where $\bar{\mathbf{q}}_{i(4)} = [\hat{\mathbf{q}}_{i(4)}, \tilde{\mathbf{q}}_{i(4)}] \, \mathbf{q}_{i(4)}$ is orthonormal.

Performing the factorization indicated by (4.53) for each level-3 group, the right side of (4.51) can be put in the desired form,

$$\tilde{\mathbf{P}}_3^H \mathbf{Z}_3 = \hat{\mathbf{\Theta}}_4 \hat{\mathbf{R}}_4 + \tilde{\mathbf{\Theta}}_4 \tilde{\mathbf{R}}_4 = \bar{\mathbf{\Theta}}_4 \bar{\mathbf{R}}_4 , \tag{4.54}$$

where $\bar{\mathbf{R}}_4$ is a level-4 block diagonal matrix with the diagonal blocks $\bar{\mathbf{r}}_{i(4)}$. The operations from (4.45) to (4.54) can be repeated to obtain (4.41). During the process of factorization, the $\mathbf{Q}$ matrices obtained from all QR factorizations can be discarded since the $\mathbf{\Theta}$ matrices are not ultimately required for the NL-LOGOS mode computation.

b) Step-II of Downward Pass

After the factorizations in Step-I, we have

$$\mathbf{P}_L^H \mathbf{Z}_L = \tilde{\mathbf{P}}_L^H \hat{\mathbf{Z}}_L + \tilde{\mathbf{U}}_L \bar{\mathbf{\Theta}}_L \bar{\mathbf{R}}_L \mathbf{V}_L^H . \tag{4.55}$$

Similar to (4.46) and (4.47), we can write

$$\tilde{\mathbf{P}}_L^H \hat{\mathbf{Z}}_L = \hat{\mathbf{\Theta}}_L \hat{\mathbf{R}}_L , \tag{4.56}$$

$$\bar{\mathbf{R}}_L \mathbf{V}_L^H = \tilde{\mathbf{V}}_L^H \tilde{\mathbf{R}}_L . \tag{4.57}$$

Thus, (4.55) becomes

$$\begin{aligned} \mathbf{P}_L^H \mathbf{Z}_L &= \hat{\mathbf{\Theta}}_L \hat{\mathbf{R}}_L + \tilde{\mathbf{U}}_L \bar{\mathbf{\Theta}}_L \tilde{\mathbf{V}}_L^H \tilde{\mathbf{R}}_L \\ &= \hat{\mathbf{\Theta}}_L \hat{\mathbf{R}}_L + \tilde{\mathbf{\Theta}}_L \tilde{\mathbf{R}}_L \\ &= \mathbf{\Theta}_L \mathbf{R}_L \end{aligned} . \tag{4.58}$$

In obtaining (4.58), arguments similar to those used in deriving (4.51) and (4.54) are used here.

4.2.4. Complexity of ΘR Factorization

The proposed procedure for determining (4.30) via (4.31) to (4.58) relies only on manipulation of diagonal and near-neighbor blocks of the MLSSM matrices $\mathbf{U}_l$, $\mathbf{V}_l$ and $\hat{\mathbf{Z}}_l$ at level-l. The full system matrix is never reconstructed. Thus, the cost of the ΘR procedure is determined by the size of the blocks in $\mathbf{U}_l$, $\mathbf{V}_l$ and $\hat{\mathbf{Z}}_l$ at all levels. Because the MLSSM provides an $O(N)$ representation of the impedance matrix at low frequencies [77], the cost to obtain (4.30) from the MLSSM representation of $\mathbf{P}_L^H \mathbf{Z}$ is expected to require $O(N)$ floating point operations. Furthermore, because the proposed factorization procedure does not require explicit computation and storage of the $\mathbf{\Theta}$ factors, the memory complexity is also expected to scale as $O(N)$.

4.2.5. Using ΘR Factorization to Determine NL-LOGOS Modes

Because the sparse ΘR factorization never stores the $\mathbf{\Theta}$ factors, all necessary quantities needed in the LOGOS factorization must be computed through the MLSSM representation of the system matrix and the $\mathbf{R}$ factors obtained from the ΘR factorization. To illustrate how this can be accomplished, consider the simple illustration given in Section 4.1. As can be seen from (4.8) and (4.9), the only extra information required is the matrix $\mathbf{Q}_{11}$, which is part of the unavailable $\mathbf{\Theta}$ matrix. Fortunately, it is easy to compute this matrix block using

$$\mathbf{Q}_{11} = \mathbf{Z}_{11} \mathbf{R}_1^{-1} . \tag{4.59}$$

Once $\mathbf{Q}_{11}$ is obtained, the LOGOS factorization discussed in Section 4.1 and in [78] can be continue without additional modifications. Therefore, an efficient ΘR factorization effectively avoids the bottleneck associated with NL-LOGOS mode calculation.

4.2.6. Numerical Examples and Discussions

In the following numerical examples, the NL-LOGOS factorization and solution methods based on the sparse ΘR factorization are applied to solve matrix equations obtained from a Nyström discretization of the electric field integral equation (EFIE) [35, 43]. The MLSSM representation of the system matrix is obtained according to [63]. A detailed analysis of the error control provided by the NL-LOGOS factorization algorithm has recently been performed[80], and the results of that analysis are used in the following examples.

The first case considered is plane wave scattering by long thin PEC strips. The strips are infinitely thin with length 200λ. The width of the strips are 0.2λ, 0.1λ, 0.05λ, 0.025λ and 0.0125λ. The strips are meshed using square cells with the sides of the cells the same as the width of the strips. Therefore, the number of cells on the strips are 1000, 2000, 4000, 8000 and 16000, respectively. The CPU time used for (i) ΘR factorization at the finest level of the oct-tree, and (ii) the full NL-LOGOS factorization is plotted in Figure 19. All curves in Figure 19 scale as about $O(N)$ because the strips have planer structure with large spatial extent in one dimension. Most of the localized LOGOS modes can be found at the finer levels of the trees. Therefore, the coarser level LOOGS factorization takes relatively little time, and the LOGOS factorization has a complexity similar to the ΘR factorization. Different tolerances are considered for the LOGOS factorization, and the corresponding MLSSM representations are built using a tolerance which is 10 times tighter than the tolerance used for the LOGOS factorization. Because the tolerance controls the sparsity of the MLSSM representation, which affects the efficiency of the sparse ΘR factorization, the CPU time used by the sparse ΘR factorization depends on the MLSSM tolerance as can be seen in Figure 19 (Note that sparse ΘR factorization involves no approximations).

The second example is a PEC sphere with radius 2λ. The sphere is meshed with DOF densities ranging from 300 DOFs/λ^2 to 4800 DOFs/λ^2. The CPU times for the ΘR

factorization at the finest level of the oct-tree and the full LOGOS factorization are plotted in Figure 20. Again, the CPU time for the finest level sparse ΘR factorization has $O(N)$ complexity. However, the CPU time for the multi-level LOGOS factorization has $O(N^{2.1})$ complexity. The higher complexity of the multi-level LOGOS factorization results from the fact that there are many non-localized modes at the finer levels of the tree. These modes increases the cost of LOGOS factorization at coarser levels. References [77, 78] describe a more asymptotically efficient OL-LOGOS factorization method for such low frequency problems using overlapped, localizing LOGOS modes. However, as described in [15], the sparse ΘR factorization is also required to expedite the calculation of OL-LOGOS modes.

A third case is a vias-like structure (see Figure 11) which contains multiple PEC strips floating on both sides of a PEC plate. There are via holes on the PEC plate through which the PEC strips on the opposite sides of the PEC plate is connected. The PEC plate is of dimension $5\lambda \times 6\lambda$. The height of the vias is 0.5λ. The width of the vias is 0.24λ. On each side the of PEC plate, the length of the via-arms that is parallel to the plate measures 5λ. The distance between two vias is 0.24λ. The via-holes have dimension $0.24\lambda \times 0.48\lambda$. The CPU times required for ΘR factorization at the finest level of the oct-tree and for the full NL-LOGOS factorization are plotted in Figure 21. The efficiencies of the ΘR factorization and the NL-LOGOS factorization for the vias case are similar to those of the sphere case shown in Figure 20.

The three cases studied above demonstrate the low frequency (i.e., h-refinement) behavior of both the ΘR factorization and the NL-LOGOS factorization on a system matrix obtained from the EFIE. In the following example, a series of PEC spheres with radii varying from 2λ to 8λ are studied. The mesh density is fixed to maintain 300 DOF/λ^2 as N is increased. The CPU times for ΘR factorization at the finest level and for full NL-LOGOS factorization are plotted in Figure 22. The ΘR factorization for the moderate frequency case has complexity $O(N^{1.5})$. The higher cost of ΘR factorization for the moderate frequency case as compared to that for the low frequency case is due to the lower compression rate on the system matrix by the MLSSM at moderate frequency.

In contrast, the NL-LOGOS factorization time scales somewhat more slowly than the low-frequency examples considered above.

4.2.7. Summary

A sparse ΘR factorization algorithm is proposed for the MLSSM representation of the system matrix associated with surface integral equation formulations of electromagnetic scattering. The ΘR factorization works closely with the MLSSM representation of the system matrix to obtain the R factors required for NL-LOGOS modes calculation at a given level of the oct-tree. The Θ factors are never explicitly formed or saved. No decompressed of any blocks of the sparse system representation is necessary, and no approximations are introduced by the ΘR factorization.

Three dimensional numerical examples have been used to illustrate that the efficiency of the ΘR factorization algorithm depends on the complexity of the underlying MLSSM representation of the system matrix. At low frequencies the MLSSM representation has an $O(N)$ complexity, and it has been observed that the complexity of the ΘR factorization is also approximately $O(N)$. (The complexity of the ΘR factorization at coarser levels of the oct-tree will depend on the number of localized modes found at finer levels of the tree, and is therefore problem dependent.) For a fixed discretization density of approximately 300 DOF/λ^2, the complexity of both the ΘR factorization and the NL-LOGOS factorization scale as approximately $O(N^{1.5})$.

4.3. Error Control in NL-LOGOS Factorization

The NL-LOGOS factorization is approximate due to its reliance on source functions that approximately localize the scattered field. For this reason, the number of localizing modes associated with a given region depends on the tolerance with which these modes need to be localized. A tighter tolerance produces a more accurate LOGOS factorization at the expense of higher costs in terms of both CPU time and memory. The purpose of this paper is to provide detailed theoretical justification for the error control provided by the NL-LOGOS factorization.

Let $\tilde{\mathbf{Z}}$ denote the factorization of $\mathbf{Z}$ in (4.1) obtained using the NL-LOGOS factorization procedure described in [78], which provides an approximate representation of the inverse, $\tilde{\mathbf{Z}}^{-1} \approx \mathbf{Z}^{-1}$. The approximate solution is given by $\tilde{\mathbf{x}} = \tilde{\mathbf{Z}}^{-1}\mathbf{F}$. The most significant error is the solution error:

$$\xi_{sol} = \frac{\left\|\mathbf{x} - \tilde{\mathbf{x}}\right\|_F}{\left\|\mathbf{x}\right\|_F}, \tag{4.60}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. It relates directly to the error in the inverse of the system matrix

$$\xi_{inv} = \frac{\left\|\mathbf{Z}^{-1} - \tilde{\mathbf{Z}}^{-1}\right\|_F}{\left\|\mathbf{Z}^{-1}\right\|_F} \tag{4.61}$$

by

$$\begin{aligned} \xi_{sol} = \frac{\left\|\mathbf{Z}^{-1}\mathbf{F} - \tilde{\mathbf{Z}}^{-1}\mathbf{F}\right\|_F}{\left\|\mathbf{Z}^{-1}\mathbf{F}\right\|_F} &\le \frac{\left\|\mathbf{Z}^{-1} - \tilde{\mathbf{Z}}^{-1}\right\|_F \left\|\mathbf{F}\right\|_F \left\|\mathbf{Z}\right\|_F}{\left\|\mathbf{Z}\right\|_F \left\|\mathbf{Z}^{-1}\mathbf{F}\right\|_F} \\ &\le \left\|\mathbf{Z}^{-1} - \tilde{\mathbf{Z}}^{-1}\right\|_F \left\|\mathbf{Z}\right\|_F \\ &= \xi_{inv} cond\left(\mathbf{Z}\right) \end{aligned}, \tag{4.62}$$

However, neither (4.60) nor (4.61) can be obtained for practical problems due to the lack of a precise solution and inverse. Therefore, to control the error in solution or the inverse of a system matrix is not practical for most problems.

A more practical alternative is the residual error,

$$\xi_{res} = \frac{\left\| \mathbf{Z}\tilde{\mathbf{x}} - \mathbf{F} \right\|_F}{\left\| \mathbf{F} \right\|_F}. \tag{4.63}$$

If $\tilde{\mathbf{x}} = \tilde{\mathbf{Z}}^{-1}\mathbf{F}$, the residual error has the following bounds

$$\begin{aligned} \xi_{res} &= \frac{\left\| \mathbf{Z}\tilde{\mathbf{x}} - \mathbf{F} \right\|_F}{\left\| \mathbf{F} \right\|_F} \le \left\| \mathbf{Z}\tilde{\mathbf{Z}}^{-1} - \mathbf{I} \right\|_F \\ &= \begin{cases} \left\| \mathbf{Z}\tilde{\mathbf{Z}}^{-1} - \mathbf{Z}\mathbf{Z}^{-1} \right\|_F \\ \left\| \mathbf{Z}\tilde{\mathbf{Z}}^{-1} - \tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^{-1} \right\|_F \end{cases}, \\ &\le \begin{cases} \xi_{inv} cond\left(\mathbf{Z}\right) \\ \xi_{mat} cond\left(\tilde{\mathbf{Z}}\right) \end{cases} \end{aligned} \tag{4.64}$$

where ξ_{mat} is the error in the relative RMS system matrix,

$$\xi_{mat} = \frac{\left\| \mathbf{Z} - \tilde{\mathbf{Z}} \right\|_F}{\left\| \mathbf{Z} \right\|_F}. \tag{4.65}$$

In (4.64), $cond(\bullet)$ indicates the condition number of a matrix, and we have used $\left\| \mathbf{Z} \right\|_F \approx \left\| \tilde{\mathbf{Z}} \right\|_F$. The purpose of this paper is to demonstrate that the NL-LOGOS factorization directly controls the relative RMS system matrix error given by (4.65). Consequently, the factorization also provides indirect control of the maximum residual error indicated by (4.64).

The notations $\mathbf{Z}_{l+1}^{(NN)}$ and $\mathbf{Z}_l$ refer to the same matrix. They are used interchangeably, possibly combined with other subscripts, in the following discussions. Let the error introduced by the multilevel LOGOS factorization be $\mathbf{Z}_{err,L+1}^{(NN)}$. The target error control is

$$\frac{\left\| \mathbf{Z}_{err,L+1}^{(NN)} \right\|_F}{\left\| \mathbf{Z}_{L+1}^{(NN)} \right\|_F} \le \varepsilon, \tag{4.66}$$

where ε is the desired tolerance. It can be seen from (4.14) that during the one level LOGOS factorization, the system matrix is projected onto two orthogonal sub-spaces: one is $\Omega^{(L)}$ spanned by the localized LOGOS source modes; and the other is $\Omega^{(N)}$, which is the orthonormal complement sub-space of $\Omega^{(L)}$. The current level factorization approximates the system matrix projected on $\Omega^{(L)}$. And, the subsequent multilevel

factorization makes approximations to the system matrix projected on $\Omega^{(N)}$. In order to control the relative error of a multilevel LOGOS factorization on a system matrix by ε, it is enough to control the relative errors introduced in the system matrix projected onto the two sub-spaces simultaneously by ε. The following discussion is divided into two parts: the first part discusses how to control the relative error of the system matrix projected onto $\Omega^{(L)}$; the second part discusses how to control the relative error of the system matrix projected onto $\Omega^{(N)}$. The rounding errors due to computer arithmetics are not considered in this work.

4.3.1. Relative Error in Sub-space $\Omega^{(L)}$

Before applying the projection matrix, $\mathbf{P} = \begin{bmatrix} \mathbf{P}^{(L)} & \mathbf{P}^{(N)} \end{bmatrix}$, the level-$L$ LOGOS factorization in (4.17) can be written as

$$\mathbf{Z}_{L+1}^{(NN)} = \mathbf{Z}_L = \begin{bmatrix} \mathbf{Z}_L^{(L)} & \mathbf{Z}_L^{(N)} \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_L^{(L)} & \mathbf{\Lambda}_L^{(N)} \end{bmatrix}^{-1}. \tag{4.67}$$

The error of level-L factorization is given by

$$\mathbf{Z}_{err,L} = \begin{bmatrix} \mathbf{Z}_{err,L}^{(L)} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}^{(L)} & \mathbf{\Lambda}^{(N)} \end{bmatrix}^{-1}. \tag{4.68}$$

It can be shown that

$$\begin{bmatrix} \mathbf{\Lambda}^{(L)} & \mathbf{\Lambda}^{(N)} \end{bmatrix}^{-1} = \begin{bmatrix} \left(\mathbf{\Lambda}^{(L)}\right)^{\dagger} \\ \left(\mathbf{\Lambda}^{(N)}\right)^{H} \end{bmatrix}, \tag{4.69}$$

where both $\left(\mathbf{\Lambda}^{(\mathrm{L})}\right)^{\dagger}$ and $\left(\mathbf{\Lambda}^{(N)}\right)^{H}$ are block diagonal matrices. The jth block in $\left(\mathbf{\Lambda}^{(\mathrm{L})}\right)^{\dagger}$ can be written as $\left(\mathbf{\Lambda}_j^{(\mathrm{L})}\right)^{\dagger} = \mathbf{r}_j^{-1}(\mathbf{q}_j^{(\mathrm{L})})^H$, where $\mathbf{q}_j^{(L)}$ and $\mathbf{r}_j$ are the results of the QR factorization on $\mathbf{\Lambda}_j^{(\mathrm{L})}$. Using (4.69), (4.68) can be written as

$$\mathbf{Z}_{err,L} = \mathbf{Z}_{err,L}^{(L)} \left(\mathbf{\Lambda}^{(L)}\right)^{\dagger}. \tag{4.70}$$

The errors introduced into the system matrix blocks that correspond to different source groups are independent of each other because $\left(\mathbf{\Lambda}^{(\mathrm{L})}\right)^{\dagger}$ is block diagonal.

To control the relative error in (4.70),

$$\left\| \mathbf{Z}_{err,L}^{(L)} \left(\mathbf{\Lambda}^{(L)} \right)^{\dagger} \right\|_F \Big/ \left\| \mathbf{Z} \; \mathbf{\Lambda}^{(L)} \left(\mathbf{\Lambda}^{(L)} \right)^{\dagger} \right\|_F < \varepsilon , \tag{4.71}$$

it is sufficient to control the error associated with the individual matrix block corresponding to the *j*th source group, $\mathbf{Z}_j$, in the following manner:

$$\left\| \mathbf{Z}_{err,L,j}^{(L)} \left(\mathbf{\Lambda}_j^{(L)} \right)^{\dagger} \right\|_F \Big/ \left\| \mathbf{Z}_j \mathbf{\Lambda}_j^{(L)} \left(\mathbf{\Lambda}_j^{(L)} \right)^{\dagger} \right\|_F < \varepsilon . \tag{4.72}$$

where $\|\bullet\|_F$ means the Frobenius norm. The denominator in (4.72) is the Frobenius norm of the matrix $\mathbf{Z}_j$ projected onto $\Omega^{(L)}$. The value N_L should be determined such that (4.72) is satisfied. The error control given by (4.72) can be simplified in the following ways:

1) A Tight Control Formula

$$\begin{aligned}
\frac{\left\| \mathbf{Z}_{err,L,j}^{(L)} \left(\mathbf{\Lambda}_j^{(L)} \right)^{\dagger} \right\|_F}{\left\| \mathbf{Z}_j \mathbf{q}_j^{(L)} \left(\mathbf{q}_j^{(L)} \right)^H \right\|_F} &= \frac{\left\| \mathbf{Z}_{err,L,j}^{(L)} \mathbf{r}_j^{-1} (\mathbf{q}_j^{(L)})^H \right\|_F}{\left\| \mathbf{Z}_j^{(L)} \mathbf{r}_j^{-1} \left(\mathbf{q}_j^{(L)} \right)^H \right\|_F} = \frac{\left\| \mathbf{Z}_{err,L,j}^{(L)} \mathbf{r}_j^{-1} \right\|_F}{\left\| \mathbf{Z}_j^{(L)} \mathbf{r}_j^{-1} \right\|_F} \\
&= \frac{\left\| \mathbf{Z}_{err,L,j}^{(L)} \left(\mathbf{Z}_j^{(L)} \right)^{\dagger} \mathbf{Z}_j^{(L)} \mathbf{r}_j^{-1} \right\|_F}{\left\| \mathbf{Z}_j^{(L)} \mathbf{r}_j^{-1} \right\|_F} \\
&\le \left\| \mathbf{Z}_{err,L,j}^{(L)} \left(\mathbf{Z}_j^{(L)} \right)^{\dagger} \right\|_F \\
&= \left\| \mathbf{Z}_{err,L,j}^{(L)} \right\|_F \\
&= \sqrt{\sum_{i=1}^{N_L} \left(1 - s_i^2 \right)}
\end{aligned} \tag{4.73}$$

From (4.73) and (4.72), the error control criteria for each source group can be derived as

$$\sum_{i=1}^{N_L} \left(1 - s_i^2 \right) \le \varepsilon^2 . \tag{4.74}$$

2) An Alternative Tight Control Formula

$$\frac{\left\|\mathbf{Z}_{err,L,j}^{(L)}\left(\mathbf{\Lambda}_j^{(L)}\right)^{\dagger}\right\|_F}{\left\|\mathbf{Z}_j\mathbf{q}_j^{(L)}\left(\mathbf{q}_j^{(L)}\right)^H\right\|_F}=\frac{\left\|\mathbf{Z}_{err,L,j}^{(L)}\mathbf{r}_j^{-1}(\mathbf{q}_j^{(L)})^H\right\|_F}{\left\|\mathbf{Z}_j^{(L)}\mathbf{r}_j^{-1}\left(\mathbf{q}_j^{(L)}\right)^H\right\|_F}=\frac{\left\|\mathbf{Z}_{err,L,j}^{(L)}\mathbf{r}_j^{-1}\right\|_F}{\left\|\mathbf{Z}_j^{(L)}\mathbf{r}_j^{-1}\right\|_F}$$

$$=\frac{\left\|\mathbf{Z}_{err,L,j}^{(L)}\mathbf{u}_{r^{-1}}\mathbf{s}_{r^{-1}}\mathbf{v}_{r^{-1}}^H\right\|_F}{\left\|\mathbf{Z}_j^{(L)}\mathbf{u}_{r^{-1}}\mathbf{s}_{r^{-1}}\mathbf{v}_{r^{-1}}^H\right\|_F}=\frac{\left\|\mathbf{Z}_{err,L,j}^{(L)}\mathbf{u}_{r^{-1}}\mathbf{s}_{r^{-1}}\right\|_F}{\left\|\mathbf{Z}_j^{(L)}\mathbf{u}_{r^{-1}}\mathbf{s}_{r^{-1}}\right\|_F} \tag{4.75}$$

$$\le\frac{\max\left(\mathbf{s}_{r^{-1}}\right)\left\|\mathbf{Z}_{err,L,j}^{(L)}\mathbf{u}_{r^{-1}}\right\|_F}{\min\left(\mathbf{s}_{r^{-1}}\right)\left\|\mathbf{Z}_j^{(L)}\mathbf{u}_{r^{-1}}\right\|_F}=\frac{\max\left(\mathbf{s}_{r^{-1}}\right)\left\|\mathbf{Z}_{err,L,j}^{(L)}\right\|_F}{\min\left(\mathbf{s}_{r^{-1}}\right)\left\|\mathbf{Z}_j^{(L)}\right\|_F}$$

$$=\mathrm{cond}\left(\mathbf{r}_j^{-1}\right)\sqrt{\sum_{j=1}^{N_L}\left(1-s_j^2\right)\Big/N_L}$$

In this case, the error control criteria for each source group is

$$\sum_{i=1}^{N_L}\left(1-s_i^2\right)\le N_L\varepsilon^2\Big/\left[\mathrm{cond}\left(\mathbf{r}_j\right)\right]^2, \tag{4.76}$$

since $\mathrm{cond}\left(\mathbf{r}_j^{-1}\right)=\mathrm{cond}\left(\mathbf{r}_j\right)$.

3) An Approximate Control Formula

This derivation is a simplified version of (4.76) by making the approximation that

$$\mathrm{cond}\left(\mathbf{r}_j\right)\approx 1. \tag{4.77}$$

(4.76) becomes

$$\sum_{i=1}^{N_L}\left(1-s_i^2\right)\le N_L\varepsilon^2. \tag{4.78}$$

4.3.2. Practical Considerations

The condition (4.74) is too strict for all of our testing cases. Even though it provides good error control, it results in less localized LOGOS modes found for each group at each level. The efficiency of the solver is greatly affected. The condition (4.76) is an alternative control criteria which depends on the condition number of $\mathbf{r}_j$ and therefore $\mathbf{\Lambda}_j^{(L)}$. So, (4.76) could not be used to determine N_L directly. It can only be used as an error check after a trial N_L is provided. By comparing (4.74) and (4.76), it is apparent that a unified criteria should be

$$\sum_{i=1}^{N_L}\left(1-s_i^2\right)\le\varepsilon^2\max\left(1,N_L\Big/\left[\mathrm{cond}\left(\mathbf{r}_j\right)\right]^2\right). \tag{4.79}$$

(4.79) reduces to (4.74), (4.76) and even (4.78) when $\mathrm{cond}\left(\mathbf{r}_j\right)\ge N_L$, 1<$\mathrm{cond}\left(\mathbf{r}_j\right)<N_L$ and $\mathrm{cond}\left(\mathbf{r}_j\right)=1$, respectively. The benefit of using (4.78) is that the computation of $\mathrm{cond}\left(\mathbf{r}_j\right)$ is not required, whose cost increases as the problems size grows. In all test cases studied in this work, (4.78) provides good error control. However, for precise error control, (4.79) should be used to check the error condition.

4.3.3. Relative Error in Sub-space $\Omega^{(N)}$

Continuing the factorization to coarser levels introduces additional approximations into the term $\mathbf{Z}_L^{(N)}$ in (4.67). Let the collective error introduced by these subsequent factorizations be indicated by $\mathbf{Z}_{err,L}^{(N)}$. It should satisfy

$$\frac{\left\|\mathbf{Z}_{err,L}^{(N)}\right\|_F}{\left\|\mathbf{Z}_{L+1}^{(NN)}\boldsymbol{\Lambda}_L^{(N)}\left(\boldsymbol{\Lambda}_L^{(N)}\right)^H\right\|_F}\le\varepsilon\,. \tag{4.80}$$

The projection matrix, $\mathbf{P}_L^H=\left[\mathbf{P}_L^{(L)}\quad\mathbf{P}_L^{(N)}\right]^H$, splits $\mathbf{Z}_L^{(N)}=\mathbf{Z}_{L+1}^{(\mathrm{NN})}\boldsymbol{\Lambda}_L^{(N)}$ into two parts, $\mathbf{Z}_L^{(LN)}$ and $\mathbf{Z}_L^{(NN)}$ as

$$\mathbf{P}_L^H\mathbf{Z}_L^{(N)}=\begin{bmatrix}\mathbf{Z}_L^{(\mathrm{LN})}\\ \mathbf{Z}_L^{(\mathrm{NN})}\end{bmatrix}. \tag{4.81}$$

The multilevel LOGOS factorization after level-L works on $\mathbf{Z}_L^{(NN)}$ only, and $\mathbf{Z}_L^{(LN)}$ is left intact. The error control for finding localizing LOGOS modes at level-$(L-1)$ for $\mathbf{Z}_L^{(NN)}$ follows the same arguments provided in Section 4.3.1 above. An error control similar to (4.66) can be achieved such that

$$\frac{\left\|\mathbf{Z}_{err,L}^{(NN)}\right\|_F}{\left\|\mathbf{Z}_L^{(NN)}\right\|_F}\le\varepsilon_{L-1}. \tag{4.82}$$

The relative tolerance ε_{L-1} is from $\varepsilon_L = \varepsilon$ used in (4.80). Comparing (4.80) and (4.82), and noting that $\mathbf{Z}_{err,L}^{(NN)} = \mathbf{Z}_{err,L}^{(N)}$, we have

$$\varepsilon_{L-1} = \rho \varepsilon_L , \tag{4.83}$$

where

$$\rho = \frac{\sqrt{\left\| \mathbf{Z}_L^{(NN)} \right\|_F^2 + \left\| \mathbf{Z}_L^{(LN)} \right\|_F^2}}{\left\| \mathbf{Z}_L^{(NN)} \right\|_F} . \tag{4.84}$$

The ratio ρ provides a scaling on the tolerance for the multilevel LOGOS factorization between level-$(L-1)$ and level-L. The numerator in (4.84) can be obtained as

$$\begin{aligned} \sqrt{\left\| \mathbf{Z}_L^{(NN)} \right\|_F^2 + \left\| \mathbf{Z}_L^{(LN)} \right\|_F^2} &= \left\| \mathbf{Z}_L^{(N)} \right\|_F \\ &= \sqrt{\left\| \mathbf{Z}_L \right\|_F^2 - \left\| \mathbf{Z}_L \mathbf{\Lambda}_L^{(L)} \left(\mathbf{\Lambda}_L^{(L)} \right)^{\dagger} \right\|_F^2} , \\ &= \sqrt{\left\| \mathbf{Z}_L \right\|_F^2 - \left\| \mathbf{r}_L^{-1} \right\|_F^2} \end{aligned} \tag{4.85}$$

where $\mathbf{r}_L$ is a block diagonal matrix containing the R-factors obtained from the QR factorizations associated with each block of $\mathbf{\Lambda}_L^{(L)}$. The tolerance relation between any two subsequent levels can be set up in the same way as that given by (4.83)-(4.85). In this way, the multilevel LOGOS factorization at any level can always achieve the desired error required by its finer level factorization.

4.3.4. Summary on Error Control Formulas

The error control formulas for determining localizing source modes are given by (4.78) or (4.79). The tolerances used in (4.78) or (4.79) for multi-level factorization should be scaled according to (4.83).

The error control formula (4.78) has been used in previous works [78, 81]. An approximate version of (4.78),

$$s_{N_L} > 1 - 0.5\varepsilon^2 , \tag{4.86}$$

was also used before [78]. It is obtained by approximating (4.78) with $1 - s_{N_L}^2 < \varepsilon^2$ and rewriting the relation by keeping the Taylor expansion terms up to the first order. The

relation, $1-s_{N_L}^2 < \varepsilon^2$, is a more conservative condition than (4.78) because s_i are ordered in non-increasing order.

The tolerance scaling factor given by (4.84) is always larger than or equal to one. Therefore, the multi-level LOGOS factorization tolerance for coarser levels are relaxed. This factor is important to improve the efficiency of the factorization while maintaining the error control.

4.3.5. Numerical Results

To evaluate the error control criteria given by (4.75) and (4.76) we consider several examples of plane wave scattering from PEC targets formulated using the electric field integral equation (EFIE). A locally corrected Nyström method with mixed-order basis functions [35] is used to obtain the matrix equations. The surfaces of the PEC objects are meshed with quadrilateral cells. In applying the Nyström method, two sets of quadrature points on each cell are used to sample the two components of the surface current. Each set contains six quadrature points, so there are 12 DOFs (quadrature points) in every quadrilateral cell. All meshes are generated such that there are at least 300 degrees of freedom (DOF) per square wavelength. In the following discussion, condition (4.75) is simply referred to as the tight condition and (4.76) as the approximate condition. The tolerance scaling for the multilevel NL-LOGOS factorization is made using (4.83)-(4.85).

The first example considered is plane wave scattering from long thin PEC strips. Due to their planer structure with large spatial extent, the strips have many localized LOGOS modes at the finer levels of the trees. Because the approximation in the LOGOS factorization is associated with the localizing modes, a larger number of approximations are made at fine levels of the tree for this geometry, and the error control must be appropriate in order to maintain the final error in factored system matrix.

The strips considered are infinitely thin with length 200λ. The width of the strips are 0.2λ, 0.1λ, 0.05λ, 0.025λ and 0.0125λ. The strips are meshed using square cells with the sides of the cells the same as the width of the strips. Therefore, the number of cells on the strips are 1000, 2000, 4000, 8000 and 16000, respectively.

The RMS error in the factored matrix and the residual error (with plane wave excitation) are plotted in Figure 23. The three different tolerances considered, 10^{-3}, 10^{-4}

and 10^{-5}, provide consistent improvement in the factorization errors. Due to low frequency breakdown of the formulation, the residual error is not as well controlled. This is particularly evident for the case with 16000 cells and tolerance of 10^{-3}. To reduce the residual error, the formulation has to be changed to improve the conditioning of the system matrix. Another possibility would be to precondition the formulation prior to factorization.

Figure 24 plots the factorization errors obtained using the tight and approximate error control formulas. With the same set of tolerances, the tight condition, (4.75), produces consistently smaller factorization error than the approximate condition, (4.76). Even though the tight condition produces smaller error, it is not a preferable condition because it needs to evaluate the condition number of $\mathbf{r}_j$ in (4.76), which is time consuming. Furthermore, the approximate condition provides similar error control with less cost. Therefore, the approximate condition is used in the following cases and other applications.

Another test case contains a sphere with radius 2λ. Plane wave excitation is used. The surface of the sphere is meshed with different densities. The factorization errors and residual errors are plotted in Figure 25. Only approximate error control is used in obtaining the results. Consistent error control is obtained as the factorization tolerance is changed.

A third case is a vias-like structure (see Figure 11) which contains multiple PEC strips floating on both sides of a PEC plate. There are via hols on the PEC plate through which the PEC strips on the opposite sides of the PEC plate are connected. The PEC plate is of dimension $5\lambda \times 6\lambda$.The height of the vias is 0.5λ. The width of the vias is 0.24λ. On each side the of PEC plate, the length of the via-arms that is parallel to the plate measures 5λ. The distance between two vias is 0.24λ. The via-holes have dimensions $0.24\lambda \times 0.48\lambda$. The factorization errors and residual errors obtained using the approximate control criteria are plotted in Figure 26. Again, consistent error control can be obtained as the tolerance changes.

Without showing the results, the cases studied above, i.e., strip, sphere and vias, have LOGOS factorization complexity of $O(N)$, $O(N^{2.1})$ and $O(N^2)$, respectively. The

efficiency of the NL-LOGOS factorization can be improved by employing different formulations, such as MFIE or CFIE [78]. It was shown that the OL-LOGOS factorization improves the efficiency of the algorithm for all formulations [77].

4.4. NL-LOGOS Based Fast Direct Solution

The NL-LOGOS factorization is derived in Section 4.1. Equation (4.17) can be written in a more general format as

$$\mathbf{Z}_{l+1}^{(NN)} \approx \begin{bmatrix} \mathbf{P}_l^{(L)} & \mathbf{P}_l^{(N)} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{I}} & \mathbf{Z}_l^{(LN)} \\ \mathbf{0} & \mathbf{Z}_l^{(NN)} \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_l^{(L)} & \mathbf{\Lambda}_l^{(N)} \end{bmatrix}^{-1}. \tag{4.87}$$

The inverse of (4.87) is

$$\left(\mathbf{Z}_{l+1}^{(NN)}\right)^{-1} \approx \begin{bmatrix} \mathbf{\Lambda}_l^{(L)} & \mathbf{\Lambda}_l^{(N)} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{I}} & -\mathbf{Z}_l^{(LN)}\left(\mathbf{Z}_l^{(NN)}\right)^{-1} \\ \mathbf{0} & \left(\mathbf{Z}_l^{(NN)}\right)^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{P}_l^{(L)} & \mathbf{P}_l^{(N)} \end{bmatrix}^{H}. \tag{4.88}$$

Therefore, the inverse of the system matrix is represented recursively using (4.88). For any number of RHS's, the solution can be obtained by multiplying the RHS's with the inverse of the system matrix. The efficiency of the fast direct solvers has been presented in [78] and the references therein.

4.5. Multi-range Localization NL-LOGOS Factorization

Sparse direct solution strategies for time-harmonic electromagnetic problems based on the use of local-global solutions (LOGOS) have been developed and improved over the past few years [25, 74-77, 79, 82]. The efficiency of the LOGOS factorization, which is the key algorithm in developing both LOGOS-based fast direct solvers and MFD methods, is primarily determined by the costs associated with computing the underlying LOGOS modes for a given system matrix. In the following, we focus on computing these functions for the case of a factorization based on non-overlapped, localizing LOGOS (NL-LOGOS) modes [78].

At each level of the factorization, NL-LOGOS modes span a single group at that level and radiate fields localized to the same group. As described in [78], when looking for NL-LOGOS modes at a given level, a sparse ΘR factorization must be carried out for

the projected system matrix at that level (the system matrix is not projected at the finest level). In practice, it occurs that the cost of the NL-LOGOS factorization is dominated by the cost to perform the ΘR factorization and subsequently compute the $\mathbf{\Lambda}_l^{(L)}$ at coarser levels of the tree. Therefore, one way to improve the efficiency of the factorization is to increase the number of localized modes that are obtained at finer levels. One way to accomplish this is to use the overlapped, localizing LOGOS (OL-LOGOS) factorization described elsewhere [77, 82]. While effective at lower frequencies, we have observed that the OL-LOGOS factorization provides little or no improvement in overall CPU efficiency for discretization densities on the order of 300 DOF/λ^2.

Here we consider an alternative strategy for increasing the number of modes determined at finer levels of the tree. This is accomplished by adding an additional step immediately following the NL-LOGOS mode calculation for each group at a given level. In particular, using a "look ahead" strategy, additional localizing LOGOS modes are found for each source group at a given level that produce scattered fields which are localized fields inside the *parent* of the current source group. These modes comprise a subset of the modes that would otherwise be determined by a calculation determined at the parent level. Because this extra step works with the data provided by the ΘR factorization performed for the current level, the cost associated with computing these extra modes is relatively small.

The new procedure can be viewed as a simple extension to the original procedure because it finds extra localizing functions by changing the range of the field localization from the self-group to the parent of the source group. In this sense, the new procedure is referred to as a "multi-range" localization strategy. The multi-range NL-LOGOS factorization effectively reduces the number of non-localized modes at each level, which in turn reduces the cost to perform the ΘR factorization at the coarser level.

4.5.1. Multi-range Localization

Consider the problem of scattering from a PEC strip that is fitted into a 4-level oct-tree. The non-empty groups at each level are indexed as indicated in Figure 31. The horizontal braces in Figure 4 denote the parent-child relation between groups in two consecutive levels of the tree. For this simple structure, any two groups at a level indexed

by contiguous integers are near-neighbor groups because they touch each other. Any two groups that have the same parent are called siblings. Siblings are near-neighbors but near-neighbors are not all siblings. In the following discussions, the near-neighbor groups and the sibling groups include the self-group by default.

Assume the system matrix has been represented by the multilevel simply sparse method (MLSSM) [62-64]. A sparse ΘR factorization [83] of the system matrix at level-4 generates the block diagonal R factors and the unsaved Q factors, such that

$$\begin{aligned}\mathbf{Z} &= \left[\mathbf{Z}_{1(4)},\ldots,\mathbf{Z}_{i(4)},\ldots,\mathbf{Z}_{M(4)}\right] \\ &= \left[\mathbf{q}_{1(4)},\ldots,\mathbf{q}_{i(4)},\ldots,\mathbf{q}_{8(4)}\right]\mathrm{diag}\left(\mathbf{r}_{i(4)}\right) = \boldsymbol{\Theta}_4\mathbf{R}_4\end{aligned}. \tag{4.89}$$

where the subscript $i(4)$ denotes the i th group at level-4. The notation $\mathrm{diag}\left(\mathbf{r}_{i(4)}\right)$ is used to indicate the block diagonal matrix having the square matrices $\mathbf{r}_{i(4)}$ along its diagonal. The notation $\left[\mathbf{Z}_{1(4)},\ldots,\mathbf{Z}_{i(4)},\ldots,\mathbf{Z}_{M(4)}\right]$ represents a source partition of the system matrix $\mathbf{Z}$ according to level-4 groups. The sparse ΘR factorization never forms the matrix $\boldsymbol{\Theta}_4$ explicitly. However, once $\mathbf{R}_4$ is obtained, any blocks in $\boldsymbol{\Theta}_4$ can be readily obtained by

$$\mathbf{q}_{i(4)} = \mathbf{Z}_{i(4)}\mathbf{r}_{i(4)}^{-1}. \tag{4.90}$$

In order to find the field generated by the sources in group $i(4)$ and localized to group $i(4)$, we only need to examine the rows of $\mathbf{q}_{i(4)}$ that correspond to the observation points in group $i(4)$. Let this portion of the matrix be denoted by $\mathbf{q}_{i(4)}^{self}$ with the rest of the matrix denoted by $\mathbf{q}_{i(4)}^{others}$. Similarly, $\mathbf{Z}_{i(4)}^{self}$ represents the portion of the matrix $\mathbf{Z}_{i(4)}$ having the same source and observation points. Matrix $\mathbf{q}_{i(4)}$ can be reordered so that

$$\mathbf{q}_{i(4)} \Rightarrow \begin{bmatrix}\mathbf{q}_{i(4)}^{self} \\ \mathbf{q}_{i(4)}^{others}\end{bmatrix}, \tag{4.91}$$

where $\mathbf{q}_{i(4)}^{self} = \mathbf{Z}_{i(4)}^{self}\mathbf{r}_{i(4)}^{-1}$. Following the procedure described in 4.1, the NL-LOGOS modes can be found for all the groups as

$$\boldsymbol{\Lambda}_4 = \begin{bmatrix} \boldsymbol{\Lambda}_4^{(L)} & \boldsymbol{\Lambda}_4^{(N)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Lambda}_{1(4)}^{(L)} & \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\Lambda}_{1(4)}^{(N)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_{2(4)}^{(L)} & \cdots & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}_{2(4)}^{(N)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \boldsymbol{\Lambda}_{8(4)}^{(L)} & \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\Lambda}_{8(4)}^{(N)} \end{bmatrix}, \quad (4.92)$$

and

$$\mathbf{P}_4 = \begin{bmatrix} \mathbf{P}_4^{(L)} & \mathbf{P}_4^{(N)} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{1(4)}^{(L)} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{P}_{1(4)}^{(N)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{2(4)}^{(L)} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{P}_{2(4)}^{(N)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{P}_{8(4)}^{(L)} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{P}_{8(4)}^{(N)} \end{bmatrix}. \quad (4.93)$$

The factorization of the system matrix can be written as

$$\mathbf{Z}_4 \approx \mathbf{P}_4 \tilde{\mathbf{Z}}_4 \boldsymbol{\Lambda}_4^{-1} = \mathbf{P}_4 \begin{bmatrix} \mathbf{I} & \mathbf{Z}_4^{(LN)} \\ \mathbf{0} & \mathbf{Z}_4^{(NN)} \end{bmatrix} \boldsymbol{\Lambda}_4^{-1}, \quad (4.94)$$

where

$$\tilde{\mathbf{Z}} = \mathbf{P}_4^H \mathbf{Z} \boldsymbol{\Lambda}_4 \approx \begin{bmatrix} \mathbf{I} & \mathbf{P}_4^{(L)H} \mathbf{Z} \boldsymbol{\Lambda}_4^{(N)} \\ \mathbf{0} & \mathbf{P}_4^{(N)H} \mathbf{Z} \boldsymbol{\Lambda}_4^{(N)} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{Z}_4^{(LN)} \\ \mathbf{0} & \mathbf{Z}_4^{(NN)} \end{bmatrix}. \quad (4.95)$$

Because the localized fields and the NL-LOGOS source modes occupy the same groups, this type of localization is referred to as self-group localization. The result of self-group localization for one group at level-4 is illustrated in Figure 27, where the shaded blocks represent all matrix elements that are generally nonzero to a certain tolerance. It should be noted that $\mathbf{q}_{i(4)}^{others}$ is never needed to determine the LOGOS modes. The procedure described in 4.1 can be generalized by:

i. changing the domain of the source modes,
ii. changing the range into which the field should be localized,
iii. a combination of (i) and (ii).

The effect of change the domain over which the source modes are defined has previously been considered elsewhere [77, 82]. In particular, it has been observed that in some cases it is advantages to allow source modes to have nonzero support both within a given group and its near neighbors while still restricting the support of the scattered field to the self-

group. The second possibility is the strategy that is discussed in detail in the next section. The possibility of combining these generalizations will not be considered here.

Equations (4.92) contains NL-LOGOS source modes which localize the scattered fields to the self-group, as illustrated in Figure 27. The cost of the NL-LOGOS factorization is primarily determined by the number of non-local modes passed from child to parent levels. Thus, the cost of the factorization at each parent level can be reduced by finding additional localizing modes at each child level. To this end, we observe that additional localizing modes can be determined for a given source group by enlarging the spatial region to which the scattered fields are localized. Furthermore, because the coarser level NL-LOGOS factorization localizes fields into the parent groups, it is most convenient to enlarge the field localization region to include all sibling groups. In this way, one can effectively determine a subset of the localizing modes associated with each parent group from information available at the child level. This in turn reduces the CPU cost to determine localizing modes at the parent level. Because the extra time spent at the child level to compute the indicated subset of parent level localizing modes is less than the time saved in computing modes at the parent level, this results in an overall time savings for the factorization.

After the level-4 modes are determined as indicated above, we project the system matrix onto $\mathbf{\Lambda}_4^{(N)}$,

$$\mathbf{Z}_4^{(N)} = \mathbf{Z}_4 \mathbf{\Lambda}_4^{(N)}. \tag{4.96}$$

The procedure described by (4.89) to (4.92) can now be used with $\mathbf{Z}_4^{(N)}$ to find the new source modes located in each level-4 group which localize the scattered field to the sibling groups. The only significant difference is the replacement of $\mathbf{q}_{i(4)}^{self}$ with $\mathbf{q}_{i(4)}^{sibling}$ in (4.91) (The sibling groups should now be excluded from $\mathbf{q}_{i(4)}^{others}$). Figure 28 illustrates the relationship between the elements of the sibling-based source modes for a given level-4 group.

Applying this procedure to $\mathbf{Z}_4^{(N)}$ for all level-4 groups results in another set of LOGOS source modes represented by

$$\boldsymbol{\Lambda}_4^s = \begin{bmatrix} \boldsymbol{\Lambda}_4^{(Ls)} & \boldsymbol{\Lambda}_4^{(Ns)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Lambda}_{1(4)}^{(Ls)} & \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\Lambda}_{1(4)}^{(Ns)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_{2(4)}^{(Ls)} & \cdots & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}_{2(4)}^{(Ns)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \boldsymbol{\Lambda}_{8(4)}^{(Ls)} & \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\Lambda}_{8(4)}^{(Ns)} \end{bmatrix}, \quad (4.97)$$

The NL-LOGOS modes found in this way must be associated with groups at the *parent* level because the fields generated by $\boldsymbol{\Lambda}_4^{(Ls)}$ are localized into the same range as the *parent* level NL-LOGOS modes.

The projected matrix

$$\mathbf{Z}_4^{(Ns)} = \mathbf{Z}_4^{(N)} \boldsymbol{\Lambda}_4^{(Ns)}. \quad (4.98)$$

contains the field that cannot be localized at the current level to either the self-groups or the sibling groups. The next step in the multilevel factorization consists in computing the level-3 self-group NL-LOGOS modes. At this point, (4.98) is projected to the LOGOS field modes at level-4:

$$\mathbf{P}_4^H \mathbf{Z}_4^{(Ns)} = \mathbf{P}_4^H \mathbf{Z}_4^{(N)} \boldsymbol{\Lambda}_4^{(Ns)} = \begin{bmatrix} \mathbf{Z}_4^{(LN)} \boldsymbol{\Lambda}_4^{(Ns)} \\ \mathbf{Z}_4^{(NN)} \boldsymbol{\Lambda}_4^{(Ns)} \end{bmatrix}. \quad (4.99)$$

The level-3 NL-LOGOS modes are found by working with

$$\mathbf{Z}_4^{(NNs)} = \mathbf{Z}_4^{(NN)} \boldsymbol{\Lambda}_4^{(Ns)} \quad (4.100)$$

at level-3. Note that with the definition $\mathbf{Z}_4^{(NN)} = \mathbf{Z}_3$, (4.100) can also be written as

$$\mathbf{Z}_4^{(NNs)} = \mathbf{Z}_3 \boldsymbol{\Lambda}_4^{(Ns)} = \mathbf{Z}_3^{(Ns)}. \quad (4.101)$$

After performing a ΘR factorization of $\mathbf{Z}_3^{(Ns)}$ at level-3 the remaining level-3 NL-LOGOS modes can be found as (the superscript, '*r*', means 'remaining')

$$\boldsymbol{\Lambda}_3^{(r)} = \begin{bmatrix} \boldsymbol{\Lambda}_3^{(Lr)} & \boldsymbol{\Lambda}_3^{(Nr)} \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{\Lambda}_{1(3)}^{(Lr)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}_{1(3)}^{(Nr)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_{2(3)}^{(Lr)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}_{2(3)}^{(Nr)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}_{3(3)}^{(Lr)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}_{3(3)}^{(Nr)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}_{4(3)}^{(Lr)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}_{4(3)}^{(Nr)} \end{bmatrix}. \quad (4.102)$$

The preliminary level-3 NL-LOGOS modes, which include the level-4 sibling-localized modes, $\mathbf{\Lambda}_4^{Ls}$, and the remaining level-3 NL-LOGOS source modes, $\mathbf{\Lambda}_3^{(Lr)}$, can be written as

$$\tilde{\mathbf{\Lambda}}_3^{(L)} = \left[\mathbf{\Lambda}_4^{(Ls)} \quad \mathbf{\Lambda}_4^{(Ns)} \mathbf{\Lambda}_3^{(Lr)} \right]. \tag{4.103}$$

In order to obtain the desired format of the projected matrix as given by (4.93) to facilitate the next level LOGOS factorization, the fields generated by the level-3 NL-LOGOS modes should be orthogonal. One more step of orthogonalization makes the fields orthonormal and the desired level-3 NL-LOGOS modes are found at the same time. The orthogonalization involves a QR factorization on the matrix blocks

$$\mathbf{Z}_{i(3)}^{self} \tilde{\mathbf{\Lambda}}_{i(3)}^{(L)} = \mathbf{QR}, \tag{4.104}$$

where $\mathbf{Z}_{i(3)}^{self}$ is the self-interaction matrix block of group $i(3)$. Multiply both sides of (4.104) with $\mathbf{R}^{-1}$. The Level-3 NL-LOGOS modes is found to be $\mathbf{\Lambda}_{i(3)}^{(L)} = \tilde{\mathbf{\Lambda}}_{i(3)}^{(L)} \mathbf{R}^{-1}$. We also find $\mathbf{P}_{i(3)}^{(L)} = \mathbf{Q}$. The non-localized part of the modes can be found by taking the orthonormal complement of $\mathbf{\Lambda}_{i(3)}^{(L)}$ and $\mathbf{P}_{i(3)}^{(L)}$, respectively.

This completes the determination of all level-3 NL-LOGOS modes with the new procedure which includes the step of finding the level-4 sibling-localized LOGOS modes at the child level. Obviously, the same procedure can be used for all levels except level-2 and level-1 because the level-2 sibling-localized modes and level-1 localized modes are trivial.

4.5.2. Numerical Results

To evaluate the performance of the multi-range NL-LOGOS factorization we consider plane wave scattering from a sequence of PEC spheres having radii of 2λ, 2.83λ, 4λ and 5.66λ. The electric field integral equation (EFIE) formulation is used, and a locally corrected Nyström method with mixed order basis functions is used to discretize the EFIE [35, 43]. The spheres are meshed using a uniform distribution of bilinear quads, with each quad cell having 12 quadrature points for the Nyström discretization. Each quadrature point represents one degree of freedom (DOF) of the electric current. The density of the mesh provides an average of 300 DOFs/λ^2. The system is filled by using

the adaptive cross approximation (ACA) [56, 57] to determine the compressed MLSSM representation of the EFIE matrix [62-64]. The MLSSM representation of the system matrix is then factored using both the original NL-LOGOS factorization and the multi-range NL-LOGOS factorization presented in this paper. In both cases, factorization tolerances of 10^{-3} and 10^{-4} are used. The corresponding ACA and MLSSM tolerances are ten times smaller than the factorization tolerance. In all cases considered, the root-mean-square (RMS) error between the solutions obtained using the two methods is less than 10^{-2}. The radar cross section (RCS) of a PEC sphere with radius 2.83λ is shown in Figure 29. Good agreements among the original LOGOS factorization, multi-range localization factorization and Mie series solution are obtained. The factorization tolerance used in obtaining the RCS results in Figure 29 is 10^{-3}. The CPU times required to perform the factorization using both the original and the proposed methods are plotted in Figure 30. While both methods have similar asymptotic CPU complexity, the CPU time required by the multi-range NL-LOGOS factorization is about 40% less than the original LOGOS factorization. Similar CPU savings have been observed for other PEC targets. The memory used by the multi-range NL-LOGOS factorization algorithm is essentially unchanged from that used by the original NL-LOGOS factorization and reported in [78].

In conclusion, The multi-range NL-LOGOS factorization reduces the cost of finding NL-LOGOS modes significantly. Numerical results indicate a net 40% reduction in the factorization time relative to the previously reported NL-LOGOS factorization algorithm with no change in the asymptotic complexity of the factorization time.

The multi-range localization strategy considers the self and parent group localization. The range of the field localization can be further expanded to the grand-parent group localization and so on until level-one. However, the cost of finding multi-scale localizing modes increases as the localizing-field-range increases and exceeds the time saved for the coarser level factorizations as some point. To determine the optimal maximum field-range that provides maximum time savings to the factorization is not studied here.

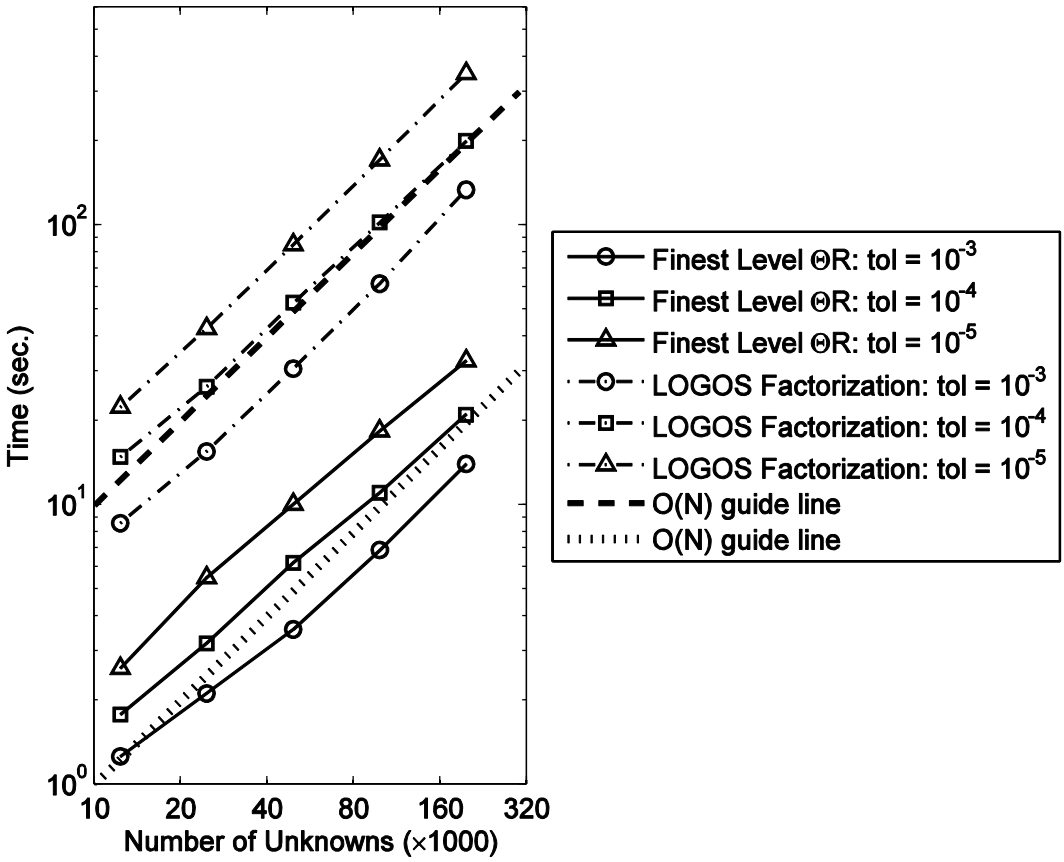


Figure 19. CPU time (in seconds) to perform ΘR factorization of MLSSM data structure at the finest level, and to perform the full multilevel NL-LOGOS factorization of the MLSSM data structure. The geometry is a long thin PEC strip of length 200λ. The width of the strip is the same as the side length of the square cells used to mesh the strip.

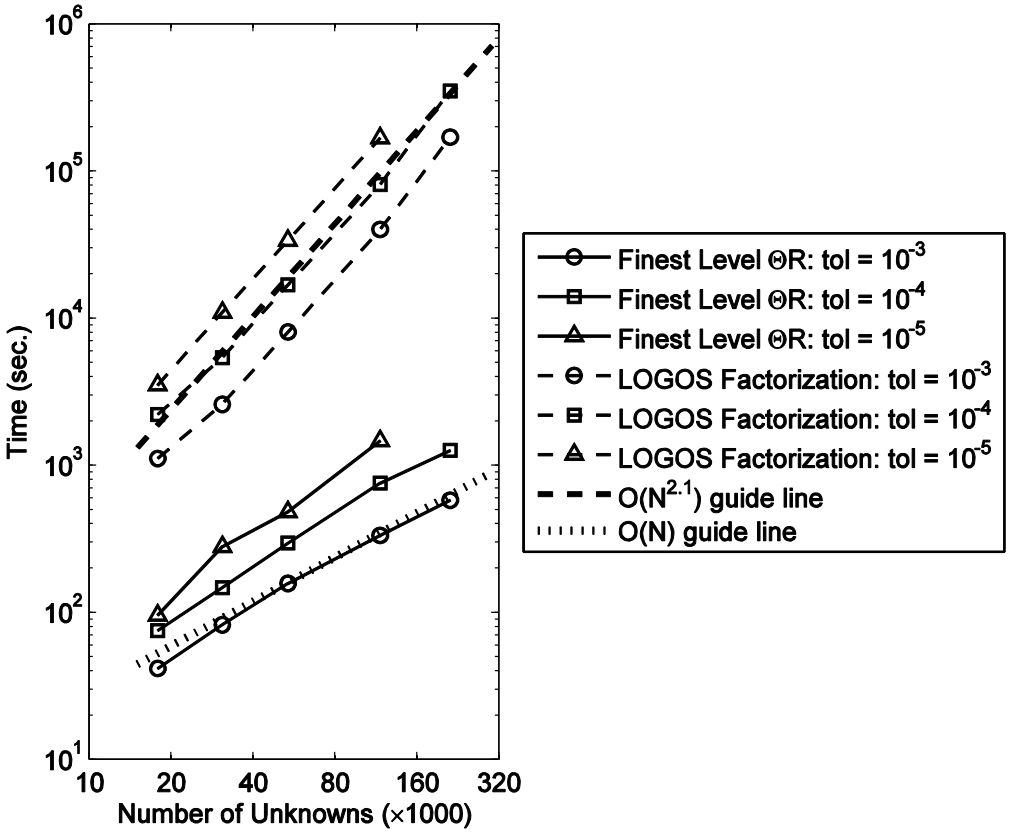


Figure 20. CPU Time (in seconds) to perform ΘR factorization of MLSSM data structure at the finest level, and to perform the full multilevel NL-LOGOS factorization of the MLSSM data structure. The geometry is a PEC sphere with radius 2λ.

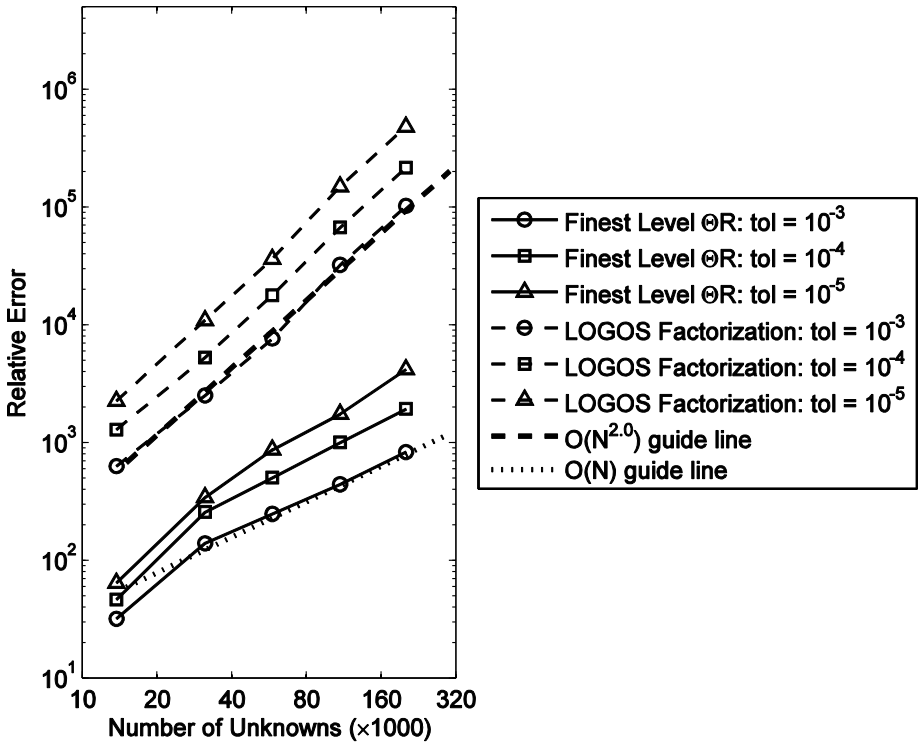


Figure 21. CPU Time (in seconds) to perform ΘR factorization of MLSSM data structure at the finest level, and to perform the full multilevel NL-LOGOS factorization of the MLSSM data structure. The geometry is shown in Figure 11.

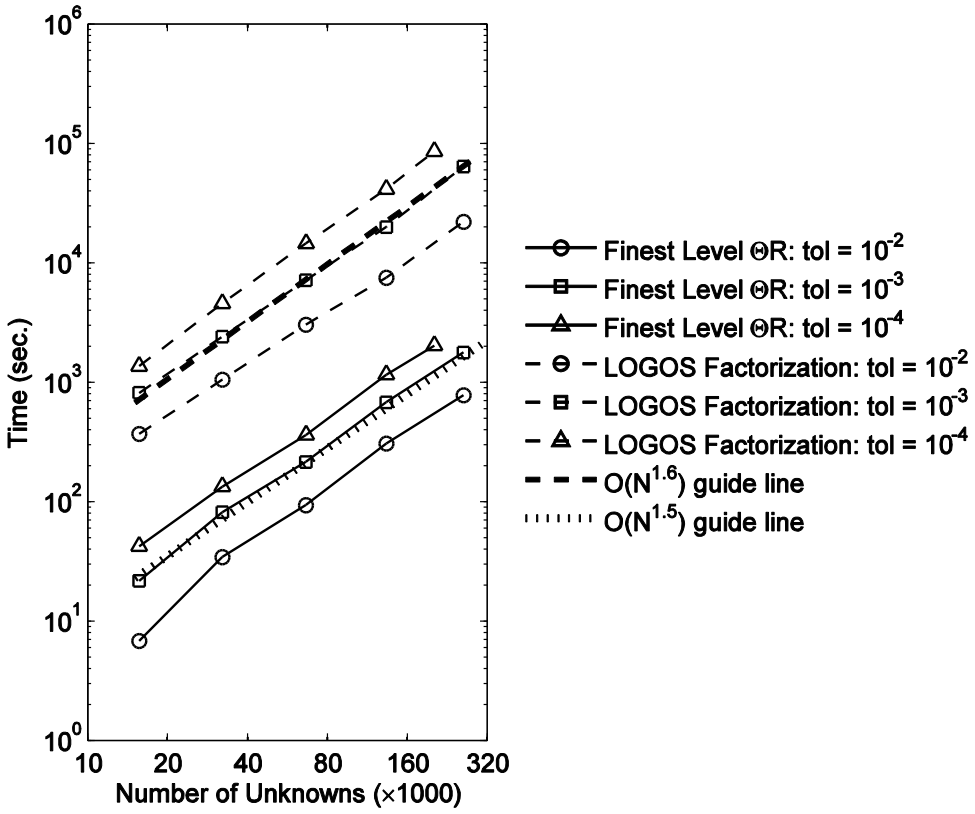


Figure 22. CPU Time (in seconds) to perform ΘR factorization of MLSSM data structure at the finest level and to perform the full multilevel NL-LOGOS factorization of the MLSSM data structure. The geometries are a series of PEC spheres with radii increasing from 2λ to 8λ. In all cases, the discretization density is approximately 300 DOF/λ^2.

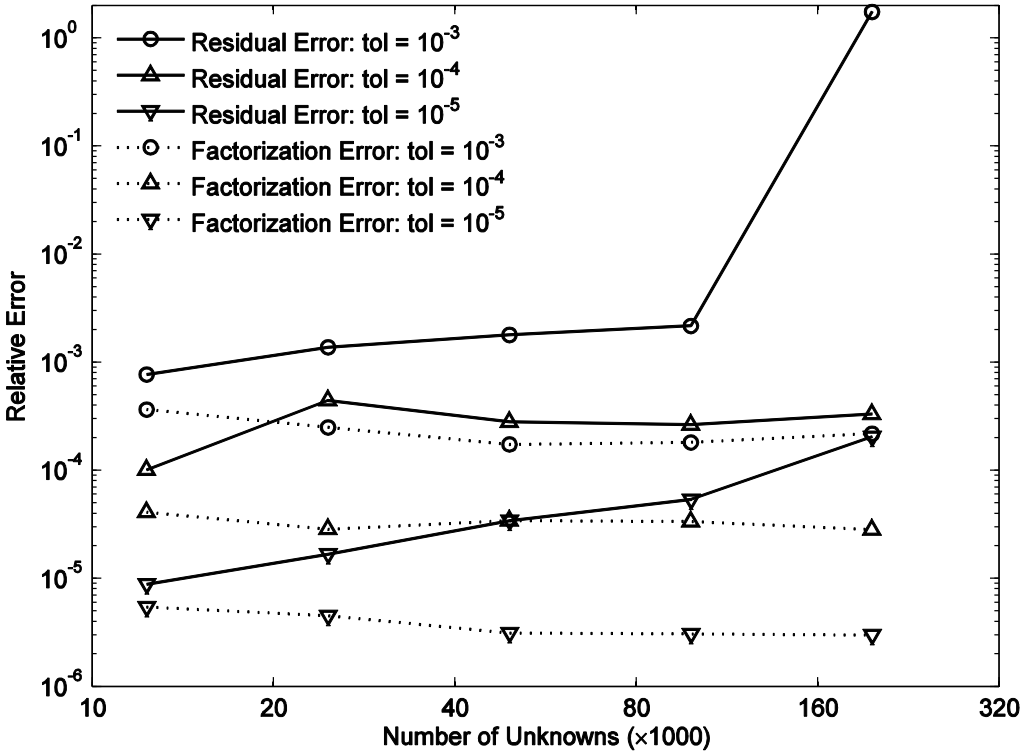


Figure 23. The residual error and RMS factorization error of scattering problems with long, infinitely thin PEC strips of length 200λ. The width of the strips are 0.2λ, 0.1λ, 0.05λ, 0.025λ and 0.0125λ. The strips are meshed using square cells with the sides of the cells the same as the width of the strips.

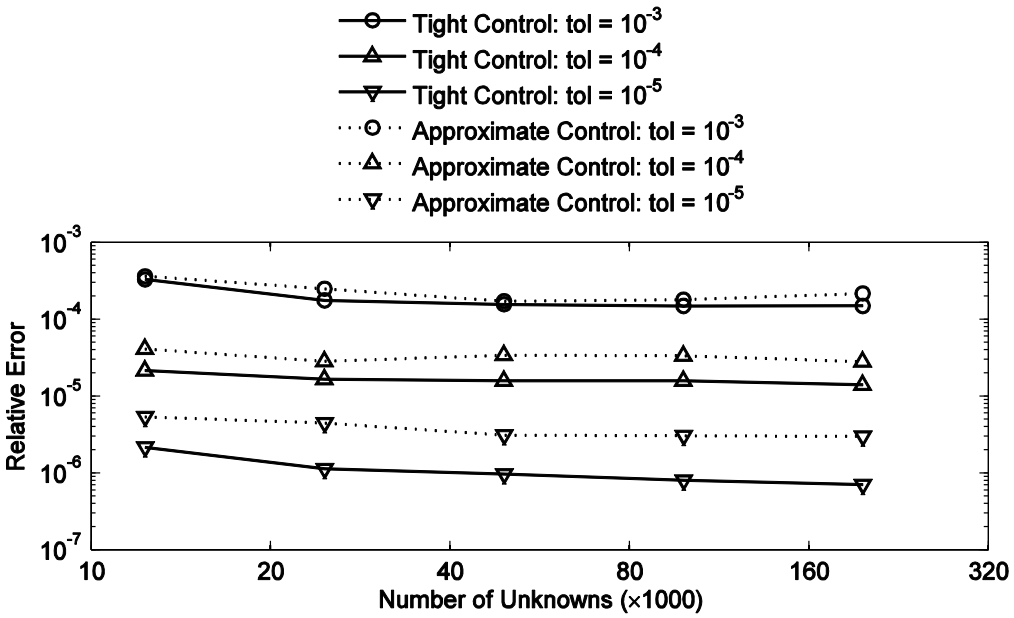


Figure 24. The RMS factorization error of a scattering problem with the same strips as those used for Figure 23. Both tight error control formulation and approximate error control formulation are used.

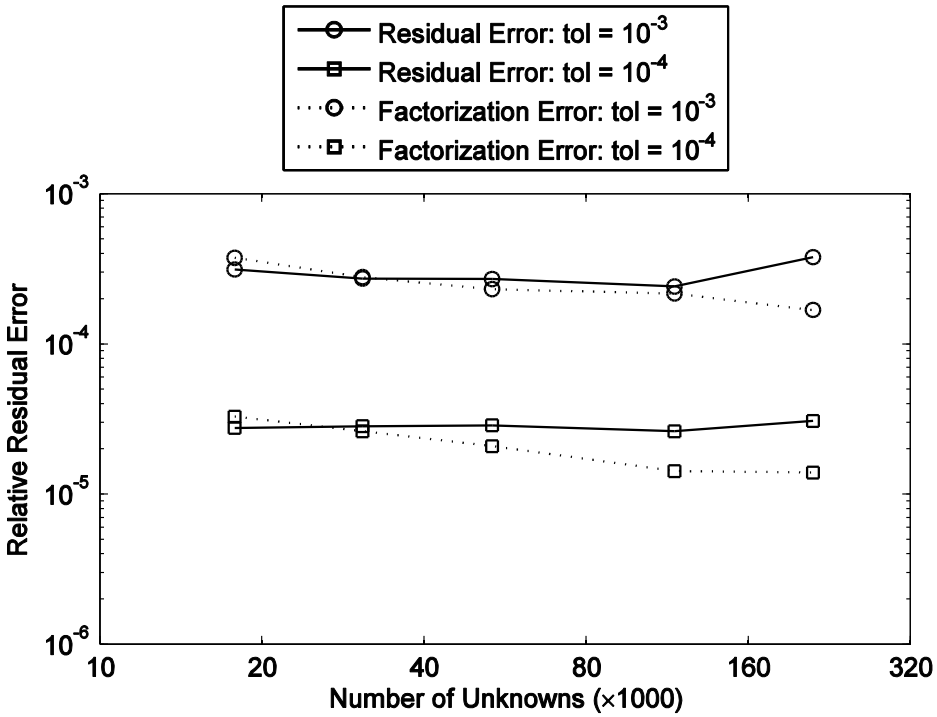


Figure 25. The residual error and the RMS factorization error of a scattering problem with a PEC sphere of radius 2λ.

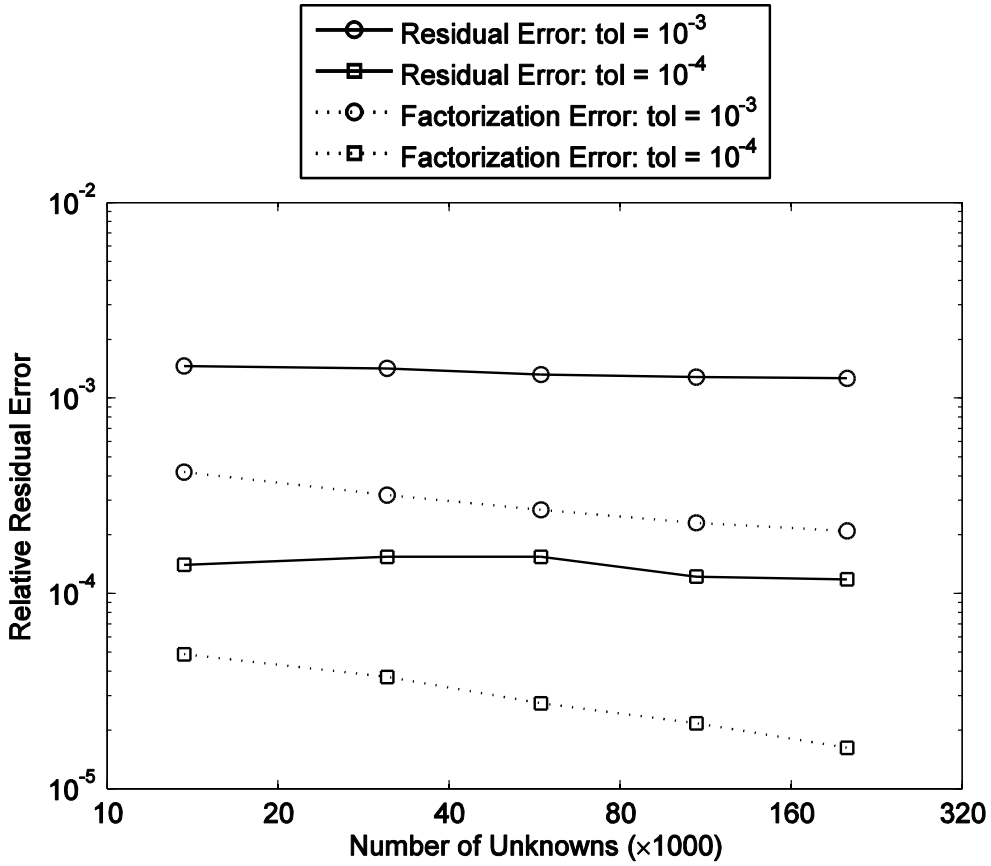


Figure 26. The residual error of a scattering problem by a vias-like structure.

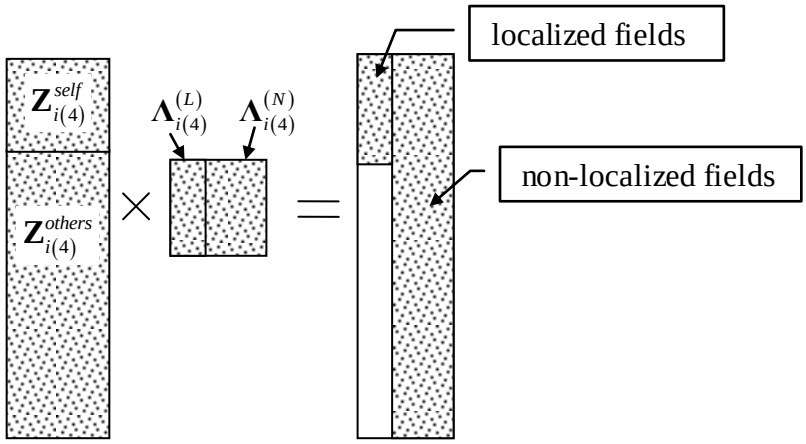


Figure 27. Level-4 column block of the impedance matrix, $\mathbf{Z}_{i(4)}$, LOGOS source modes, $\mathbf{\Lambda}_{i(4)}$, and the structure of the fields generated by the localizing and non-localizing source modes.

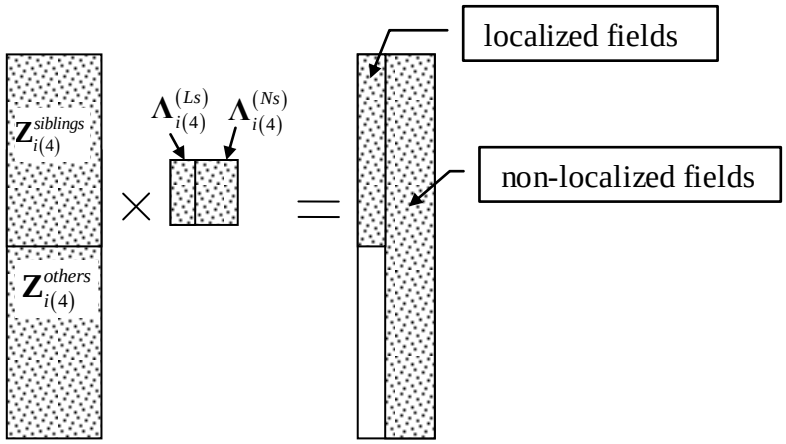


Figure 28. The structure of a block of a projected impedance matrix, $\mathbf{Z}_{i(4)}^{(N)}$, the sibling LOGOS source modes, and the fields generated by the localizing and non-localizing portions of the sibling source modes.

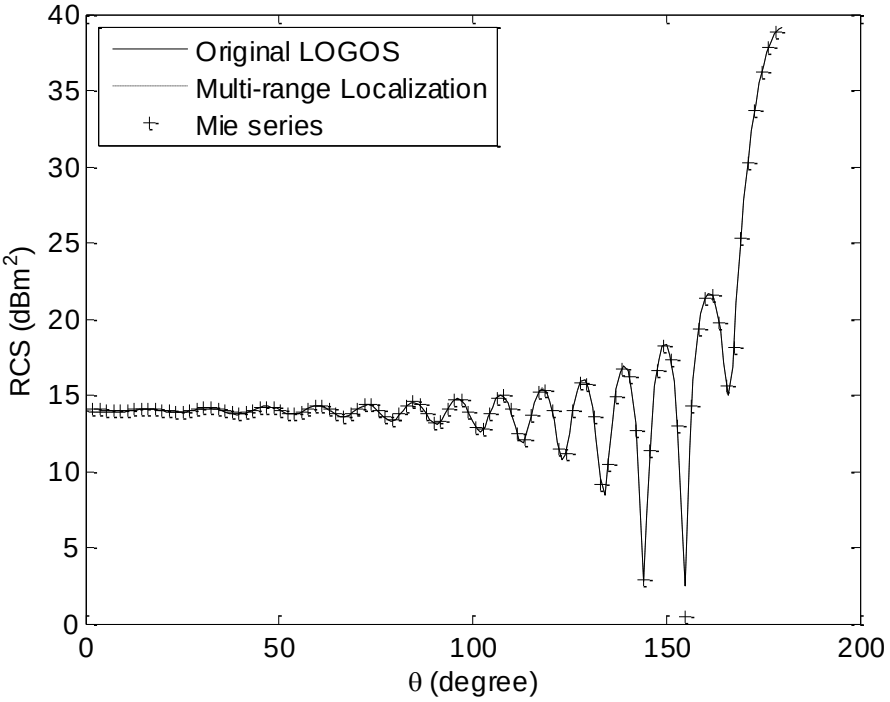


Figure 29. RCS of a PEC sphere with radius 2.83λ. Results are obtained from original LOGOS factorization, multi-range localization factorization and Mie series solution. Factorization tolerance is 10^{-3}.

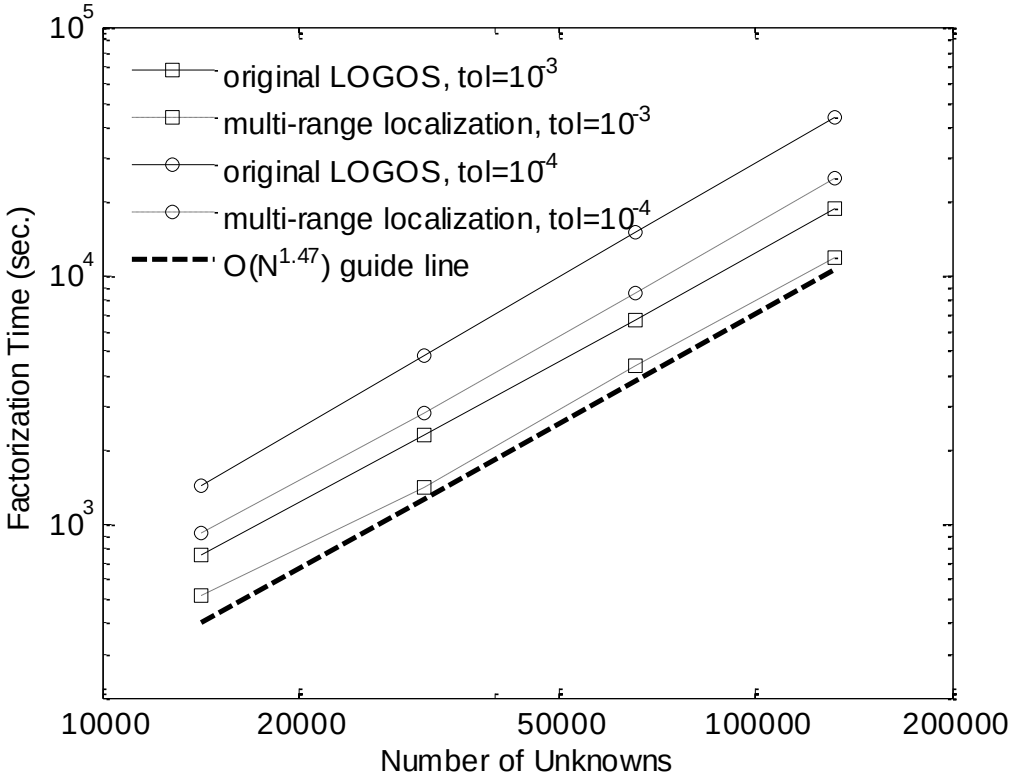


Figure 30. NL-LOGOS factorization time for Nyström discretization of EFIE formulation of scattering from a sequence of PEC spheres. In all cases, the discretization density is 300 DOF/λ^2. The matrix fill tolerance is ten times smaller than the LOGOS factorization tolerance.

Chapter 5. MFD Analysis using NL-LOGOS Modes

In many critical electromagnetic applications, it is important to design relatively small subsystems (such as an antenna) for optimal operation on a much larger existing platform (such as an aircraft). It is also sometimes desired to modify a small region of a large object in order to improve its electromagnetic performance as a whole. In most of these scenarios, only those structures in a small, well-defined design region are modified during each design cycle; the remainder of the object (platform region) does not change from one design iteration to another. Such problems require solving the following matrix equation

$$\begin{bmatrix} \mathbf{Z}_{pp} & \mathbf{Z}_{pd} \\ \mathbf{Z}_{dp} & \mathbf{Z}_{dd} \end{bmatrix} \begin{bmatrix} \mathbf{J}_p \\ \mathbf{J}_d \end{bmatrix} = \begin{bmatrix} \mathbf{E}_p \\ \mathbf{E}_d \end{bmatrix}, \tag{5.1}$$

where the subscripts *p* and *d* represents the platform and design regions, respectively. $\mathbf{J}$ and $\mathbf{E}$ are the unknown and excitation vectors, respectively. The system matrix is now broken up into a two-by-two block-matrix. The task here is to solve for the unknowns, $\mathbf{J}$, or other more practical quantities that are functions of $\mathbf{J}$. Most of the existing fast, high-fidelity electromagnetic simulation tools solve large problems iteratively [3, 69, 84, 85]. When dealing with the design tasks described above, they have to re-solve the entire problem (platform + design region) in order to accurately model the *in situ* performance of the subsystem or the performance of the whole system for each new design. Except that the system matrix for the fixed platform region can be reused, no other information about the fixed platform region can be easily reused for solving the equation with different designs. As the platform grows, the large computational costs make this approach infeasible for design problems.

Schur complement method [26] is a simple and efficient method to reuse most of the fixed platform information. Using Schur complement method, the inverse of the system matrix in (5.1) is given by

$$\begin{bmatrix} \mathbf{Z}_{pp} & \mathbf{Z}_{pd} \\ \mathbf{Z}_{dp} & \mathbf{Z}_{dd} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{Z}_{pp}^{-1}\mathbf{Z}_{pd} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{Z}_{pp}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_{dd}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{Z}_{dp}\mathbf{Z}_{pp}^{-1} & \mathbf{I} \end{bmatrix}, \tag{5.2}$$

where

$$\mathbf{X}_{dd} = \mathbf{Z}_{dd} - \mathbf{Z}_{dp}\mathbf{Z}_{pp}^{-1}\mathbf{Z}_{pd}\,. \tag{5.3}$$

In (5.2), $\mathbf{Z}_{pp}$ is inverted once and reused for all designs. Therefore, the design efficiency can be improved.

The off-diagonal blocks, $\mathbf{Z}_{dp}$ and $\mathbf{Z}_{pd}$, of the system matrix cannot be reused in the Schur complement solution method as given by (5.2) because they are design dependent. One way to break this dependency is to introduce a fixed equivalent surface enclosing the design structure. According to Green's theorem [30], the direct interaction between the platform and design structures can be represented indirectly through the usage of the equivalent surface as

$$\mathbf{Z}_{dp} = \mathbf{Z}_{dx}\mathbf{Z}_{xp}, \tag{5.4}$$

$$\mathbf{Z}_{pd} = \mathbf{Z}_{px}\mathbf{Z}_{xd}, \tag{5.5}$$

where the subscript, *x*, represents the equivalent surface, which will be referred to as the 'Green-box'. With (5.4) and (5.5) written, the terms, $\mathbf{Z}_{xp}$ and $\mathbf{Z}_{px}$, can be filled once and reused for all designs. The quantities in (5.2) that cannot be reused are all independent of the fixed platform structures. Therefore, all information related to the fixed parts of a design problem is reused in a method that combine the Schur complement method and the Green-box. Chakraborty and Jandhyala [27] reported a similar method with slightly different formulation and discussed its superior performance as compared to that of both the direct inversion method and the Schur complement methods without Green-box. The formulation used by the MFD method discussed in this paper differs from what used by Chakraborty and Jandhyala in that the equivalent currents on the Green-box never come into the formulation explicitly in the MFD formulation.

One concern on introducing the Green-box to rewrite the off-diagonal blocks is how to deal with the structures that are intersecting with the Green-box. This problem has been addressed by Li and Chew [86]. Essentially, the Green-box is never used to represent interactions that involve mesh cells that are intersecting or near to the Green-box; the Green-box is only used to represent interactions between mesh cells that are both far from the Green-box. Here, the region that is near to the Green-box is referred to as the buffer region. Any mesh cells whose centers fall in the buffer region are referred to as buffer cells. The matrix elements involving the buffer cells are all filled in directly

without referring to the Green-box. As a result, the system matrix looks like (See Appendix A for the details)

$$\begin{bmatrix} \mathbf{Z}_{pp} & \mathbf{Z}_{pb} & \mathbf{Z}_{px}\mathbf{Z}_{xd} \\ \mathbf{Z}_{bp} & \mathbf{Z}_{bb} & \mathbf{Z}_{bd} \\ \mathbf{Z}_{dx}\mathbf{Z}_{xp} & \mathbf{Z}_{db} & \mathbf{Z}_{dd} \end{bmatrix}, \tag{5.6}$$

where the subscript, *b*, represents the buffer region. It should be noted that the buffer region introduced here is actually part of the originally fixed platform region. Therefore, the structures in the buffer region is fixed during the design process. The subscript, *p*, in (5.6) has been used to represent the part of the platform that excludes the buffer region. Figure 31 illustrates the definition of the regions over a simple structure.

The matrix (5.6) is the system matrix utilized by the MFD method to represent a general design problem. It reduces to the two-by-two system matrix

$$\begin{bmatrix} \mathbf{Z}_{pp} & \mathbf{Z}_{px}\mathbf{Z}_{xd} \\ \mathbf{Z}_{dx}\mathbf{Z}_{xp} & \mathbf{Z}_{dd} \end{bmatrix} \tag{5.7}$$

with the following symbolic substitutions:

$$\mathbf{Z}_{pp} \Leftrightarrow \begin{bmatrix} \mathbf{Z}_{pp} & \mathbf{Z}_{pb} \\ \mathbf{Z}_{bp} & \mathbf{Z}_{bb} \end{bmatrix}, \tag{5.8}$$

$$\mathbf{Z}_{px}\mathbf{Z}_{xd} \Leftrightarrow \begin{bmatrix} \mathbf{Z}_{px}\mathbf{Z}_{xd} \\ \mathbf{Z}_{bd} \end{bmatrix}, \tag{5.9}$$

$$\mathbf{Z}_{dx}\mathbf{Z}_{xp} \Leftrightarrow \begin{bmatrix} \mathbf{Z}_{dx}\mathbf{Z}_{xp} & \mathbf{Z}_{db} \end{bmatrix}. \tag{5.10}$$

The Schur complement method of inverting the system matrix (5.7) can also be used to invert (5.6). In fact, it is sufficient to work with (5.7) instead of (5.6) to derive the MFD method. The derivation for general cases represented by (5.6) follows the same procedure with symbolic substitutions (5.8) to (5.10).

As the problem size grows, both $\mathbf{Z}_{pp}^{-1}$ and $\mathbf{X}_{dd}^{-1}$ in (5.2) need to be obtained using fast, efficient methods. Recently, various fast direct solvers based on sparse representation of system matrix [20, 21, 24, 25, 70-77, 79, 82, 87-93] have been proposed. With direct solvers, the inverse (or a representation of the inverse) of a matrix can be obtained. It becomes apparent that any of these methods can be combined with the Schur complement method to solve the above design problems in a modular way, i.e. the

inverse of the platform system matrix, $\mathbf{Z}_{pp}^{-1}$, can be obtained once and reused for all the designs. However, the inverse of Schur complement, $\mathbf{X}_{dd}$, is difficult because it is not easy to obtain a sparse representation for it due to the second term in (5.3). Furthermore, it is not clear how different that $\mathbf{X}_{dd}$ may behaves as compared to $\mathbf{Z}_{dd}$ due to the second term. It is one of the purpose of this paper to show how to process $\mathbf{X}_{dd}$ using LOGOS. The MFD method derived in this paper also reduces the off-diagonal blocks, $\mathbf{Z}_{px}\mathbf{Z}_{xd}$ and $\mathbf{Z}_{dx}\mathbf{Z}_{xp}$, that representing the interaction between the platform and design structures.

The basic idea of MFD analysis method with 2D and 3D numerical results has been presented earlier [94, 95]. The reduced order model (ROM) that could be derived from the MFD method is also discussed in [96]. This chapter presents the detailed derivation of the MFD analysis method based on the non-overlapping, non-radiating LOGOS framework [74-79], which has been used to develop an $O(N \log N)$ direct solvers for low frequency electromagnetic applications. 3D numerical results are used to demonstrate the efficiency of the MFD method and the ROM derived from it.

5.1. Modular LOGOS Factorization of Platform Region

Without losing generality, the system matrix (5.7) is used to illustrate the derivation of MFD method for simplicity in notation. The substitutions (5.8) to (5.10) can be used to obtain the derivation for more general cases.

As has been discussed earlier, it is clear that any direct solvers can be used to implement $\mathbf{Z}_{pp}^{-1}$ and Schur complement solution process can reuse $\mathbf{Z}_{pp}^{-1}$ for any specific designs. However, the off-diagonal blocks, $\mathbf{Z}_{pd}$ and $\mathbf{Z}_{dp}$, are not processed by this method. For problems where the platform unknowns is more than the design unknown, there are many redundancies on the platform contributing to the interactions between platform and design structures. The LOGOS factorization discussed in this section is one algebraic and error-controllable way of filtering out those redundancies. At the same time, the LOGOS factorization also constructs $\mathbf{Z}_{pp}^{-1}$.

The LOGOS procedure discussed in this section can be looked as a procedure to find the LOGOS modes for the full system matrix (5.7) even though the design structure has not been specified. The LOGOS modes can only exist on the platform structure at this stage. To find the non-radiating LOGOS modes on the platform structure, the first block column, $\begin{bmatrix} \mathbf{Z}_{pp} \\ \mathbf{Z}_{dx}\mathbf{Z}_{xp} \end{bmatrix}$, of the system matrix (5.7) is analyzed with the LOGOS procedure [78]. Since the localized non-radiating LOGOS modes on platform is required to radiate no energy (within prescribed tolerance) to the undetermined design structure, it becomes necessary to require that the localized non-radiating LOGOS modes radiate no energy (within prescribed tolerance) on to the Green-box. This is equivalent to work with $\begin{bmatrix} \mathbf{Z}_{pp} \\ \mathbf{Z}_{xp} \end{bmatrix}$ during LOGOS factorization. The error introduced into the LOGOS factorization can be well controlled [80]. However, the energy radiated by the platform non-radiating LOGOS modes and received by a specific design is not guaranteed to be small because the exact values of $\mathbf{Z}_{dx}$ cannot be taken into account before a specific design is known. One way to solve this problem is to propose a few typical design structures and find

$$\alpha = \max \left\| \mathbf{Z}_{dx} \right\|_F , \tag{5.11}$$

where $\left\| \cdot \right\|_F$ means Frobenius norm. Assume for any subsequent designs, $\left\| \mathbf{Z}_{dx} \right\|_F \le \alpha$, then

$$LOGOS\left(\begin{bmatrix} \mathbf{Z}_{pp} \\ \mathbf{Z}_{dx}\mathbf{Z}_{xp} \end{bmatrix} \right) = LOGOS\left(\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_{dx} \end{bmatrix} \begin{bmatrix} \mathbf{Z}_{pp} \\ \mathbf{Z}_{xp} \end{bmatrix} \right) \supseteq LOGOS\left(\begin{bmatrix} \mathbf{Z}_{pp} \\ \alpha\mathbf{Z}_{xp} \end{bmatrix} \right), \tag{5.12}$$

where $LOGOS(\mathbf{A})$ is used to indicate the LOGOS modes derived from the matrix $\mathbf{A}$. $\left\| \mathbf{Z}_{dx} \right\|_F \le \alpha$ dictates that the energy radiated by the platform and collected by the Green-box is not magnified more than α times when they are received by the design structure. The last step in (5.12) means that the platform non-radiating LOGOS modes that do not radiate, within some controlled error, onto the Green-box also do not radiate into the design-region. Consequently, a modular factorization method for the platform region, taking into account the Green-box, can be obtained according to the last step of (5.12). The parameter α is used only for controlling the accuracy of the LOGOS modes. It does

not change the matrix elements. α can also be updated with $\|\mathbf{Z}_{dx}\|_F$ if it is found that $\|\mathbf{Z}_{dx}\|_F > \alpha$ for a trial design and used for the subsequent trials.

Based on the above analysis, the LOGOS factorization for the platform region can be carried out by working with the platform system matrix

$$\mathbf{Z}_p = \begin{bmatrix} \mathbf{Z}_{pp} & \mathbf{Z}_{px} \\ \mathbf{Z}_{xp} & \mathbf{0} \end{bmatrix}. \tag{5.13}$$

Assume the number of levels of the tree used to decompose the platform and Green-box is L. At the finest level L, define $\mathbf{Z}_{p,L+1}^{(NN)} = \mathbf{Z}_{p,L} = \mathbf{Z}_p$ so that the level index used in the following derivation is valid for all levels. The multilevel LOGOS factorization produces the following LOGOS modes for $\mathbf{Z}_p$ at level $l = L, \cdots, 1$

$$\mathbf{\Lambda}_{p,l} = \begin{bmatrix} \mathbf{\Lambda}_{p,l}^{(L)} & \mathbf{\Lambda}_{p,l}^{(N)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{5.14}$$

and

$$\mathbf{P}_{p,l} = \begin{bmatrix} \mathbf{P}_{p,l}^{(L)} & \mathbf{P}_{p,l}^{(N)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{5.15}$$

where the subscript p denotes the platform region. The superscript (L) and (N) denotes the 'localized' and 'non-localized' parts of the modes. Projecting $\mathbf{Z}_{p,l+1}^{(NN)}$ onto the LOGOS modes, we have

$$\tilde{\mathbf{Z}}_{p,l} = \mathbf{P}_{p,l}^H \mathbf{Z}_{p,l+1}^{(NN)} \mathbf{\Lambda}_{p,l} = \begin{bmatrix} \mathbf{I} & \mathbf{Z}_{p,l}^{(LN)} \\ \mathbf{0} & \mathbf{Z}_{p,l}^{(NN)} \end{bmatrix}, \tag{5.16}$$

where

$$\mathbf{Z}_{p,l}^{(LN)} = \begin{bmatrix} \left(\mathbf{P}_{p,l}^{(L)}\right)^H \mathbf{Z}_{pp,l+1}^{(NN)} \mathbf{\Lambda}_{p,l}^{(N)} & \left(\mathbf{P}_{p,l}^{(L)}\right)^H \mathbf{Z}_{px,l+1}^{(N\bullet)} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{pp,l}^{(LN)} & \mathbf{Z}_{px,l}^{(L\bullet)} \end{bmatrix}, \tag{5.17}$$

$$\mathbf{Z}_{p,l}^{(NN)} = \begin{bmatrix} \left(\mathbf{P}_{p,l}^{(N)}\right)^H \mathbf{Z}_{pp,l+1}^{(NN)} \mathbf{\Lambda}_{p,l}^{(N)} & \left(\mathbf{P}_{p,l}^{(N)}\right)^H \mathbf{Z}_{px,l+1}^{(N\bullet)} \\ \mathbf{Z}_{xp,l+1}^{(\bullet N)} \mathbf{\Lambda}_{p,l}^{(N)} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{pp,l}^{(NN)} & \mathbf{Z}_{px,l}^{(N\bullet)} \\ \mathbf{Z}_{xp,l}^{(\bullet N)} & \mathbf{0} \end{bmatrix}. \tag{5.18}$$

The last steps in (5.17) and (5.18) define some short forms of the corresponding terms in their previous steps. The double superscripts of $\mathbf{Z}$ in the parentheses mean that $\mathbf{Z}$ is

multiplied by the left and right multipliers, where the first one denotes the left multiplier and the second one denotes the right multiplier. The multiplier is either the 'localized' or the 'non-localized' part of the LOGOS modes. The dot symbol means no multiplier or identity multiplier. Now, we can factor $\mathbf{Z}_{p,l+1}^{(NN)}$ as

$$\mathbf{Z}_{p,l+1}^{(NN)} = \mathbf{P}_{p,l} \begin{bmatrix} \mathbf{I} & \mathbf{Z}_{p,l}^{(LN)} \\ \mathbf{0} & \mathbf{Z}_{p,l}^{(NN)} \end{bmatrix} \mathbf{\Lambda}_{p,l}^{-1}. \tag{5.19}$$

The multiple scattering between the platform and design structures happens only through the non-local LOGOS modes. It is contained in $\mathbf{Z}_{p,l}^{(NN)}$, which is reduced after each level of factorization. At the coarsest level of LOGOS factorization, $\mathbf{Z}_{xp,1}^{(\cdot N)}$ and $\mathbf{Z}_{px,1}^{(N\cdot)}$ in $\mathbf{Z}_{p,1}^{(NN)}$ consist of the left over multiple scattering interface to the design structure that cannot be reduced further by LOGOS criterion.

The factorization discussed in this section is for the platform system matrix (5.13). The design system matrix can be defined as

$$\mathbf{Z}_d = \begin{bmatrix} \mathbf{0} & \mathbf{Z}_{xd} \\ \mathbf{Z}_{dx} & \mathbf{Z}_{dd} \end{bmatrix}. \tag{5.20}$$

Comparing (5.13) and (5.20) with the full system matrix (5.7), it can be seen that the matrix blocks in (5.13) related to the Green-box (with subscript *x*) serve as an interface to the design system matrix (5.20) which provides similar interfaces. Therefore, the design system matrix can be plugged into the platform system matrices according to (5.7). The factored form of the platform system matrix have more interfaces to the design system matrix. However, the interfaces have never been altered by the factorization. It is still straight forward to plug the design system matrix into the factored form of the platform system matrix by matching the interfaces provided by the Green-box. At the coarsest level of the LOGOS factorization the result of plugging design system matrix into platform system matrix is

$$\begin{bmatrix} \mathbf{Z}_{pp,1}^{(NN)} & \mathbf{Z}_{px,1}^{(N\cdot)} \mathbf{Z}_{xd} \\ \mathbf{Z}_{dx} \mathbf{Z}_{xp,1}^{(\cdot N)} & \mathbf{Z}_{dd} \end{bmatrix}. \tag{5.21}$$

If the design structure is so small that no factorization is necessary, (5.21) can be inverted directly by the Schur complement method with the Schur complement of $\mathbf{Z}_{pp,1}^{(NN)}$ defined by

$$\mathbf{X}_{dd} = \mathbf{Z}_{dd} - \mathbf{Z}_{dx}\mathbf{Z}_{xp,1}^{(\cdot N)}\left(\mathbf{Z}_{pp,1}^{(NN)}\right)^{-1}\mathbf{Z}_{px,1}^{(N\cdot)}\mathbf{Z}_{xd}\,. \tag{5.22}$$

Many problems have large number of unknowns on design structures that makes the direct inversion of (5.22) not practical. The next section discusses how to factor the design system matrix to reduce $\mathbf{X}_{dd}$ before it is inverted.

5.2. Modular LOGOS Factorization of Design Region

After the platform system matrix is factored and a design is specified, the following task is to find the inverse of the matrix given by (5.21). Because the platform structure is available and that the platform has already been factored, the design system matrix can be taken as

$$\mathbf{Z}_d = \begin{bmatrix} \mathbf{0} & \mathbf{Z}_{px,1}^{(N\cdot)}\mathbf{Z}_{xd} \\ \mathbf{Z}_{dx} & \mathbf{Z}_{dd} \end{bmatrix}. \tag{5.23}$$

The same multilevel LOGOS factorization procedure used in the last subsection can be used here to factor (5.23). The difference between the design system matrix given by (5.23) and the platform system matrix given by (5.13) is that the design system matrix includes explicitly the interaction between platform and design by including the term $\mathbf{Z}_{px,1}^{(N\cdot)}$. Without this term, the design region non-radiating LOGOS modes should have been required to radiate no energy (to prescribed tolerance) onto the enclosing Green-box, which is apparently a too stringent requirement. With the term $\mathbf{Z}_{px,1}^{(N\cdot)}$ in (5.23), the design region localized non-radiating LOGOS modes are required to radiate no energy (to prescribed tolerance) onto the non-localized LOGOS modes, which is an even weaker requirement than requiring that the design region localized LOGOS modes radiate no energy onto the platform. Weaker requirements results in more localized LOGOS modes, and therefore, less non-localized modes at the coarsest level. The factorization at each level results in the following LOGOS modes of the system

$$\mathbf{\Lambda}_{d,l'} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_{d,l'}^{(N)} & \mathbf{\Lambda}_{d,l'}^{(L)} \end{bmatrix}, \tag{5.24}$$

$$\mathbf{P}_{d,l'} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{d,l'}^{(N)} & \mathbf{P}_{d,l'}^{(L)} \end{bmatrix}, \tag{5.25}$$

where the level index $l' = L', \cdots, 1$ and L' is the total number of levels of the design region tree, which is not necessarily the same as the platform region tree. The system matrix transforms according to

$$\mathbf{Z}_{d,l'+1}^{(NN)} = \mathbf{P}_{d,l'} \begin{bmatrix} \mathbf{Z}_{d,l'}^{(NN)} & \mathbf{0} \\ \mathbf{Z}_{d,l'}^{(LN)} & \mathbf{I} \end{bmatrix} \mathbf{\Lambda}_{d,l'}^{-1}. \tag{5.26}$$

where

$$\mathbf{Z}_{d,l'}^{(LN)} = \begin{bmatrix} \left(\mathbf{P}_{d,l'}^{(L)}\right)^H \mathbf{Z}_{dx,l'+1}^{(N\bullet)} & \left(\mathbf{P}_{d,l'}^{(L)}\right)^H \mathbf{Z}_{dd,l'+1}^{(NN)} \mathbf{\Lambda}_{d,l'}^{(N)} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{dx,l'}^{(L\bullet)} & \mathbf{Z}_{dd,l'}^{(LN)} \end{bmatrix}, \tag{5.27}$$

$$\mathbf{Z}_{d,l'}^{(NN)} = \begin{bmatrix} \mathbf{0} & \mathbf{Z}_{px,1}^{(N\bullet)} \mathbf{Z}_{xd,l'+1}^{(\bullet N)} \mathbf{\Lambda}_{d,l'}^{(N)} \\ \left(\mathbf{P}_{d,l'}^{(N)}\right)^H \mathbf{Z}_{dx,l'+1}^{(N\bullet)} & \left(\mathbf{P}_{d,l'}^{(N)}\right)^H \mathbf{Z}_{dd,l'+1}^{(NN)} \mathbf{\Lambda}_{d,l'}^{(N)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{Z}_{px,1}^{(N\bullet)} \mathbf{Z}_{xd,l'}^{(\bullet N)} \\ \mathbf{Z}_{dx,l'}^{(N\bullet)} & \mathbf{Z}_{dd,l'}^{(NN)} \end{bmatrix}. \tag{5.28}$$

The next subsection will show that the matrix that need to be inverted finally is

$$\begin{bmatrix} \mathbf{Z}_{pp,1}^{(NN)} & \mathbf{Z}_{px,1}^{(N\bullet)} \mathbf{Z}_{xd,1}^{(\bullet N)} \\ \mathbf{Z}_{dx,1}^{(N\bullet)} \mathbf{Z}_{xp,1}^{(\bullet N)} & \mathbf{Z}_{dd,1}^{(NN)} \end{bmatrix}. \tag{5.29}$$

This matrix is inverted by the Schur complement method to reuse the terms that is related to the platform. The Schur complement of $\mathbf{Z}_{pp,1}^{(NN)}$ is given by

$$\mathbf{X}_{dd} = \mathbf{Z}_{dd,1}^{(NN)} - \mathbf{Z}_{dx,1}^{(N\bullet)} \mathbf{Z}_{xp,1}^{(\bullet N)} \left(\mathbf{Z}_{pp,1}^{(NN)}\right)^{-1} \mathbf{Z}_{px,1}^{(N\bullet)} \mathbf{Z}_{xd,1}^{(\bullet N)}. \tag{5.30}$$

$\mathbf{X}_{dd}$ given by (5.30) have been reduced to include only those DOFs that are necessary to account for the multiple scattering between platform and design structures that is above the prescribed tolerance.

5.3. Solution of Matrix Equation

After the LOGOS factorization of the platform and design regions, the multiple scattering between platform and design regions have been pushed to the coarsest level of

the trees of platform and design regions. The only field that can go through the Green-box from design region to the platform region is given by $\mathbf{Z}_{xd,1}^{(\cdot N)}$. This field is received by the terms $\mathbf{Z}_{px,1}^{(N\cdot)}$ and $\mathbf{Z}_{px,l}^{(L\cdot)}$, where $l = L,\cdots,1$. The only field that can go through the Green-box from platform region to the design region is given by $\mathbf{Z}_{xp,1}^{(\cdot N)}$. This field is received by the terms $\mathbf{Z}_{dx,1}^{(N\cdot)}$ and $\mathbf{Z}_{dx,l'}^{(L\cdot)}$, where $l' = L',\cdots,1$. When plugging the factored form of design system matrix back into the factored form of platform system matrix, $\mathbf{Z}_{xd,1}^{(\cdot N)}$ is always paired with $\mathbf{Z}_{px,1}^{(N\cdot)}$ and $\mathbf{Z}_{px,l}^{(L\cdot)}$; $\mathbf{Z}_{xp,1}^{(\cdot N)}$ is always paired with $\mathbf{Z}_{dx,1}^{(N\cdot)}$ and $\mathbf{Z}_{dx,l'}^{(L\cdot)}$.

To illustrate a simple solution process, the design tree is assumed to be a single level tree, i.e., $L' = 1$. To make the discussion even simpler, it is assumed that the design system matrix is not factored at all. In another word, $\mathbf{Z}_{d,1}^{(NN)} = \mathbf{Z}_{d,1} = \mathbf{Z}_d$. In this case, the LOGOS modes, (5.14) and (5.15), are also the LOGOS modes for the full system matrix (5.7). Therefore, once a specific design is specified by (5.20), the LOGOS factorization for the full system matrix can be obtain by modifying (5.19), (5.17) and (5.18) as

$$\mathbf{Z}_{l+1}^{(NN)} = \mathbf{P}_{p,l} \begin{bmatrix} \mathbf{I} & \mathbf{Z}_l^{(LN)} \\ \mathbf{0} & \mathbf{Z}_l^{(NN)} \end{bmatrix} \mathbf{\Lambda}_{p,l}^{-1}. \tag{5.31}$$

where

$$\mathbf{Z}_l^{(LN)} = \begin{bmatrix} \left(\mathbf{P}_{p,l}^{(L)}\right)^H \mathbf{Z}_{pp,l+1}^{(NN)} \mathbf{\Lambda}_{p,l}^{(N)} & \left(\mathbf{P}_{p,l}^{(L)}\right)^H \mathbf{Z}_{px,l+1}^{(N\cdot)} \mathbf{Z}_{xd} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{pp,l}^{(LN)} & \mathbf{Z}_{px,l}^{(L\cdot)} \mathbf{Z}_{xd} \end{bmatrix}, \tag{5.32}$$

$$\mathbf{Z}_l^{(NN)} = \begin{bmatrix} \left(\mathbf{P}_{p,l}^{(N)}\right)^H \mathbf{Z}_{pp,l+1}^{(NN)} \mathbf{\Lambda}_{p,l}^{(N)} & \left(\mathbf{P}_{p,l}^{(N)}\right)^H \mathbf{Z}_{px,l+1}^{(N\cdot)} \mathbf{Z}_{xd} \\ \mathbf{Z}_{dx} \mathbf{Z}_{xp,l+1}^{(\cdot N)} \mathbf{\Lambda}_{p,l}^{(N)} & \mathbf{Z}_{dd} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{pp,l}^{(NN)} & \mathbf{Z}_{px,l}^{(N\cdot)} \mathbf{Z}_{xd} \\ \mathbf{Z}_{dx} \mathbf{Z}_{xp,l}^{(\cdot N)} & \mathbf{Z}_{dd} \end{bmatrix}. \tag{5.33}$$

$\mathbf{\Lambda}_{p,l}$ and $\mathbf{P}_{p,l}$ are given by (5.14) and (5.15). The inverse of the full system matrix is obtained recursively as

$$\left(\mathbf{Z}_{l+1}^{(NN)}\right)^{-1} = \mathbf{\Lambda}_{p,l} \begin{bmatrix} \mathbf{I} & -\mathbf{Z}_l^{(LN)} \left(\mathbf{Z}_l^{(NN)}\right)^{-1} \\ \mathbf{0} & \left(\mathbf{Z}_l^{(NN)}\right)^{-1} \end{bmatrix} \mathbf{P}_{p,l}^H. \tag{5.34}$$

At level one, (5.31) yields (5.21) and it is inverted by Schur complement method. Thus, the inverse of the original full system matrix, $\mathbf{Z} = \mathbf{Z}_{L+1}^{(NN)}$, is obtained.

If the design system matrix involved in (5.21) is further factored using the method discussed in the subsection 5.2, the LOGOS modes consist of platform and design region LOGOS modes that are independent on each other. The platform region LOGOS modes (5.14) and (5.15) still results in the factorization given by (5.31) to (5.33). Let (5.21) be denoted by

$$\mathbf{Z}_{0,L'} = \begin{bmatrix} \mathbf{Z}_{pp,1}^{(NN)} & \mathbf{Z}_{px,1}^{(N\bullet)}\mathbf{Z}_{xd} \\ \mathbf{Z}_{dx}\mathbf{Z}_{xp,1}^{(\bullet N)} & \mathbf{Z}_{dd} \end{bmatrix}. \tag{5.35}$$

The subscript 0 results from the relation $\mathbf{Z}_{p,1}^{(NN)} = \mathbf{Z}_{p,0}$. Let $\mathbf{Z}_{0,L'+1}^{(NN)} = \mathbf{Z}_{0,L'}$. The design region LOGOS modes (5.24) and (5.25) factors (5.35) as

$$\mathbf{Z}_{0,l'+1}^{(NN)} = \mathbf{P}_{d,l'} \begin{bmatrix} \mathbf{Z}_{0,l'}^{(NN)} & \mathbf{0} \\ \mathbf{Z}_{0,l'}^{(LN)} & \mathbf{I} \end{bmatrix} \mathbf{\Lambda}_{d,l'}^{-1}. \tag{5.36}$$

where

$$\mathbf{Z}_{0,l'}^{(LN)} = \begin{bmatrix} \left(\mathbf{P}_{d,l'}^{(L)}\right)^H \mathbf{Z}_{dx,l'+1}^{(N\bullet)}\mathbf{Z}_{xp,1}^{(\bullet N)} & \left(\mathbf{P}_{d,l'}^{(L)}\right)^H \mathbf{Z}_{dd,l'+1}^{(NN)}\mathbf{\Lambda}_{d,l'}^{(N)} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{dx,l'}^{(L\bullet)}\mathbf{Z}_{xp,1}^{(\bullet N)} & \mathbf{Z}_{dd,l'}^{(LN)} \end{bmatrix}, \tag{5.37}$$

$$\mathbf{Z}_{0,l'}^{(NN)} = \begin{bmatrix} \mathbf{Z}_{pp,1}^{(NN)} & \mathbf{Z}_{px,1}^{(N\bullet)}\mathbf{Z}_{xd,l'+1}^{(\bullet N)}\mathbf{\Lambda}_{d,l'}^{(N)} \\ \left(\mathbf{P}_{d,l'}^{(N)}\right)^H \mathbf{Z}_{dx,l'+1}^{(N\bullet)}\mathbf{Z}_{xp,1}^{(\bullet N)} & \left(\mathbf{P}_{d,l'}^{(N)}\right)^H \mathbf{Z}_{dd,l'+1}^{(NN)}\mathbf{\Lambda}_{d,l'}^{(N)} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{pp,l}^{(NN)} & \mathbf{Z}_{px,1}^{(N\bullet)}\mathbf{Z}_{xd,l'}^{(\bullet N)} \\ \mathbf{Z}_{dx,l'}^{(N\bullet)}\mathbf{Z}_{xp,1}^{(\bullet N)} & \mathbf{Z}_{dd,l'}^{(NN)} \end{bmatrix}. \tag{5.38}$$

The inverse of (5.36) is given recursively by

$$\left(\mathbf{Z}_{0,l'+1}^{(NN)}\right)^{-1} = \mathbf{\Lambda}_{d,l'} \begin{bmatrix} \left(\mathbf{Z}_{0,l'}^{(NN)}\right)^{-1} & \mathbf{0} \\ -\mathbf{Z}_{0,l'}^{(LN)}\left(\mathbf{Z}_{0,l'}^{(NN)}\right)^{-1} & \mathbf{I} \end{bmatrix} \mathbf{P}_{d,l'}^H. \tag{5.39}$$

At level one, $\mathbf{Z}_{0,1}^{(NN)}$ given by (5.38) when $l' = 1$ is the final system matrix that needs to be inverted using Schur complement method. Thus, the inversion of (5.35) and thus the inversion of the full system matrix can be obtained.

5.4. Reduced Order Models

The design problem as formulated by (5.1) can be efficiently solved by combining the Schur complement method and the Green-box. The inverse of the system matrix can be written as

$$\begin{bmatrix} \mathbf{Z}_{pp} & \mathbf{Z}_{pd} \\ \mathbf{Z}_{dp} & \mathbf{Z}_{dd} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{Z}_{pp}^{-1}\mathbf{Z}_{px}\mathbf{Z}_{xd} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{Z}_{pp}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_{dd}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{Z}_{dx}\mathbf{Z}_{xp}\mathbf{Z}_{pp}^{-1} & \mathbf{I} \end{bmatrix}, \tag{5.40}$$

where
$$\mathbf{X}_{dd} = \mathbf{Z}_{dd} - \mathbf{Z}_{dx}\mathbf{Z}_{xp}\mathbf{Z}_{pp}^{-1}\mathbf{Z}_{px}\mathbf{Z}_{xd}. \tag{5.41}$$

The solution for $\mathbf{J}_p$ and $\mathbf{J}_d$ can be written as

$$\mathbf{J}_p = \mathbf{Z}_{pp}^{-1}\mathbf{E}_p - \mathbf{Z}_{pp}^{-1}\mathbf{Z}_{px}\mathbf{Z}_{xd}\mathbf{X}_{dd}^{-1}\left(\mathbf{E}_d - \mathbf{Z}_{dx}\mathbf{Z}_{xp}\mathbf{Z}_{pp}^{-1}\mathbf{E}_p\right), \tag{5.42}$$

$$\mathbf{J}_d = \mathbf{X}_{dd}^{-1}\left(\mathbf{E}_d - \mathbf{Z}_{dx}\mathbf{Z}_{xp}\mathbf{Z}_{pp}^{-1}\mathbf{E}_p\right). \tag{5.43}$$

The solution cost for $\mathbf{J}_d$ during the design process is independent of platform if the platform excitation $\mathbf{E}_p$ is known and fixed because the term $\mathbf{Z}_{xp}\mathbf{Z}_{pp}^{-1}\mathbf{E}_p$ can be computed independent of the design details and reused for every designs. Therefore, the Schur complement method combined with Green-box produces naturally the ROM for the systems contained inside the Green-box.

There are many other quantities that depends on both $\mathbf{J}_p$ and $\mathbf{J}_d$, such as radar cross section (RCS). Let the interested quantity, $\mathbf{F}$, relate to $\mathbf{J}_p$ and $\mathbf{J}_d$ by

$$\mathbf{F} = \mathbf{S}_p\mathbf{J}_p + \mathbf{S}_d\mathbf{J}_d, \tag{5.44}$$

where $\mathbf{S}_p$ and $\mathbf{S}_d$ are referred to as the solution operators. In many cases, the interested quantity is known and fixed in the design process, and so does the solution operator $\mathbf{S}_p$. Substituting (5.42) into (5.44), we have

$$\mathbf{F} = \mathbf{S}_p\mathbf{Z}_{pp}^{-1}\mathbf{E}_p - \mathbf{S}_p\mathbf{Z}_{pp}^{-1}\mathbf{Z}_{px}\mathbf{Z}_{xd}\mathbf{X}_{dd}^{-1}\left(\mathbf{E}_d - \mathbf{Z}_{dx}\mathbf{Z}_{xp}\mathbf{Z}_{pp}^{-1}\mathbf{E}_p\right) + \mathbf{S}_d\mathbf{J}_d. \tag{5.45}$$

The first term and the parts, $\mathbf{S}_p\mathbf{Z}_{pp}^{-1}\mathbf{Z}_{px}$ and $\mathbf{Z}_{xp}\mathbf{Z}_{pp}^{-1}\mathbf{E}_p$, in the second term in (5.45) can be evaluated once the platform is inverted and reused for every design. Therefore, the solution cost for $\mathbf{F}$ during the design cycles does not depend on the size of the platform provided that the number of rows in the solution operator $\mathbf{S}_p$ does not depends on the size of platform. The introduction of solution operators in the Schur complement solution procedure is one way to build ROM for the design problems where the platform excitation is known and the interested quantities are fixed. These are fair requirements for most of the applications.

One special case for the interested quantity $\mathbf{F}$ is that it represents $\mathbf{J}_p$ and $\mathbf{J}_d$. The solution operator obtained by stacking $\mathbf{S}_p$ and $\mathbf{S}_d$ should be identity. The number of rows in $\mathbf{S}_p$ is the same as the number of unknowns in platform. Therefore, the solution cost does depends on the size of the platform.

5.5. Numerical Results

The high-order locally corrected Nyström (LCN) method employing the mixed-order basis functions [35, 40, 43] is used in this section to obtain the matrix equations from the electric field integral equations for the scattering problems. The scatterer surface is meshed using quadrilateral planer cells (So, the geometry is not discretized to high-order). Over each cell, the mixed-order basis functions are chosen to be complete to the order $p+1=3$ and $p=2$ (see notations in [35]) along the two local curve linear coordinate directions. Two sets of basis functions are needed to expand the two current components flowing on the surface cell. Therefore, there are 2 sets of quadrature points on each cell and each set contains 6 quadrature points. The tolerance for LOGOS factorization used in the following examples is 10^{-3}.

5.5.1. The Width of Buffer Region

There are two factors affecting the accuracy of the representation given by (5.4) and (5.5). One factor is the width of the buffer region that determines the minimum distance between the source and field points whose interactions are represented using the Green-box. The smaller the width of the buffer region, the stronger the singular interactions need to be represented. The other factor is the density of sampling on the Green-box surface which determines how accurate the integral derived from the equivalence principle can be approximated. The denser the sampling is, the more accurate the representations, (5.4) and (5.5), are. In order to simplify the sampling process, the Green-box is meshed using the same criteria as those used in meshing the structures. The same Nyström discretization scheme as that used for discretizing the integral equations are used to discretize the integration over the Green-box. The accuracy of the representations, (5.4) and (5.5), is modulated by changing the width of the buffer region.

In order to check the accuracy of the representations given by (5.4) and (5.5), a simple test case is built as shown in Figure 32. The platform and design cubes are of side length 0.2λ. The distance between the two cubes is Δ, which is the parameter used to control the accuracy of using the Green-box. The Green-box is a larger cube with the design cube sits at its center. One side of the Green-box runs through the middle between the platform and design cubes, therefore, the side length of the Green-box is 0.2λ+Δ.

The system matrix for the structure illustrated by Figure 32 is

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_{pp} & \mathbf{Z}_{pd} \\ \mathbf{Z}_{dp} & \mathbf{Z}_{db} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{pp} & \mathbf{Z}_{px}\mathbf{Z}_{xd} \\ \mathbf{Z}_{dx}\mathbf{Z}_{xp} & \mathbf{Z}_{db} \end{bmatrix}. \tag{5.46}$$

The RMS errors of the different matrix blocks are given by

$$\text{Error}\left(\mathbf{Z}_{pd}\right) = \frac{\left\|\mathbf{Z}_{px}\mathbf{Z}_{xd} - \mathbf{Z}_{pd}\right\|_F}{\left\|\mathbf{Z}_{pd}\right\|_F}, \tag{5.47}$$

$$\text{Error}\left(\mathbf{Z}_{dp}\right) = \frac{\left\|\mathbf{Z}_{dx}\mathbf{Z}_{xp} - \mathbf{Z}_{dp}\right\|_F}{\left\|\mathbf{Z}_{dp}\right\|_F}, \tag{5.48}$$

$$\text{Error}\left(\mathbf{Z}\right) = \frac{\left\|\begin{bmatrix} \mathbf{Z}_{pp} & \mathbf{Z}_{px}\mathbf{Z}_{xd} \\ \mathbf{Z}_{dx}\mathbf{Z}_{xp} & \mathbf{Z}_{db} \end{bmatrix} - \mathbf{Z}\right\|_F}{\left\|\mathbf{Z}\right\|_F}. \tag{5.49}$$

Figure 33 shows the RMS errors defined in (5.47)-(5.49). The error decreases as the width of the buffer increases. In order to control the RMS error in the full system matrix to be less than 10^{-3} and 10^{-4}, the width of the buffer should be larger than 0.2λ and 0.3λ respectively. The RMS error shown in Figure 33 decreases rather slowly when the width of the buffer is larger than 0.5λ. Denser sampling on the Green-box is necessary to reduce the error effectively. In the following test cases, the buffer width is kept larger than 0.3λ to achieve RMS error in the full system matrix to be less than 10^{-4}.

5.5.2. Validating MFD Method

A PEC sphere with radius 2λ is used to validate the MFD method. Part of the sphere is defined as the design region and the rest is the platform region. Figure 34 shows the geometry and the Green-box. The cube is the Green-box with side length 2.25λ. The dark patch on the sphere is the design patch defined by intersecting the sphere with a cube of

side length 2λ. The part of the platform enclosed by the Green-box and that outside of the Green-box but within 0.2λ to the Green-box are defined as the buffer structure. The platform is discretized such that there are approximately 300 DOFS per square wavelength. Three different meshes are used to describe the design patch. One is of the same density of the platform mesh. The other two are denser (about 1200 and 4800 DOFS per square wavelength). For this problem, the α in (5.11) is 6.8 for the normal design mesh and 6.9 and 7.2 for the denser design mesh.

The RCS obtained by the MFD method is compared with the Mie series solutions in Figure 35. The numerical results are obtained with $\alpha = 1$ and no significant error is observed. This is because the value of actual α is not large and the LOGOS factorization with tolerance 10^{-3} can still achieve error below 10^{-2}. The dashed curve in Figure 35 is obtained by MFD method without buffer region witch demonstrate large error in RCS.

5.5.3. Efficiency of MFD Method

In order to demonstrate the efficiency of the MFD method, a series of spheres with different radius is chosen. Part of the sphere is defined as the design patch. The definition of the design patch, the Green-box and the buffer region is the same as the last example. The only difference of the different spheres is their radius. The platform consists of the major part of the spheres.

Figure 36 shows the number of non-localized modes at level-1 of the platform tree after the platform factorization. These modes contains all the fields that can go from the platform to the design through the Green-box. The number of these modes are limited by the size of the Green-box [97]. As the size of the platform grows, the number of non-localized modes at level-1 approaches a constant number (about 510 in this case). We do observe that the number of non-localized modes decreases as the platform unknown increase from about 10000. This is because we are using the sphere as an example. When the sphere is small, the design part of the sphere interact strongly with most of the platform part of the sphere because they are near to each other. As the radius of the sphere increases, the design part of the sphere have nearby structures more like a plate, which should have less strong interactions as compared with a sphere.

The efficiency of the MFD method is studied by examining the time spent in a design cycle which includes all operations related to the design structure. Three types of design requirements are examined in the example. The first one requires the solution over the whole structure. This solution can be used to find all quantities that depends on the global solution of the problem, such as RCS. The second type of design requirement is to obtain local quantities within the Green-box. It requires the solution over the design structure and the buffer structure. It also needs the equivalent current distribution over the Green-box that represents the field radiated from platform to design structure. Because only the level-1 non-local LOGOS current modes on the platform radiates non-zero fields into the Green-box, it is enough to have the equivalent currents on the Green-box to represent these fields only. Let $\mathbf{J}_{p,1}^{(N)}$ denotes the non-local LOGOS current, the equivalent current on the Green-box that generate the same fields within the Green-box as $\mathbf{J}_{p,1}^{(N)}$ does is given by

$$\mathbf{J}_{eq} = \mathbf{Z}_{xp,1}^{(\bullet N)} \mathbf{J}_{p,1}^{(N)} \tag{5.50}$$

where all the terms have been represented in the discrete form according to the Nyström method. The third type of design requirements is to obtain the solution over the design structure only. This solution can be used to obtain local quantities about the design structure, such as antenna's input impedance. The three requirements are referred to as type-I, type-II and type-III design task, respectively, in the following discussion.

Figure 37 shows the time spent on different steps during one design cycle. The design matrix setup time is almost constant for all platform sizes. It depends only on the size of the design structure and the Green-box. For type-I design task, the time spent for obtaining the full current distribution over the whole structure increases almost linearly in the simulation range as the size of the platform increases. It is expected to increase more than linear as the problem size grows because the underlying problem is approaching a high-frequency problem.

For type-II and type-III design tasks, the time spent on obtaining the solutions are independent on the size of the platform. Because the design matrix setup time is also constant for all platform sizes, the type-II and type-III design tasks have design cycle that is totally independent on the size of the platform. This nice behavior comes from the fact

that the DOFs that is necessary to represent the interactions between the platform and the small design structures are determined by the electrical size of the design structures [97]. The Green-box is of the similar size as the design structures. It is sufficient to represent the platform-'Green-box' interaction and the 'Green-box'-design interactions with DOFs that depends on the size of design only. Therefore, the design matrix setup time and the type-II and type-III solution time does not depends on the size of the platform.

The design cycle shown in Figure 37 is dominated by the matrix setup and inverse time. However, this time does not depends on the number of right hand sides (RHS). The solution time increase with the increasing of the number of RHS. Therefore, for problems with many RHS, the solution time can dominate the design cycle and the reduction on the solution time becomes more meaningful.

The memory usage for the MFD method is determined by the sparse representation of the system matrix and the LOGOS factorization results [78]. For the type-I design task, all data have to be available during the solution process. The memory usage is high during the solution process. For the type-II and type-III design tasks, only the level-1 non-local projection of the platform system matrix is needed in the solution process (see (5.21)). All other data can be cleared from the memory. Due to the same argument as used for design timing, the memory usage during the design cycle does not depends on the size of the platform, too.

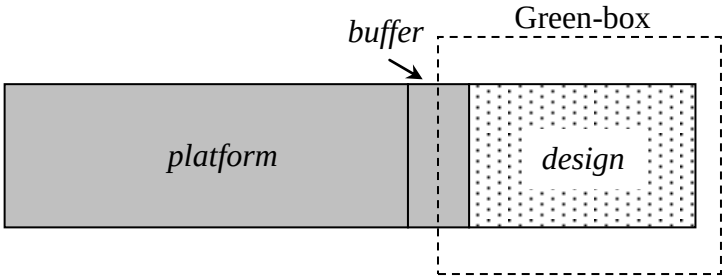


Figure 31. A design problem consists of a large fixed platform and small design structure.

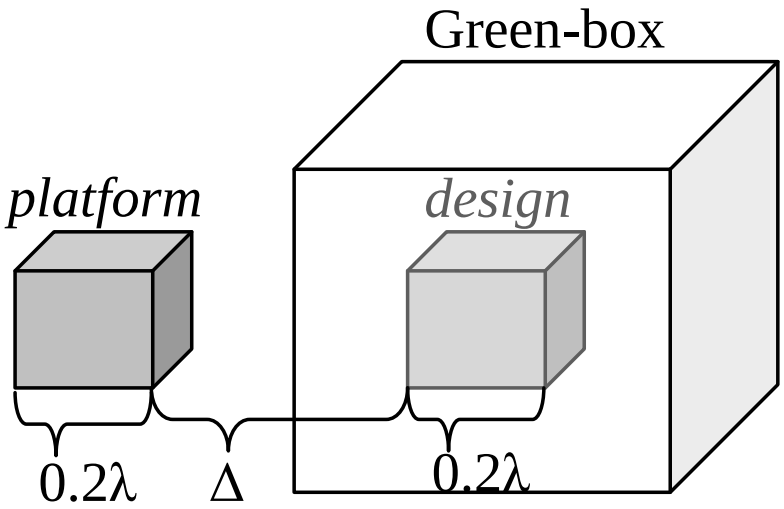


Figure 32. A test case to show the effect of buffer region. The platform and design cubes are of side length 0.2λ. The distance between the two cubes is Δ. The Green-box is a larger cube with the design cube sits at its center. One side of the Green-box runs through the middle between the platform and design cubes.

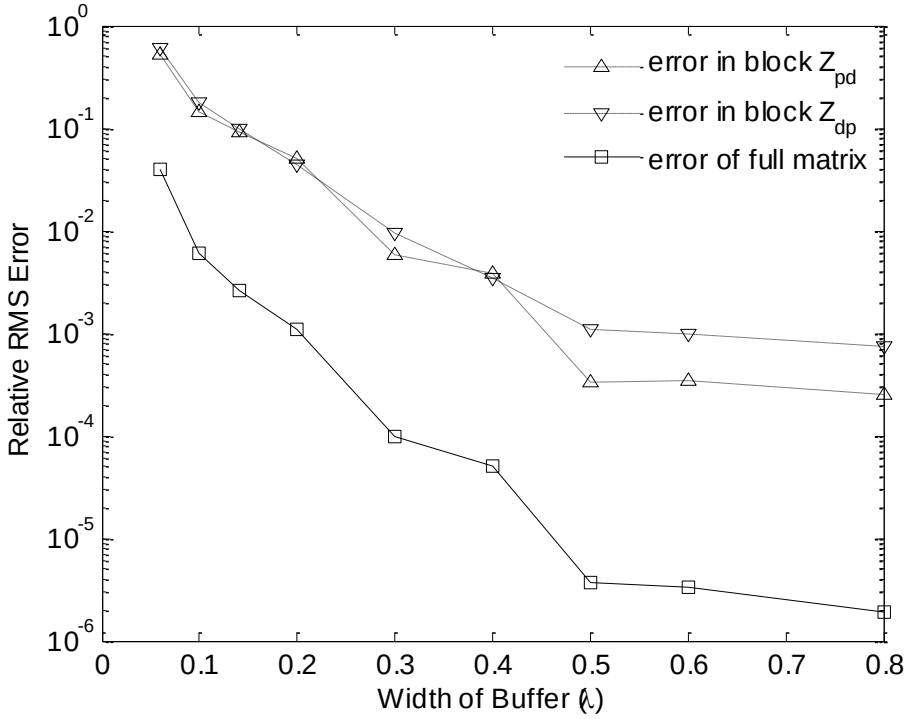


Figure 33. The RMS error of matrices that is represented using equivalence principle.

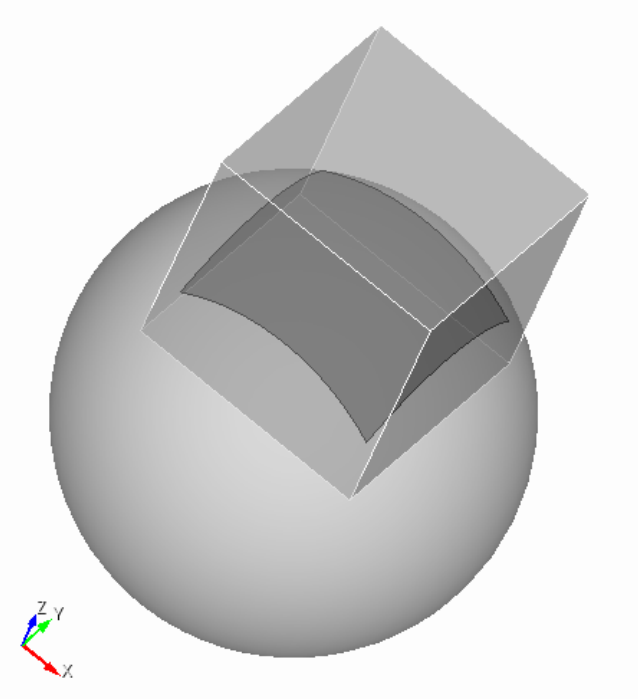


Figure 34. A sphere with a part of it is defined as the design region. The radius of the sphere is 2λ. The cube is the Green-box with side length 2.25λ. The dark patch on the sphere is the design patch defined by intersecting the sphere with a cube with side length 2λ.

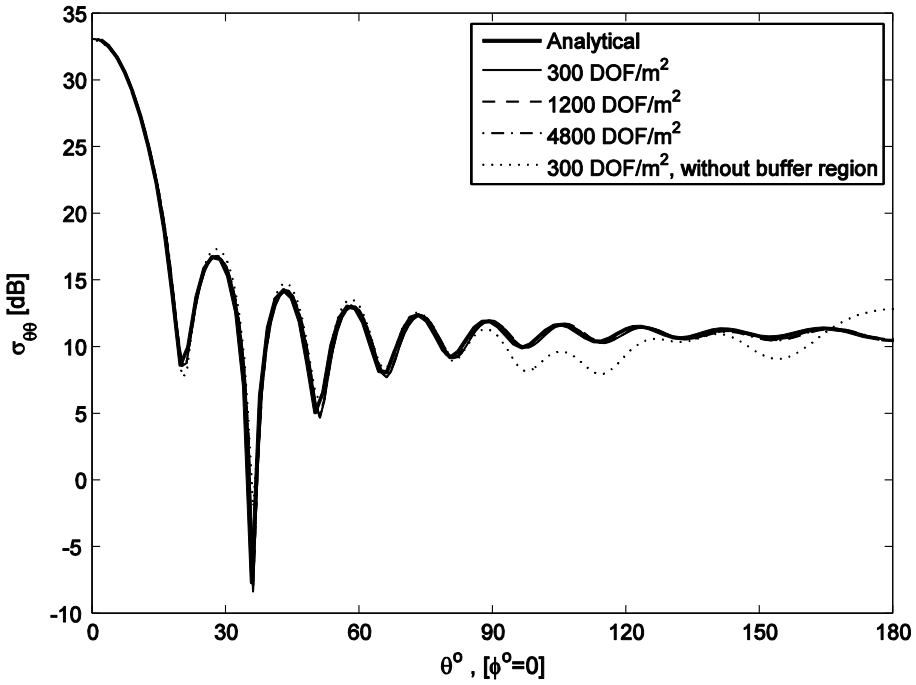


Figure 35. RCS of PEC sphere with radius 2λ.

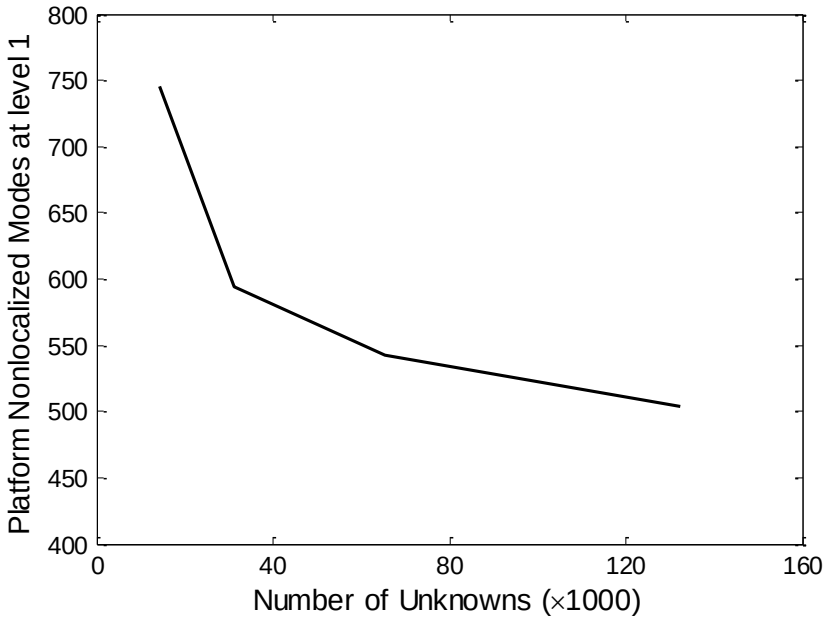


Figure 36. The number of nonlocalized modes at level 1 after the platform factorization. These are the only modes that radiates into the Green-box.

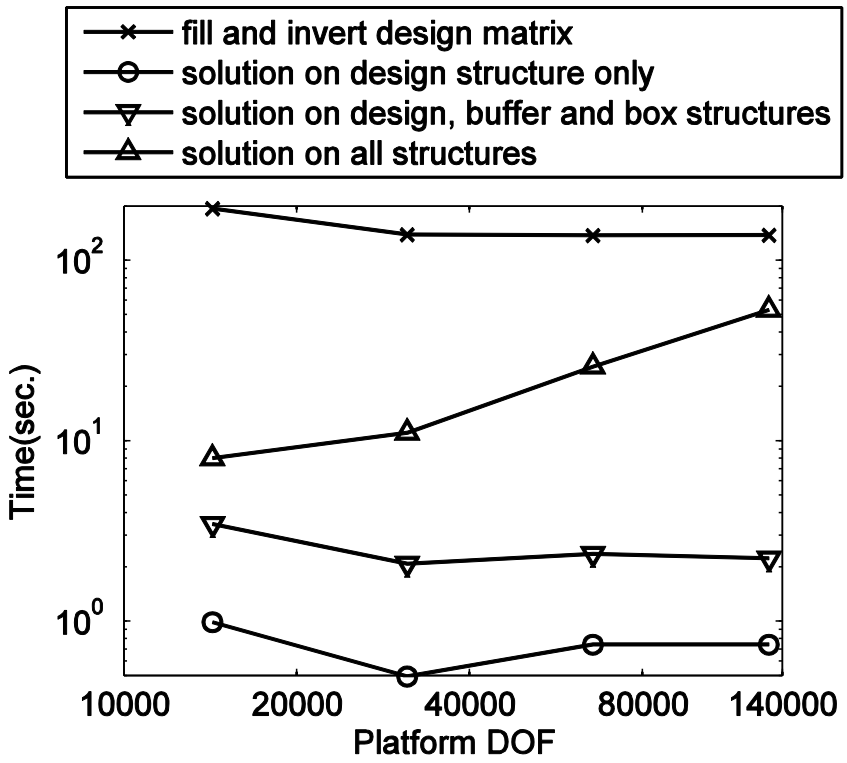


Figure 37. Performance of MFD method for design problem.

Chapter 6. Conclusions

A modular, fast, direct analysis method has been formulated based on the LOGOS concept. The advantage of the proposed MFD method vis-à-vis the standard (non-modular) LOGOS solution method is that the former approach allows the independent factorization of platform region. This in turn implies that matrix factors associated with the fixed platform region can be saved and reused during the design process. This re-use provides significant computational savings when analyzing problems that involve a small design region embedded on/within a large, fixed platform. This type of scenario is encountered in a number of practical applications, such as designing antennas to operate on a larger platform or optimizing a small part of a large circuit.

Besides the reusability of the platform factorization, the non-radiating LOGOS factorization in the MFD method also improves the design efficiency by eliminating the weak multiple scattering modes between the platform and the design structures in an error controlled way. It provides an efficient way of finding the local solutions within the Green-box.

The result of LOGOS factorization in the MFD method can be used to produce reduced order models (ROM) for design tasks that have known excitation field and fixed measuring quantities.

6.1. Contributions

The principle, non-trivial new contributions contained in this dissertation include the following items:

1. A general modular, fast, direct (MFD) analysis method based on the NL-LOGOS factorization has been developed. This LOGOS-based MFD method has been applied to Nyström discretizations of three-dimensional electromagnetic problems involving conductors and materials.
2. A method for deriving reduced-order models (ROMs) from the MFD method has been described. This capability has been demonstrated numerically for three-dimensional electromagnetic problems.

3. A rigorous study in error control for NL-LOGOS factorization has been provided. This study has produced new error control estimates, provided formal justification for previously used error control strategies, and indicated how to properly scale the NL-LOGOS factorization tolerance across multiple levels.
4. A generalization of the NL-LOGOS factorization, referred to as a multi-range localization strategy, has been developed. It has been demonstrated that this strategy significantly improves the overall computational efficiency of the NL-LOGOS factorization. While demonstrated here for the NL-LOGOS factorization of IE formulations, it is expected that the advantages of this improvement to the LOGOS will also be useful for all types of LOGOS-based factorizations. This strategy may be particularly useful for LOGOS-based factorizations of finite element matrices.

6.2. Future Work

As mentioned in the introduction, the current work is summarized by Figure 2. However, the work reported here has actually been performed with a view towards developing the more general MFD analysis and design strategy indicated in Figure 1. Furthermore, while the current work has focused on a specific class of problems and formulations (EM problems at not very high frequency, surface integral equation and Nyström discretization), the general MFD scheme is not necessarily restricted to these specifications. For these reasons, this dissertation provides the basis for the development of much more general MFD design and analysis methods and tools.

Appendix A. Green-box Setup

Consider an electromagnetic design problem with fixed platform structures and variable but confined design structures as shown in Figure 38. All possible design structures are completely confined within a design region, d, with boundary S_d. The fixed platform structures occupy the platform region, p, with boundary S_p. In Figure 38-(a), a gap, which is referred to as the buffer region, is assumed deliberately between the platform and design regions. For applications in which the platform and design regions are connected with each other (see Figure 38-(b)), the buffer region can still be assigned around the design region. In these cases, a portion of the platform structure will be contained within the buffer region. For simplicity, the derivation in Section A.1 assumes the buffer region is empty. The more general case is discussed in Section A.2.

A.1. Green-box for Separated Platform and Design Structures

A virtual surface Γ (as shown in Figure 38) is built to enclose the design structure and passes the middle of the buffer region. Γ does not touch the structures in the platform and design regions. Γ is referred to as the Green-box since the derivations below are fundamentally based on the Green's Identities. The existence of the buffer region is not a necessary condition to derive the algorithm if the singularities associated with the Green-box are properly dealt with. The reason to have this buffer region is to avoid the time-consuming process of dealing with the strongly varying fields on the Green-box Γ when there are sources nearby. In practice, there are a lot of situations where platform and design structures are connected. It is shown in Section A.2 that the derivations here can still be used with minor modifications.

The subscripts p, d and x in the following equations denotes the platform, design and Green-box, respectively. Let $\hat{n}_p$ and $\hat{n}_d$ denote the outward normal of Γ with respect to platform and design region, respectively. Assume there are no sources other than a specified incident field, $\mathbf{E}^{inc}$. For simplicity, only PEC platforms and designs are considered. Let the current distribution on the platform and design structures be denoted

by $\mathbf{J}_p$ and $\mathbf{J}_d$, respectively. The scattered field due to each of these sources can be written in terms of the magnetic vector potentials as

$$\mathbf{E}_\Sigma(\vec{r}) = \frac{\nabla\nabla\cdot\mathbf{A}_\Sigma + k_0^2\mathbf{A}_\Sigma}{j\omega\varepsilon_0}, \tag{A.1}$$

$$\mathbf{H}_\Sigma(\vec{r}) = \nabla\times\mathbf{A}_\Sigma, \tag{A.2}$$

where the magnetic vector potential is given by

$$\mathbf{A}_\Sigma(\vec{r}) = \int_{S_\Sigma}\mathbf{J}_\Sigma(\mathbf{r}')G(\mathbf{r},\mathbf{r}')d\mathbf{r}', \tag{A.3}$$

and $G(\mathbf{r},\mathbf{r}') = e^{-jk_0R}/4\pi R$ is the free-space Green's function, $R = |\mathbf{r}-\mathbf{r}'|$. The subscript Σ can be replaced by either p or d here. It could also be replaced with other subscripts in later discussions to obtain the field generated by the corresponding source.

Consider the equivalent problem in the region bounded by Γ and infinity, where the platform resides. On Γ, the equivalent currents $\mathbf{J}_{\Gamma_d}$ and $\mathbf{M}_{\Gamma_d}$ can be found according to the equivalence principle such that they produce the same fields in region p as that scattered by the design structures. The equivalent currents are given by

$$\mathbf{J}_{\Gamma_d} = \hat{n}_d\times\mathbf{H}_d\big|_{\text{on }\Gamma}, \tag{A.4}$$

$$\mathbf{M}_{\Gamma_d} = \mathbf{E}_d\big|_{\text{on }\Gamma}\times\hat{n}_d. \tag{A.5}$$

The electric field due to the equivalent currents on Γ is

$$\mathbf{E}_{\Gamma_d}(\vec{r}) = \frac{\nabla\nabla\cdot\mathbf{A}_{\Gamma_d} + k_0^2\mathbf{A}_{\Gamma_d}}{j\omega\varepsilon_0} - \nabla\times\mathbf{F}_{\Gamma_d}, \tag{A.6}$$

where $\mathbf{A}_{\Gamma_d}$ is defined in (A.3) and the electric vector potential is defined as

$$\mathbf{F}_\Sigma(\mathbf{r}) = \int_{S_\Sigma}\mathbf{M}_\Sigma(\mathbf{r}')G(\mathbf{r},\mathbf{r}')d\mathbf{r}'. \tag{A.7}$$

Thus, the electric field at any point in the platform region can be written as

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}^{inc}(\mathbf{r}) + \mathbf{E}_p(\mathbf{r}) + \mathbf{E}_{\Gamma_d}(\mathbf{r}). \tag{A.8}$$

Since the source and field points never meet when evaluating $\mathbf{E}_{\Gamma_d}(\mathbf{r})$ in (A.6), the differential operations can be moved inside the integrals. Equation (A.6) can thus be rewritten as

$$\mathbf{E}_{\Gamma_d}(\mathbf{r}) = \int_{S_{\Gamma_d}} \left[\frac{\nabla\nabla G(\mathbf{r},\mathbf{r}') \cdot \mathbf{J}_{\Gamma_d}(\mathbf{r}') + k_0^2 G(\mathbf{r},\mathbf{r}') \mathbf{J}_{\Gamma_d}(\mathbf{r}')}{j\omega\varepsilon_0} - \nabla G(\mathbf{r},\mathbf{r}') \times \mathbf{M}_{\Gamma_d}(\mathbf{r}') \right] d\mathbf{r}' . \tag{A.9}$$

The same procedure can be used to find the electric field in the design region

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}^{inc}(\mathbf{r}) + \mathbf{E}_d(\mathbf{r}) + \mathbf{E}_{\Gamma_p}(\mathbf{r}), \tag{A.10}$$

where

$$\mathbf{E}_{\Gamma_p}(\mathbf{r}) = \int_{S_{\Gamma_p}} \left[\frac{\nabla\nabla G(\mathbf{r},\mathbf{r}') \cdot \mathbf{J}_{\Gamma_p}(\mathbf{r}') + k_0^2 G(\mathbf{r},\mathbf{r}') \mathbf{J}_{\Gamma_p}(\mathbf{r}')}{j\omega\varepsilon_0} - \nabla G(\mathbf{r},\mathbf{r}') \times \mathbf{M}_{\Gamma_p}(\mathbf{r}') \right] d\mathbf{r}' . \tag{A.11}$$

The equivalent currents $\mathbf{J}_{\Gamma_p}$ and $\mathbf{M}_{\Gamma_p}$ are given by

$$\mathbf{J}_{\Gamma_p} = \hat{n}_p \times \mathbf{H}_p\big|_{\text{on }\Gamma}, \tag{A.12}$$

$$\mathbf{M}_{\Gamma_p} = \mathbf{E}_p\big|_{\text{on }\Gamma} \times \hat{n}_p . \tag{A.13}$$

Following the method of moments (MoM) procedure [98], approximating the surface currents on the platform and design structures by a basis function expansion

$$\mathbf{J}_{p(d)} = \sum_{n=1}^{N_{p(d)}} C_n \mathbf{B}_n , \tag{A.14}$$

and testing the fields on the surface of the structures by a set of testing functions, $\mathbf{T}_m$, a linear system of equations can be obtained from (A.8) and (A.10):

$$-\int_{S_m} \mathbf{T}_m(\mathbf{r}) \cdot \mathbf{E}^{inc}(\mathbf{r}) d\mathbf{r} = \sum_{n=1}^{N_p} C_n \int_{S_m} \mathbf{T}_m(\mathbf{r}) \cdot \mathbf{E}_{p,n}(\mathbf{r}) d\mathbf{r} + \sum_{n=1}^{N_d} C_n \int_{S_m} \mathbf{T}_m(\mathbf{r}) \cdot \mathbf{E}_{\Gamma_d,n}(\mathbf{r}) d\mathbf{r} , \tag{A.15}$$

$$
\begin{aligned}
&-\int_{S_m} \mathbf{T}_m(\mathbf{r}) \cdot \mathbf{E}^{inc}(\mathbf{r})\, d\mathbf{r} \\
&= \sum_{n=1}^{N_d} C_n \int_{S_m} \mathbf{T}_m(\mathbf{r}) \cdot \mathbf{E}_{d,n}(\mathbf{r})\, d\mathbf{r} + \sum_{n=1}^{N_p} C_n \int_{S_m} \mathbf{T}_m(\mathbf{r}) \cdot \mathbf{E}_{\Gamma_p,n}(\mathbf{r})\, d\mathbf{r}
\end{aligned}
\quad . \tag{A.16}
$$

The quantities $\mathbf{E}_{p,n}(\mathbf{r})$ and $\mathbf{E}_{d,n}(\mathbf{r})$ are the electric fields due to current basis $\mathbf{B}_n(\mathbf{r})$ in the platform and design regions, respectively. They are obtained from (A.1) and (A.3) by substituting $\mathbf{B}_n$ for $\mathbf{J}_\Sigma(\mathbf{r}')$ in (A.3). $\mathbf{E}_{\Gamma_d,n}(\mathbf{r})$ and $\mathbf{E}_{\Gamma_p,n}(\mathbf{r})$ are the fields due to the current basis $\mathbf{B}_n$ in the design and platform regions, respectively. Each of them has been obtained indirectly using (A.9) and (A.11), where the equivalent currents are now due to the fields radiated by a current basis $\mathbf{B}_n$ in the design or platform regions. If they had been obtained directly in the same way as $\mathbf{E}_{p,n}(\mathbf{r})$ and $\mathbf{E}_{d,n}(\mathbf{r})$, (A.15) and (A.16) would be the ordinary matrix equations resulting from an application of MoM to the EFIE.

The equivalent currents can be obtained using (A.4), (A.5), (A.12), (A.13) and (A.1)-(A.3). The results are

$$
\mathbf{J}_{\Gamma_d,n}(\mathbf{r}') = \hat{n}_d(\mathbf{r}') \times \int_{S_n \text{ in } d} \nabla G(\mathbf{r}', \mathbf{r}'') \times \mathbf{B}_n(\mathbf{r}'')\, d\mathbf{r}'', \tag{A.17}
$$

$$
\mathbf{M}_{\Gamma_d,n}(\mathbf{r}') = -\hat{n}_d \times \int_{S_n \text{ in } d} \frac{\nabla\nabla G(\mathbf{r}', \mathbf{r}'') \cdot \mathbf{B}_n(\mathbf{r}'') + k_0^2 G(\mathbf{r}', \mathbf{r}'') \mathbf{B}_n(\mathbf{r}'')}{j\omega\varepsilon_0} d\mathbf{r}'', \tag{A.18}
$$

$$
\mathbf{J}_{\Gamma_p,n}(\mathbf{r}') = \hat{n}_p(\mathbf{r}') \times \int_{S_n \text{ in } p} \nabla G(\mathbf{r}', \mathbf{r}'') \times \mathbf{B}_n(\mathbf{r}'')\, d\mathbf{r}'', \tag{A.19}
$$

$$
\mathbf{M}_{\Gamma_p,n}(\mathbf{r}') = -\hat{n}_p \times \int_{S_n \text{ in } p} \frac{\nabla\nabla G(\mathbf{r}', \mathbf{r}'') \cdot \mathbf{B}_n(\mathbf{r}'') + k_0^2 G(\mathbf{r}', \mathbf{r}'') \mathbf{B}_n(\mathbf{r}'')}{j\omega\varepsilon_0} d\mathbf{r}''. \tag{A.20}
$$

By changing the order of integrals, the last terms of (A.15) and (A.16) can be rewritten as

$$\int_{S_m} \mathbf{T}_m(\mathbf{r}) \cdot \mathbf{E}_{\Gamma_d,n}(\mathbf{r}) d\mathbf{r} = \int_{S_{\Gamma_d}} \left[\int_{S_m \text{ in } p} \mathbf{T}_m(\mathbf{r}) \cdot \frac{\nabla\nabla G(\mathbf{r},\mathbf{r}') + k_0^2 \overline{\overline{I}} G(\mathbf{r},\mathbf{r}')}{j\omega\varepsilon_0} d\mathbf{r} \cdot \mathbf{J}_{\Gamma_d,n}(\mathbf{r}') \right.$$
$$\left. - \int_{S_m \text{ in } p} \mathbf{T}_m(\mathbf{r}) \times \nabla G(\mathbf{r},\mathbf{r}') d\mathbf{r} \cdot \mathbf{M}_{\Gamma_d,n}(\mathbf{r}') \right] d\mathbf{r}' \quad ,(A.21)$$

and

$$\int_{S_m} \mathbf{T}_m(\mathbf{r}) \cdot \mathbf{E}_{\Gamma_p,n}(\mathbf{r}) d\mathbf{r} = \int_{S_{\Gamma_p}} \left[\int_{S_m \text{ in } d} \mathbf{T}_m(\mathbf{r}) \cdot \frac{\nabla\nabla G(\mathbf{r},\mathbf{r}') + k_0^2 \overline{\overline{I}} G(\mathbf{r},\mathbf{r}')}{j\omega\varepsilon_0} d\mathbf{r} \cdot \mathbf{J}_{\Gamma_p,n}(\mathbf{r}') \right.$$
$$\left. - \int_{S_m \text{ in } d} \mathbf{T}_m(\mathbf{r}) \times \nabla G(\mathbf{r},\mathbf{r}') d\mathbf{r} \cdot \mathbf{M}_{\Gamma_p,n}(\mathbf{r}') \right] d\mathbf{r}' \quad .(A.22)$$

After discretizing the Green-box in a similar manner as that has been done for the platform and design structures, the integral in (A.21) and (A.22) over Green-box can be written as summation of integrals over the discretized cells:

$$\int_{S_\Gamma} f(\mathbf{r}) d\bar{r} = \sum_{n_b=1}^{N_b} \int_{S_{n_b}} f(\mathbf{r}) d\mathbf{r} . \quad (A.23)$$

Due to the assumed gap between the platform and design structures and the position of the Green-box, the integral over each cell can be approximated to desired accuracy by proper fixed point quadrature rules. Thus,

$$\int_{S_\Gamma} f(\mathbf{r}) d\mathbf{r} = \sum_{n_b=1}^{N_b} \sum_{q=1}^{Q} \omega_q f\left[\mathbf{r}(\alpha_q)\right], \quad (A.24)$$

where α and ω are the abscissas and weights of a quadrature rule of order Q. Thus, the integral over Γ_d and Γ_p in (A.21) and (A.22) can be written as summations of functions evaluated at the abscissas on the discretized Green-box and weighted by the quadrature weights.

The linear system of equations (A.15) and (A.16) can now be written in a matrix form as

$$\begin{bmatrix} \mathbf{Z}_{pp} & \mathbf{Z}_{px} \cdot \mathbf{Z}_{xd} \\ \mathbf{Z}_{dx} \cdot \mathbf{Z}_{xp} & \mathbf{Z}_{dd} \end{bmatrix} \begin{bmatrix} \mathbf{J}_p \\ \mathbf{J}_d \end{bmatrix} = \begin{bmatrix} \mathbf{E}_p \\ \mathbf{E}_d \end{bmatrix} . \quad (A.25)$$

where $\mathbf{Z}_{px} = \begin{bmatrix} \mathbf{Z}_{pJ} & \mathbf{Z}_{pM} \end{bmatrix}$, $\mathbf{Z}_{xd} = \begin{bmatrix} \mathbf{Z}_{Jd} \\ \mathbf{Z}_{Md} \end{bmatrix}$, $\mathbf{Z}_{dx} = \begin{bmatrix} \mathbf{Z}_{dJ} & \mathbf{Z}_{dM} \end{bmatrix}$ and $\mathbf{Z}_{xp} = \begin{bmatrix} \mathbf{Z}_{Jp} \\ \mathbf{Z}_{Mp} \end{bmatrix}$. The matrix elements are given by

$$\mathbf{Z}_{pp,mn} = \int_{S_m \text{ in } p} \mathbf{T}_m(\mathbf{r}) \cdot \mathbf{E}_{p,n}(\mathbf{r}) d\mathbf{r}, \tag{A.26}$$

$$\mathbf{Z}_{dd,mn} = \int_{S_m \text{ in } d} \mathbf{T}_m(\mathbf{r}) \cdot \mathbf{E}_{d,n}(\mathbf{r}) d\mathbf{r}, \tag{A.27}$$

$$\mathbf{Z}_{pJ,mk} = \omega_q \int_{S_m \text{ in } p} \mathbf{T}_m(\mathbf{r}) \cdot \frac{\nabla\nabla G(\mathbf{r}, \mathbf{r}_k') + k_0^2 \overline{\overline{I}} G(\mathbf{r}, \mathbf{r}_k')}{j\omega\varepsilon_0} d\mathbf{r}, \tag{A.28}$$

$$\mathbf{Z}_{pM,mk} = -\omega_q \int_{S_m \text{ in } p} \mathbf{T}_m(\mathbf{r}) \times \nabla G(\mathbf{r}, \mathbf{r}_k') d\mathbf{r}, \tag{A.29}$$

$$\mathbf{Z}_{Jd,kn} = \mathbf{J}_{\Gamma_d,n}(\mathbf{r}_k'), \tag{A.30}$$

$$\mathbf{Z}_{Md,kn} = \mathbf{M}_{\Gamma_d,n}(\mathbf{r}_k'), \tag{A.31}$$

$$\mathbf{Z}_{dJ,mk} = \omega_q \int_{S_m \text{ in } d} \mathbf{T}_m(\vec{r}) \cdot \frac{\nabla\nabla G(\mathbf{r}, \mathbf{r}_k') + k_0^2 \overline{\overline{I}} G(\mathbf{r}, \mathbf{r}_k')}{j\omega\varepsilon_0} d\mathbf{r}, \tag{A.32}$$

$$\mathbf{Z}_{dM,mk} = -\omega_q \int_{S_m \text{ in } d} \mathbf{T}_m(\mathbf{r}) \times \nabla G(\mathbf{r}, \mathbf{r}_k') d\mathbf{r}, \tag{A.33}$$

$$\mathbf{Z}_{Jp,kn} = \mathbf{J}_{\Gamma_p,n}(\mathbf{r}_k'), \tag{A.34}$$

$$\mathbf{Z}_{Mp,kn} = \mathbf{M}_{\Gamma_p,n}(\mathbf{r}_k'). \tag{A.35}$$

The excitation vector elements are given by

$$\mathbf{E}_{p,m} = -\int_{S_m \text{ in } p} \mathbf{T}_m(\vec{r}) \cdot \mathbf{E}^{inc}(\mathbf{r}) d\mathbf{r}, \tag{A.36}$$

$$\mathbf{E}_{d,m} = -\int_{S_m \text{ in } d} T_m(\vec{r}) \cdot \mathbf{E}^{inc}(\mathbf{r}) d\mathbf{r}. \tag{A.37}$$

The subscripts J and M represent the equivalent electric and magnetic currents, respectively. With (A.25) written, the Green-box has been used to separate and connect

the platform and design regions. The information about platform and design regions can now be saved separately into the platform and design matrices as

$$\mathbf{Z}_p = \begin{bmatrix} \mathbf{Z}_{pp} & \mathbf{Z}_{px} \\ \mathbf{Z}_{xp} & 0 \end{bmatrix}, \tag{A.38}$$

and

$$\mathbf{Z}_d = \begin{bmatrix} 0 & \mathbf{Z}_{xd} \\ \mathbf{Z}_{dx} & \mathbf{Z}_{dd} \end{bmatrix}, \tag{A.39}$$

respectively. The original matrix should be reconstructed according to (A.25).

A.2. Green-box for Connected Platform and Design Structures

Most practical problems have platform and design structures connected as shown in Fig. 2. The difference between Figure 38-(b) and Figure 38-(a) is the additional connecting structure in the buffer region. The connecting structure is actually part of the platform. Let the letters, p, d, S_p and S_d, denote the same parts of platform and design structures as that in Figure 38-(a). The structure in the buffer region is denoted by 'b'. According to the new setup of the problem, (A.8) and (A.10) should be modified to include the contribution from the structures in the buffer region as

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}^{inc}(\mathbf{r}) + \mathbf{E}_p(\mathbf{r}) + \mathbf{E}_b(\mathbf{r}) + \mathbf{E}_{\Gamma_d}(\mathbf{r}), \tag{A.40}$$

and

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}^{inc}(\mathbf{r}) + \mathbf{E}_d(\mathbf{r}) + \mathbf{E}_b(\mathbf{r}) + \mathbf{E}_{\Gamma_p}(\mathbf{r}). \tag{A.41}$$

where $\mathbf{E}_b(\mathbf{r})$ can be obtained directly using (A.1) and (A.3) by replacing the subscript 'Σ' with 'b'. With such modifications, it can be shown that the matrix equation (A.25) changes to

$$\begin{bmatrix} \mathbf{Z}_{pp} & \mathbf{Z}_{pb} & \mathbf{Z}_{px} \cdot \mathbf{Z}_{xd} \\ \mathbf{Z}_{bp} & \mathbf{Z}_{bb} & \mathbf{Z}_{bd} \\ \mathbf{Z}_{dx} \cdot \mathbf{Z}_{xp} & \mathbf{Z}_{db} & \mathbf{Z}_{dd} \end{bmatrix} \begin{bmatrix} \mathbf{J}_p \\ \mathbf{J}_b \\ \mathbf{J}_d \end{bmatrix} = \begin{bmatrix} \mathbf{E}_p \\ \mathbf{E}_b \\ \mathbf{E}_d \end{bmatrix}, \tag{A.42}$$

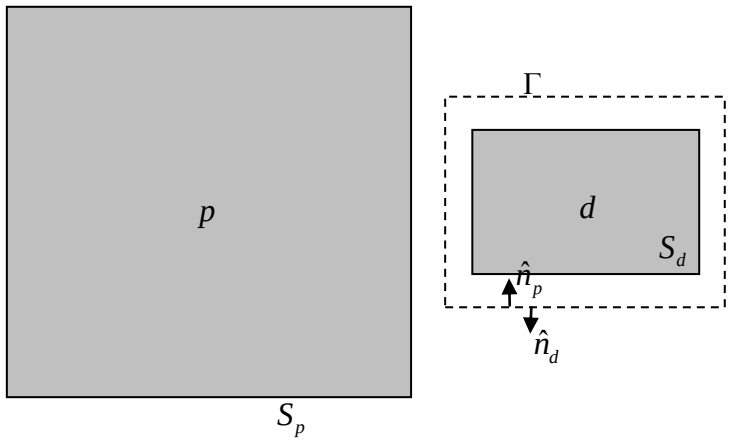
where the matrix elements that are not related to the buffer region structures are still calculated in the same way as those in (A.25); those related to the buffer region can be filled in using either (A.26)or (A.27) by changing the corresponding subscripts. The platform and design matrices change to

$$\mathbf{Z}_p = \begin{bmatrix} \mathbf{Z}_{pp} & \mathbf{Z}_{p,b} & \mathbf{Z}_{px} \\ \mathbf{Z}_{b,p} & \mathbf{Z}_{b,b} & 0 \\ \mathbf{Z}_{xp} & 0 & 0 \end{bmatrix}, \tag{A.43}$$

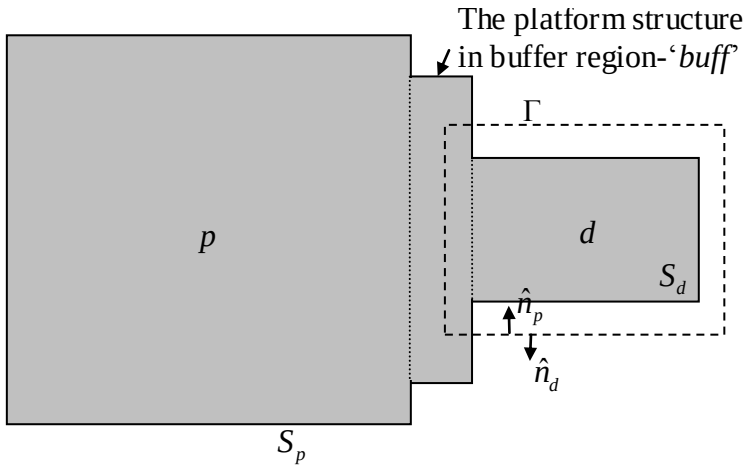
and

$$\mathbf{Z}_d = \begin{bmatrix} 0 & 0 & \mathbf{Z}_{bd} \\ 0 & 0 & \mathbf{Z}_{b,d} \\ \mathbf{Z}_d & \mathbf{Z}_{d,b} & \mathbf{Z}_{dd} \end{bmatrix}. \tag{A.44}$$

By comparing (A.25) and (A.42), it is apparent that the formulation for the connected platform and design structures is a simple modification from that for well separated ones. This modification does not change the modular LOGOS factorization and solution procedures.



(a)



(b)

Figure 38 A design problem consists of a large fixed platform and small design structure. a) The platform and design structures are separated. b) The platform and design structures are connected.

References

[1] Y. Saad, *Iterative methods for sparse linear systems*. Boston MA: WPS, 1996.

[2] J. Jin, *The Finite Element Method in Electromagnetics* Second ed. New York: Wiley, 2002.

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

[4] N. A. Ozdemir and J. F. Lee, "A low rank IE-QR algorithm for matrix compression in volume integral equations," *IEEE transactions on Magnetics,* vol. 40, pp. 1017-1020, 2004.

[5] S. M. Seo and J. F. Lee, "A single-level low rank IE-QR algorithm for PEC scattering problems using EFIE formulation," *IEEE transactions on Antennas and Propagation,* vol. 52, pp. 2141-2146, 2004.

[6] K. Zhao and J. F. Lee, "A single-level dual rank IE-QR algorithm to model large microstrip antenna arrays," *IEEE transactions on Antennas and Propagation,* vol. 52, pp. 2580-2585, 2004.

[7] S. Kapur and D. E. Long, "IES3: A fast integral equation solver for efficient 3-dimensional extraction," in *Proc. ICCAD*, 1997, pp. 448-455.

[8] R. Burkholder and J. F. Lee, "Fast dual MGS block-factorization algorithm for dense MoM matrices," *IEEE transactions on Antennas and Propagation,* vol. 52, pp. 1693-1699, 2004.

[9] J. Parron, J. M. Rius, and J. R. Mosig, "Application of the multilevel matrix decomposition algorithm for the frequency analysis of large microstrip antenna array," *IEEE transactions on Magnetics,* vol. 38, pp. 721-724, 2002.

[10] J. Parron, J. Romeu, J. M. Rius, and J. R. Mosig, "Method of moments enhancement technique for the analysis of Sierpinkski pre-fractal antennas," *IEEE transactions on Antennas and Propagation,* vol. 51, pp. 1872-1876, 2003.

[11] J. M. Rius, J. Parron, E. Ubeda, and J. R. Mosig, "Multilevel matrix decomposition for analysis of electrically large electromagnetic problems in 3-D," *Microwave and Optical Technology Letters,* vol. 22, pp. 177-182, 1999.

[12] A. Breuer, P. Borderies, and J. R. Poirier, "A multilevel implementation of the QR compression for method of moments," *IEEE transactions on Antennas and Propagation,* vol. 51, pp. 2520-2522, 2003.

[13] D. Gope and V. Jandhyala, "Oct-tree-based multilevel low-rank decomposition algorithm for rapid 3-D parasitic extraction," *IEEE Transactions on computer-aided design of integrated circuits and systems,* vol. 23, pp. 1575-1580, 2004.

[14] L. Tsang and Q. Li, "Wave scattering with UV multilevel partitioning method for volume scattering by discrete scatters," *Microwave and Optical Technology Letters,* vol. 41, pp. 354-361, 2004.

[15] G. H. Golub and C. F. Van Loan, *Matrix Computations,* Third ed.: Johns Hopkins University Press, 1996.

[16] I. S. Duff, "Direct methods," Technical Report RAL-98-056, Rutherford Appleton Laboratory 1998.

[17] M. Heath, E. Ng, and B. Peyton, "Parallel algorithms for sparse linear systems," *SIAM Review,* vol. 33, pp. 420-460, 1991.

[18] J. W. Demmel, J. R. Gilbert, and X. S. Li, *SuperLU Users' Guide,* 2007.

[19] UMFPACK, "http://www.cise.ufl.edu/research/sparse/umfpack/."
[20] F. X. Canning and K. Rogovin, "Fast direct solution of standard moment-method matrices," *IEEE Antennas and Propagation Magazine,* vol. 40, pp. 15-26, 1998.
[21] F. X. Canning and K. Rogovin, "A universal matrix solver for integral-equation-based problems," *IEEE Antennas and Propagation Magazine,* vol. 45, pp. 19-26, 2003.
[22] Y. Chen, "A fast direct algorithm for the Lippmann-Schwinger integral equation in two dimensions," *Adv. Comput. Math. ,* vol. 16, pp. 175-190, 2002.
[23] W. Hackbusch, B. N. Khoromskij, and R. Kriemann, "Direct Schur complement method by domain decomposition based on H-matrix approximation," *Computing and Visualization in Science,* vol. 8, pp. 179-188, 2004.
[24] P. G. Martinsson and V. Rokhlin, "A fast direct solver for boundary integral equations in two dimensions," *Journal of Computational Physics,* vol. 205, pp. 1-23, May 2005.
[25] R. J. Adams, A. Zhu, and F. X. Canning, "Efficient solution of integral equations in a localizing basis," *Journal of Electromagnetic Waves and Applications,* vol. 19, pp. 1583-1594, 2005.
[26] F. Zhang, *The Schur Complement and Its Applications*. New York: Springer Science, 2005.
[27] S. Chakraborty and V. Jandhyala, "A surface equivalence-based method to enable rapid design and layout iterations of coupled electromagnetic components in integrated packages," in *IEEE 13th Topical Meeting on Electrical Performance of Electronic Packaging* Port, OR, 2004, pp. 45-48.
[28] B. Zhao, "Level of Detail and Resolution Study of Small Bumps on Large Geometry and Their Impact on Overall Scattering Properties," in *Electrical and Computer Engineering*. vol. M.Sc. Lexington: University of Kentucky, 2007.
[29] J. A. Stratton, *Electromagnetic Theory*. New York and London: McGraw-Hill Book Company, Inc., 1941.
[30] R. F. Harrington, *Time-Harmonic Electromagnetic Fields*. New York: IEEE Press, 2001.
[31] J. R. Mautz and R. F. Harrington, "Electromagnetic scattering from a homogeneous material body of revolution," *AEU,* vol. 33, pp. 71-80, 1979.
[32] A. J. Poggio and E. K. Miller, "Integral equation solutions of three dimensional scattering problems," in *Computer Techniques for Electromagnetics,* R. Mittra, Ed. New York: Pergamon, 1973.
[33] C. Müller, *Foundations of the Mathematical Theory of Electromagnetic Waves* vol. 301. Berlin: Springer-Verlag, 1969.
[34] A. Zhu, S. D. Gedney, and J. L. Visher, "A Study of Combined Field Formulations for Material Scattering for a Locally Corrected Nyström Discretization," *IEEE transactions on Antennas and Propagation,* vol. 53, pp. 4111-4120, 2005.
[35] S. D. Gedney, A. Zhu, and C. C. Lu, "Study of Mixed-Order Basis Functions for the Locally-Corrected Nyström Method," *IEEE Transactions on Antennas and Propagation,* vol. 52, pp. 2996-3004, November 2004.
[36] C. A. Brebbia, *The Boundary Element Method for Engineers*: London: Pentech Press, 1978.

[37] K. K. Mei and J. V. Bladel, "Scattering by perfectly-conducting rectangular cylinders," *IEEE transactions on Antennas and Propagation,* vol. 11, pp. 185-192, 1963.
[38] M. G. Andreasen, "Scattering from parallel metallic cylinders with arbitrary cross sections," *IEEE transactions on Antennas and Propagation,* vol. 12, pp. 746-754, 1964.
[39] N. Morita, N. Kumagai, and J. R. Mautz, *Integral Equation Methods for Electromagnetics*. Norwood: Artech House, Inc, 1991.
[40] L. F. Canino, J. J. Ottusch, M. A. Stalzer, J. L. Visher, and S. M. Wandzura, "Numerical solution of the Helmholtz equation in 2D and 3D using a high-order Nyström discretization," *Journal of Computational Physics,* vol. 146, pp. 627-663, 1998.
[41] S. D. Gedney, "Application of the High-Order Nyström Scheme to the Integral Equation Solution of Electromagnetic Interaction Problems," in *IEEE International Symposium on Electromagnetic Compatibility*, Washington, D.C., 2000, pp. 289-294.
[42] F. Caliskan and A. F. Peterson, "The need for mixed-order representations with the locally corrected Nystrom method," *IEEE Antennas and Wireless Propagation Letters,* vol. 2, pp. 72- 73, 2003.
[43] S. D. Gedney, "On Deriving a Locally Corrected Nyström Scheme from a Quadrature Sampled Moment Method," *IEEE Transactions on Antennas and Propagation,* vol. 51, pp. 2402-2412, September 2003.
[44] S. D. Gedney, A. Zhu, W.-H. Tang, G. Liu, and P. Petre, "A Fast, High-Order Quadrature Sampled Pre-Corrected FFT for Electromagnetic Scattering," *Microwave and Optical Technology Letters,* vol. 36, pp. 343-349, March 5 2003.
[45] A. Zhu and S. D. Gedney, "A Quadrature Sampled Pre-Corrected FFT Method for the Electromagnetic Scattering from Inhomogeneous Objects," *IEEE Antennas and Wireless Propagation Letters,* vol. 2, p. in press, 2003.
[46] S. D. Gedney and C. C. Lu, "A high-order locally corrected Nyström solution for a unified volume/surface integral formulation for electromagnetic scattering," Electrical and Computer Engineering, University of Kentucky Final Report MDA972-01-1-0022, DARPA, August 2004.
[47] W. Hackbusch and Z. P. Nowak, "On the fast matrix multiplication in the boundary element method by panel clustering," *Numerische Mathematik,* vol. 54, pp. 463-491, 1989.
[48] S. Sauter, "Variable order panel clustering," *Computing,* vol. 64, pp. 223-261, 1999.
[49] W. Dahmen and R. Schneider, "Wavelets on manifolds I: Construction and domain decomposition," *SIAM Journal of Mathematical Analysis,* vol. 31, pp. 184-230, 1999.
[50] V. Rokhlin, "Rapid solution of integral equations of classical potential theory," *Journal of Computational Physics,* vol. 60, pp. 187-207, 1985.
[51] L. Greengard and V. Rokhlin, "A new version of the fast multipole method for the Laplace in three dimensions," in *Acta Numerica*: Cambridge University Press, 1997, pp. 229-269.

[52] S. Börm and L. Grasedyck, "Low-rank approximation of integral operators by interpolation," *Computing,* vol. 72, pp. 325-332, 2004.
[53] M. Bebendorf, "Approximation of boundary element matrices," *Numerische Mathematik,* vol. 86, pp. 565-589, 2000.
[54] M. Bebendorf and S. Rjasanov, "Adaptive low-rank approximation of colocation matrices," *Computing,* vol. 70, pp. 1-24, 2003.
[55] E. Tyrtyshnikov, "Incomplete cross approximation in the mosaic-skeleton method," *Computing,* vol. 64, pp. 367-380, 2000.
[56] J. Shaeffer and F. X. Canning, "Adaptive cross approximation for MOM matrix fill for PC problem sizes to 157000 unknowns," in *IEEE/ACES International Conference on Wireless Communications and Applied Computational Electromagnetics*. vol. 1 Hawaii, April 2005, pp. 748-753.
[57] K. Zhao, M. N. Vouvakis, and J.-F. Lee, "The adaptive cross approximation algorithm for accelerated method of moments computations of EMC problems," *IEEE Transactions on Electromagnetic Compatibility,* vol. 47, pp. 763-773, 2005.
[58] S. Börm and L. Grasedyck, "Hybrid cross approximation of integral operators," *Numerische Mathematik,* vol. 101, pp. 221-249, 2005.
[59] L. Banjai and W. Hackbusch, "H- and H^2-matrices for low and high frequency Helmholtz equation," Technical Report 17, Max-Planck-Institut für Mathematik in den Naturwissenschaften 2005.
[60] S. Börm, "H^2-matrices - multilevel methods for the approximation of integral operators," *Comput. Vis. Sci.,* vol. 7, pp. 173-181, 2004.
[61] W. Hackbusch and S. Börm, "Data-sparse approximation by adaptive H^2-matrices," *Computing,* vol. 69, pp. 1-35, 2002.
[62] F. X. Canning and K. Rogovin, "Simply sparse, a general compression/solution method for MoM programs," in *IEEE Antennas and Propagation Society International Symposium*, 2002, pp. 234-237.
[63] J. Cheng, S. A. Maloney, R. J. Adams, and F. X. Canning, "Efficient Fill of a Nested Representation of the EFIE at Low Frequencies," in *IEEE AP-S International Symposium* San Diego, CA, 2008, p. s309p4.
[64] A. Zhu, R. J. Adams, and F. X. Canning, "Modified simply sparse method for electromagnetic scattering by PEC," in *IEEE Antennas and Propagation Society International Symposium*, Washington, DC, 2005, pp. 427-430.
[65] R. Coifman, V. Rokhlin, and S. Wandzura, "The Fast Multipole Method: A Pedestrian Prescription," *IEEE Antennas and Propagation Magazine,* vol. 35, pp. 7-12, June 1993.
[66] C.-c. Lu and W. C. Chew, "A multilevel algorithm for solving a boundary integral equation of wave scattering," *Microwave and Optical Technology Letters,* vol. 7, pp. 456-461, 1994.
[67] F. X. Canning, "Compression of Interaction Data using Directional Sources and Testers," in *Non Provisional US Patent Application* USA, 2003.
[68] J. Cheng, X. Xu, and R. J. Adams, "Efficient Fill Algorithm for Multilevel Simply Sparse Method," *Private communication.,* 2008.
[69] W. C. Chew, J. M. Jin, E. Michielssen, and J. M. Song, "Fast solution methods in electromagnetics," *IEEE Transactions on Antennas and Propagation,* vol. 45, pp. 533-543, 1997.

[70] C.-C. Lu and W. C. Chew, "The use of Huygens' equivalence principle for solving the volume integral equation of scattering," *IEEE Transactions on Antennas and Propagation,* vol. 41, pp. 897-904, 1993.
[71] E. Michielssen, A. Boag, and W. C. Chew, "Scattering from eleongated objects: Direct solution in O(Nlog2N) Operations," *IEE Proc. Microwaves and Antennas,* vol. 143, pp. 277-283, 1996.
[72] M. Bebendorf, "Efficient inversion of the galerkin matrix of general second-order elliptic operators with nonsmooth coefficients," *Mathematics of Computation,* vol. 74, pp. 1179-1199, 2005.
[73] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, X. Sun, A. J. van der Veen, and D. White, "Some Fast Algorithms for Sequentially Semiseparable Representations," *SIAM Journal on Matrix Analysis and Applications,* vol. 27, pp. 341-364, 2005.
[74] R. J. Adams, F. X. Canning, and A. Zhu, "Sparse representations of integral equations in a localizing basis," *Microwave and Optical Technology Letters,* vol. 47, pp. 236-240, 2005.
[75] A. Zhu, R. J. Adams, F. X. Canning, and S. D. Gedney, "Sparse solution of an integral equation formulation of scattering from open PEC targets," *Microwave and Optical Technology Letters,* vol. 48, pp. 476-480, 2006.
[76] A. Zhu, R. J. Adams, F. X. Canning, and S. D. Gedney, "Schur factorization of the impedance matrix in a localizing basis," *Journal of Electromagnetic Waves and Applications,* vol. 20, pp. 351-362, 2006.
[77] R. J. Adams, A. Zhu, and F. X. Canning, "Sparse factorization of the TMz impedance matrix in an overlapped localizing basis," *Progress in Electromagnetics Research,* vol. 61, pp. 291-322, 2006.
[78] R. J. Adams, Y. Xu, X. Xu, J.-s. Choi, S. D. Gedney, and F. X. Canning, "Modular fast direct electromagnetic analysis using local-global solution modes," *IEEE transactions on Antennas and Propagation,* vol. 56, pp. 2427-2441, 2008.
[79] Y. Xu, X. Xu, R. J. Adams, S. D. Gedney, and F. X. Canning, "Sparse direct solution of the electric field integral equation using non-overlapped localizing LOGOS modes," *Microwave and Optical Technology Letters,* vol. 50, pp. 303-307, 2008.
[80] X. Xu, Y. Xu, R. J. Adams, S. D. Gedney, and F. X. Canning, "Error Analysis for Matrix Factorizations using Non-overlapped Localizing Basis Functions," *In preparation,* 2008.
[81] Y. Xu, X. Xu, and R. J. Adams, "Improved LOGOS factorization algorithm for low-to-moderate frequencies," Honolulu, HI, 2007, pp. 5579-5582.
[82] J.-S. Choi, R. J. Adams, and F. X. Canning, "Sparse factorization of finite element matrices using overlapped localizing solution modes," *Microwave and Optical Technology Letters,* vol. 50, pp. 1050-1054, 2008.
[83] Y. Xu, X. Xu, R. J. Adams, and F. X. Canning, "Sparse factorizations for fast local mode computations," *In preparation,* 2008.
[84] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, "AIM: Adaptive Integral Method for Solving Large-Scale Electromagnetic Scattering and Radiation Problems," *Radio Science,* vol. 31, pp. 1225-1251, 1996.

[85] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed.: Philadelphia, PA: Society for Industrial and Applied Mathematics, 2003.
[86] M.-K. Li and W. C. Chew, "Wave-Field Interaction With Complex Structures Using Equivalence Principle Algorithm," *IEEE transactions on Antennas and Propagation,* vol. 55, p. 9, 2007.
[87] L. Gurel and W. C. Chew, "Recursive algorithms for calculating the scattering from N strips or patches," *IEEE Transactions on Antennas and Propagation,* vol. 38, pp. 507-515, April 1990 1990.
[88] J. W. H. Liu, "The multifrontal method for sparse matrix solution: Theory and practice," *SIAM Review,* vol. 34, pp. 82-109, 1992.
[89] T. A. Davis and I. S. Duff, "An unsymmetric-pattern multifrontal method for parallel sparse LU factorization," *SIAM Journal of Matrix Analylsis and Applications,* vol. 18, pp. 140-158, 1997.
[90] L. Gurel and W. C. Chew, "Fast direct (noniterative) solvers for integral-equation formulations of scattering problems," in *IEEE Antennas and Propagation International Symposium,* Atlanta, 1998, pp. 298-301.
[91] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, "A supernodal approach to sparse partial pivoting," *SIAM Journal of Matrix Analylsis and Applications,* vol. 20, pp. 720-755, 1999.
[92] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster, "A fully asynchronous multifrontal solver using distributed dynamic scheduling," *SIAM Journal of Matrix Analylsis and Applications,* vol. 23, pp. 15-41, 2001.
[93] T. A. Davis, *Direct Methods for Sparse Linear Systems*. Philadelphia: Society for Industrial and Applied Mathematics 2006.
[94] X. Xu, Y. Xu, and R. J. Adams, "Modular Fast Direct Design and Analysis via LOGOS," in *URSI North American Radio Science Meeting* Ottawa, ON, Canada, 2007, pp. 92-95.
[95] X. Xu, R. J. Adams, Y. Xu, and F. X. Canning, "Modular fast direct analysis method applied to connected platform and design structures," in *IEEE International Symposium on Antennas and Propagation* San Diego, USA, 2008.
[96] X. Xu and R. J. Adams, "Full-wave reduced-order models for local computational analysis," in *IEEE International Symposium on Antennas and Propagation* San Diego, USA, 2008.
[97] O. M. Bucci and G. Francescetti, "On the degrees of freedom of scattered fields," *IEEE transactions on Antennas and Propagation,* vol. 37, pp. 918-926, 1989.
[98] R. F. Harrington, *Field Computation by Moment Methods* New York: IEEE Press, 1993.

Vita

Xin Xu

Born on August 3, 1976 in Siping, China.

Education

- PH. D. in Physics, National University of Singapore, Singapore, 2002–2007
- M. Sc. in Physics, National University of Singapore, Singapore, 1999–2002
- B. Sc. in Physics, Jilin University, China, 1995–1999

Work Experience

- Associate Scientist, Temasek Laboratories, Singapore, 2001–2006

Honors

- URSI Student Fellowship Grant Award, U.S. National Committee of the International Union of Radio Science (USNC/URSI), The National Academies (NAS), 2007
- Research Challenge Trust Fund (RCTF) Fellowships, Commonwealth of Kentucky Research Challenge Trust Fund, 2007–present

Publications

- Xin Xu, Yuan Xu, Robert J. Adams, and F. X. Canning, Modular Fast Direct Analysis using Non-radiating Local-Global Solution Modes, *In preparation*, 2008.
- X. Xu and R. J. Adams, Improved Non-overlapped Localizing Factorization Based on Multi-range Localization, *In preparation*, 2008.
- Y. Xu, X. Xu, R. J. Adams, and F. X. Canning, Sparse factorizations for fast local mode computations, *In preparation*, 2008.
- X. Xu, Y. Xu, R. J. Adams, S. D. Gedney, and F. X. Canning, Error Analysis for Matrix Factorizations using Non-overlapped Localizing Basis Functions, *In preparation*, 2008.
- R. J. Adams, Y. Xu, X. Xu, J.-s. Choi, S. D. Gedney, and F. X. Canning, Modular fast direct electromagnetic analysis using local-global solution modes, *IEEE transactions on Antennas and Propagation*, vol. 56, pp. 2427-2441, 2008.

- X. Xu, R. J. Adams, Y. Xu, and F. X. Canning, Modular fast direct analysis method applied to connected platform and design structures, in IEEE International Symposium on Antennas and Propagation, San Diego, CA, USA, 2008.
- X. Xu and R. J. Adams, Full-wave reduced-order models for local computational analysis, in IEEE International Symposium on Antennas and Propagation San Diego, CA, USA, 2008.
- X. Xu, Y. Xu, and R. J. Adams, Modular Fast Direct Design and Analysis via LOGOS, in URSI North American Radio Science Meeting ,Ottawa, ON, Canada, pp. 92-95, 2007.
- Y. Xu, X. Xu, and R. J. Adams, Improved LOGOS factorization algorithm for low-to-moderate frequencies, Honolulu, HI, pp. 5579-5582, 2007.
- Y. Xu, X. Xu, R. J. Adams, S. D. Gedney, F. X. Canning, Sparse direct solution of the electric field integral equation using non-overlapped localizing LOGOS modes, Microwave and Optical Technology Letters, 50(2), pp303-307, 2007.
- X. Xu, A. Qing, Y. B. Gan, and Y. P. Feng, An Experimental Study on Electromagnetic properties of Random Fiber Composite Materials, Microwave and Optical Technology Letters, 49(1), pp185-190, 2006
- Qing, X. Xu, and Y. B. Gan, Anisotropy of composite materials with inclusion with orientation preference, IEEE Trans. Antennas Propagat., vol. 53, no. 2, pp. 737-744, 2005
- X. Xu, A. Qing, Y. B. Gan, and Y. P. Feng, Effective properties of fiber composite materials, J. Electromag. Waves Appli., Vol. 18, No. 5, pp649-662, 2004
- Qing, X. Xu, and Y. B. Gan, Effective permittivity tensor of Composite Material with aligned spheroidal inclusion, J. Electromag. Waves Appli., vol. 18, no. 7, pp. 899-910, 2004
- Xian Ning Xie, Hong Jing Chung, Hai Xu, Xin Xu, Chorng Haur Sow, and Andrew Thye Shen Wee, Probe-Induced Native Oxide Decomposition and Localized Oxidation on 6H-SiC (0001) Surface: An Atomic Force Microscopy Investigation J. Am. Chem. Soc., 126 (24), 7665 -7675, 2004.

- X. Xu, Y. P. Feng, Excitons in coupled quantum dots, J. Phys. Chem. Solids. Vol. 64 No. 11 pp. 2301-2306, Nov. 2003
- X. Wang, C. F. Wang, Y. B. Gan, X. Xu and L.W. Li, Computation of scattering cross section of targets situated above lossy half space, Electro. Lett. Vol. 39, No. 8, pp683-684, April, 2003
- X. Xu, Y. P. Feng, Quantum Confinement and Excitonic Effects in Vertically Coupled Quantum Dots, Key Engineering Materials, Vol. 227, pp.171-176, 2002
- T. S. Koh, Y. P. Feng, X. Xu and H. N. Spector, Excitons in semiconductor quantum discs, J. Phys. Condensed Matter. Vol. 13, No 7, pp.1485-1498, Feb. 2001