The Dissertation Committee for Paul Hikaru Tsuji
certifies that this is the approved version of the following dissertation:

Fast Algorithms for Frequency-Domain Wave Propagation

Committee:

Lexing Ying, Supervisor

Bjorn Engquist

Sergey Fomel

Omar Ghattas

Kui Ren

Fast Algorithms for Frequency-Domain Wave Propagation

by

Paul Hikaru Tsuji, B.S., M.S.C.A.M

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2012

Dedicated to my family and friends.

Acknowledgments

I wish to thank many people who have supported me over the past 4 years of graduate school. First, I would like to thank my advisor Lexing Ying, who has always steered me in the right direction and given me indispensable knowledge on the field of computational mathematics. Secondly, I would like to thank my fellow student Jack Poulson, who I have collaborated with over the past half-year in developing the parallel spectral element code. Next, I would like to thank Sergey Fomel, who allowed me access to the TACC resources and velocity models which were used in the chapter on seismic wave propagation. Finally, I would like to thank family and friends at ICES for making my time in Austin enjoyable.

Fast Algorithms for Frequency-Domain Wave Propagation

Paul Hikaru Tsuji, Ph.D.
The University of Texas at Austin, 2012

Supervisor: Lexing Ying

High-frequency wave phenomena is observed in many physical settings, most notably in acoustics, electromagnetics, and elasticity. In all of these fields, numerical simulation and modeling of the forward propagation problem is important to the design and analysis of many systems; a few examples which rely on these computations are the development of metamaterial technologies and geophysical prospecting for natural resources. There are two modes of modeling the forward problem: the frequency domain and the time domain. As the title states, this work is concerned with the former regime.

The difficulties of solving the high-frequency wave propagation problem accurately lies in the large number of degrees of freedom required. Conventional wisdom in the computational electromagnetics commmunity suggests that about 10 degrees of freedom per wavelength be used in each coordinate direction to resolve each oscillation. If K is the width of the domain in wavelengths, the number of unknowns N grows as $O(K^2)$ for surface discretizations and $O(K^3)$ for volume discretizations in 3D. The memory requirements and asymptotic complexity estimates of direct algorithms such as the multifrontal method are too costly for such problems. Thus, iterative solvers must be used. In this dissertation, I will present fast algorithms which, in conjunction with GMRES, allow the solution of the forward problem in $O(N)$ or $O(N \log N)$ time.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The fields of computational electromagnetics, acoustics, and elasticity have improved significantly in the past few decades. With the advancement of processor speeds, problems that were deemed impossible to solve years ago are now trivially done in a few seconds. As a result, the frequency band which scientists and engineers can simulate on a computer has expanded greatly. Despite the current hardware trend following Moore's law, however, the need for fast numerical algorithms is ever present.

For the medium-to-high frequency range, there are two solution methods. The first method involves taking the geometric optics approximation of the underlying PDE and assuming the phase and amplitude functions are separated. These approximations are accurate as the frequency approaches infinity, or as the width of the structure being analyzed becomes much larger than the wavelength. The simplification reduces the computational complexity of the problem greatly; unfortunately, these methods are inaccurate in the medium frequency range, as they do not account for diffraction, caustics, creeping waves, and other phenomena seen in full wave theory. The second

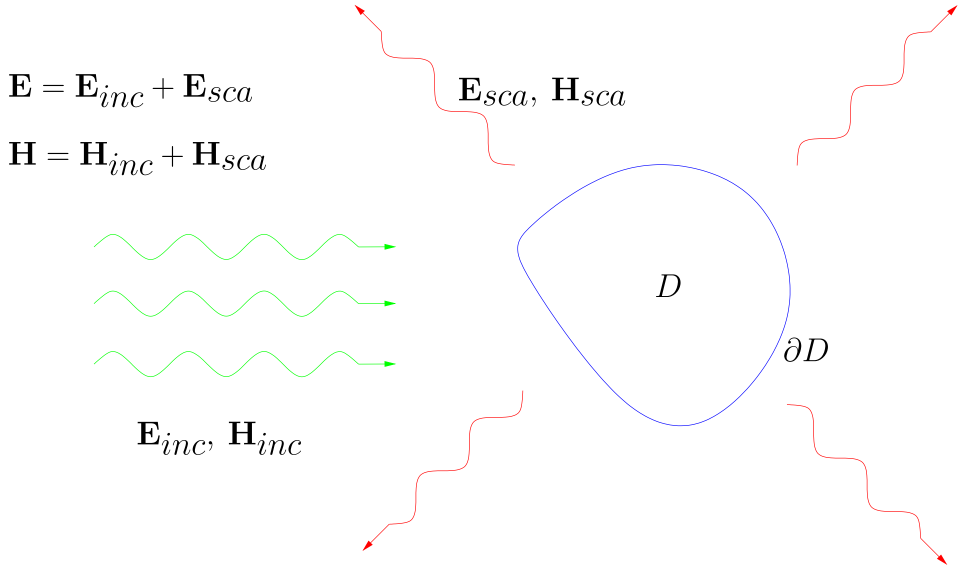

$\mathbf{E} = \mathbf{E}_{inc} + \mathbf{E}_{sca}$
$\mathbf{H} = \mathbf{H}_{inc} + \mathbf{H}_{sca}$
$\mathbf{E}_{sca}$, $\mathbf{H}_{sca}$
D
∂D
$\mathbf{E}_{inc}$, $\mathbf{H}_{inc}$

Figure 1.1: The exterior scattering problem. An incident field (usually a plane wave) is propagating in the medium and reflects off of a scatterer in the domain. The total field $\mathbf{E}$ is comprised of the incident field and reflected or scattered field $\mathbf{E}_{\text{sca}}$.

method is the direct numerical simulation of the true wave equation, whether it is Maxwell's equations, the Helmholtz equation, or the elastic wave equation. This will produce an accurate representation of the physics for all frequencies, but it is very expensive to compute as one enters the high-frequency range. The goal of this thesis is to develop algorithms which allow the efficient solution of high-frequency problems using direct numerical methods.

1.2 Current status of fast solvers for boundary integral equations

For scattering problems in piecewise homogeneous media, the problem can be formulated as a boundary integral equation on the surface of the scattering object, with the unknown quantity being the induced electric current. The general setup of the problem is roughly illustrated in figure 1.1. The strength of this approach is that the outgoing radiation condition is automatically satisfied by the formulation; thus, an absorbing boundary condition does not need to be introduced. Once the integral equation is discretized, the resulting computational task is to solve a dense $N \times N$ linear system of equations, where N is the number of degrees of freedom. Because the system matrix is dense, a standard direct solver such as Gaussian elimination would take $O(N^3)$ operations. Recently, fast direct solvers have been developed for the integral equations in potential theory in 2D and 3D [65, 45], as well as for scattering theory in the low-to-medium frequency range in 2D [66, 64]; these methods rely on the fact that the off-diagonal blocks of the impedance matrix are low-rank, and can be compressed using hierarchical matrices and other low-rank factorization schemes. The complexity estimates of these direct solvers is typically $O(N)$ for non-oscillatory problems and $O(N \log N)$ for low-frequency scattering problems. For the high-frequency regime and 3D problems, however, these estimates break down, and the solvers are no longer efficient.

On the other hand, iterative methods can be used. The boundary element discretization leads to a dense linear system which is well-conditioned for convex scatterers, resulting in a reasonable number of iterations necessary for GMRES convergence. The main computational bottleneck in this situation is the matrix-vector multiplication required at each iteration; because the system matrix is dense, the product takes $O(N^2)$ flops to perform. For large N, this task becomes too costly to compute. To accelerate the matrix-vector product, a variety of techniques have been introduced:

- **Fast multipole methods (FMM)**. The original FMM was developed for potential theory by Greengard and Rokhlin [46], then adapted for the Helmholtz equation and high-frequency applications [81, 82, 22]. In electromagnetics, it is known as the multilevel fast multipole algorithm (MLFMA) [84, 85]. Recent work has produced variants of the FMM which use different expansion techniques [20, 101]. For high-frequency problems, these methods usually scale as $O(N \log N)$.
- **FFT-based methods**. FFT-based methods take advantage of the convolutional structure of the integral operator; by mapping the original sources to equivalent sources on a cartesian grid, one can use the FFT to compute the interactions, then map the potentials back on to the original grid. Both the pre-corrected FFT method [76] and adaptive integral method [12] are variants of this algorithm; in low-frequency applications, these methods scale as $O(N \log N)$.

Recently, the Directional Multilevel Algorithm or Directional FMM was introduced by Enguist and Ying [32]. By utilizing the directional low-rank property in the high-frequency regime, they showed that an $O(N \log N)$ algorithm can be achieved. The approach for Maxwell's equations presented here is an extension of the Directional FMM. This method maintains the

$O(N \log N)$ complexity estimate for all frequencies and is a strong competitor to the most commonly used high-frequency FMM [84] in electromagnetics. In addition to this extension, I present some work on accelerating the uncertainty quantification computations in high-frequency acoustic scattering utilizing the Directional FMM.

1.3 Current status of fast solvers for finite element methods

For fully heterogeneous media, the entire volume must be discretized, making finite element and finite difference methods more attractive. After introducing perfectly matched layers to emulate the radiation condition, the PDE can be discretized on a tetrahedral or hexahedral mesh, resulting in a large sparse linear system; this makes direct solvers such as the multifrontal method [28, 63] attractive. The complexity of the multifrontal factorization is $O(N^{3/2})$ in 2D and $O(N^2)$ in 3D; thus, for 3D problems, the direct solution becomes too expensive in terms of computational time and memory requirements, and iterative methods are more practical.

Unfortunately, these sparse systems are highly indefinite. Preconditioners and iterative methods that would converge nicely for positive-definite problems either completely fail or converge in a number of iterations which depends linearly with frequency; the shortcomings and difficulties of these methods are well documented in [36, 39]. There has been extensive work done on the preconditioning and iterative solvers for frequency-domain wave problems, with the majority of the literature being focused on finite difference methods for the Helmholtz equation. Recently, a preconditioner which has garnered much attention is the shifted Laplacian [9, 37]. The main goal of the method is to “shift” the eigenvalues of the Helmholtz operator away from the origin and into the positive half of the complex plane; in doing so, the condition number is improved and the problem becomes less indefinite. For $\imath = \sqrt{-1}$ and wavenumber κ, discretizing the complex-perturbed operator $\Delta + (1 + \imath\beta)\kappa^2$ for $\beta > 0$ and applying the inverse approximately can act as an effective preconditioner. The inversion can be achieved by either multigrid or incomplete LU decomposition. Despite these advances, convergence of the iterative solver still deteriorates as frequency increases.

Recently, a new family of preconditioners for time-harmonic wave equations, called the “sweeping” preconditioners, was introduced in [34, 35]. Developed for Cartesian finite difference grids of the Helmholtz equation, the initial step is to order the degrees of freedom lexicographically in layers so that the linear system can be written in block tridiagonal form; this process allows a block LDL^t factorization. The main observation then is that the inverse of each Schur complement block on the diagonal of D is the Green’s function of the Dirichlet half-space problem; as a result, they can be represented efficiently using the hierarchical matrix algebra, or by truncating the half-space problem further by moving the perfectly matched layer. The approximate inverse of the LDL^t factorization can be constructed with $O(N)$ complexity in 2D and $O(N^{4/3})$ complexity in 3D, and applied to a right hand side with $O(N)$

complexity in 2D and $O(N \log N)$ complexity in 3D. As a preconditioner, the number of GMRES iterations necessary for convergence is drastically lower compared to other preconditioning techniques, and is essentially independent of frequency. In this dissertation, I will show that these preconditioning methods work for higher-order discretizations on uniform and non-uniform meshes, with more complex physical equations. A highly scalable parallel version of the preconditioner will be presented, with computations performed on the Lonestar machine at the Texas Advanced Computing Center (TACC). I will also provide a proof illustrating the frequency-independent nature of the algorithm.

1.4 Physical Models

Throughout this work, various frequency-domain models will be investigated, with each model capturing a different set of physics. The common denominator in all of these problems is that the physical domain is unbounded; that is, there is no wall or boundary enclosing the system, and wave propagation is outgoing. Thus, each set of equations carries its own outgoing radiation condition, which I will list here:

- For acoustics, we have the Helmholtz equation with an anisotropic coefficient tensor and Sommerfeld radiation condition,

$$\begin{aligned} \nabla \cdot (a \nabla u) + \frac{\omega^2}{c^2} u &= f \qquad \text{in } \mathbb{R}^d && (1.1) \\ \lim_{r \to \infty} r \left(\frac{\partial u}{\partial r} - \imath \kappa u \right) &= 0, && (1.2) \end{aligned}$$

where u is the pressure, c is the wave speed, f is the acoustic source, $r = |\mathbf{r}|$, and the wavenumber in this instance is $\kappa = \frac{\omega}{c}$. The material tensor a takes the form

$$a = \begin{pmatrix} a_{xx} & a_{xy} & a_{xz} \\ a_{yx} & a_{yy} & a_{yz} \\ a_{zx} & a_{zy} & a_{zz} \end{pmatrix}. \tag{1.3}$$

- For electromagnetics, we have Maxwell's equations augmented by the Silver Muller radiation conditions,

$$\begin{aligned} \nabla \times \mathbf{E} &= -\imath\omega\mu_0\mu_r\mathbf{H} && (1.4) \\ \nabla \times \mathbf{H} &= \imath\omega\varepsilon_0\varepsilon_r\mathbf{E} + \mathbf{J} && (1.5) \\ \nabla \cdot (\varepsilon_0\varepsilon_r\mathbf{E}) &= q_e && (1.6) \\ \nabla \cdot (\mu_0\mu_r\mathbf{H}) &= 0 && (1.7) \\ \lim_{|\mathbf{r}|\to\infty} (\mathbf{H} \times \mathbf{r} - |\mathbf{r}|\mathbf{E}) &= 0 && (1.8) \\ \lim_{|\mathbf{r}|\to\infty} (\mathbf{E} \times \mathbf{r} + |\mathbf{r}|\mathbf{H}) &= 0. && (1.9) \end{aligned}$$

Here, $\mathbf{E}$ is the electric field, $\mathbf{H}$ is the magnetic field, $\mathbf{J}$ is the current distribution, ω is the angular frequency, ε_0 is the permittivity of free space, μ_0 is the permeability of free space, $\mathbf{r} \in \mathbb{R}^d$, and $\imath = \sqrt{-1}$. The charge distribution q_e satisfies the continuity equation $\nabla \cdot \mathbf{J} = -\imath\omega q_e$. The free-space wavenumber is defined as $\kappa = \omega\sqrt{\mu_0\varepsilon_0}$; the material is characterized by the relative permittivity and permeability tensors

$$\varepsilon_r = \begin{pmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{pmatrix}, \qquad \mu_r = \begin{pmatrix} \mu_{xx} & \mu_{xy} & \mu_{xz} \\ \mu_{yx} & \mu_{yy} & \mu_{yz} \\ \mu_{zx} & \mu_{zy} & \mu_{zz} \end{pmatrix}. \tag{1.10}$$

For the 2D case, I will consider transverse electric propagation modes, where **H** is oriented in the z direction and **E** is only in the x-y plane. The material parameters are then simplified to

$$\varepsilon_r = \begin{pmatrix} \varepsilon_{xx} & \varepsilon_{xy} \\ \varepsilon_{yx} & \varepsilon_{yy} \end{pmatrix}, \qquad \mu_r = \mu_{zz} \tag{1.11}$$

- For linear elasticity, the time-harmonic problem for the displacement field $u = (u_1, u_2, u_3)$ is given as

$$\begin{aligned} -(C_{ijkl}u_{k,l})_{,j} - \omega^2 \rho u_i &= f_i \quad \text{in } \mathbb{R}^3 \\ \lim_{r\to\infty} r\left(\frac{\partial u_{\mathrm{s}}}{\partial r} - \imath\kappa_{\mathrm{s}} u_{\mathrm{s}}\right) &= 0 \\ \lim_{r\to\infty} r\left(\frac{\partial u_{\mathrm{p}}}{\partial r} - \imath\kappa_{\mathrm{p}} u_{\mathrm{p}}\right) &= 0. \end{aligned} \tag{1.12}$$

where C_{ijkl} is the fourth-order tensor, u is the displacement vector, and ρ is the material density. The last two equations are the Kupradze-Sommerfeld radiation conditions [14], where the displacement field can be decomposed into its solenoidal part u_{s} and irrotational part u_{p}, and κ_{s} and κ_{p} are the wavenumbers for the S-wave and P-wave, respectively.

Part I:
Fast Algorithms for Boundary Integral Equations

The first part of this thesis will be concerning fast algorithms for boundary integral equations. Chapter 2 will detail the work I have done in developing fast multipole methods for Maxwell's equations in 3D. I will begin with some background material on boundary integral equations and discretization issues, then move on to the main algorithm and its implementation. I will present numerical results on some benchmark problems at the end of the section. Chapter 3 will go over some of the work that has been explored in random surface scattering. I will briefly review boundary integral equations for acoustics, then continue with the stochastic methods which have been employed for these problems.

Chapter 2

Directional FMM for Maxwell's Equations

2.1 Boundary Integral Equations for Electromagnetics

In electromagnetics, the integral equations can be obtained in various ways. First, radiation formulas due to a current distribution $\mathbf{J}$ must be defined for the fields $\mathbf{E}$ and $\mathbf{H}$. Consider Maxwell's equations (1.9) in free space, where ε_r and μ_r are both the identity tensor. Taking the curl of (1.4) and substituting into the right hand side of (1.5) for $\nabla \times \mathbf{H}$ results in

$$\nabla \times \nabla \times \mathbf{E} - \kappa^2 \mathbf{E} = -\imath\omega\mu_0 \mathbf{J}. \tag{2.1}$$

Using the vector identity $\nabla \times \nabla \times \mathbf{E} = \nabla(\nabla \cdot \mathbf{E}) - \nabla^2 \mathbf{E}$ and continuity equation $\nabla \cdot \mathbf{J} = -\imath\omega\rho$, one arrives at the vector Helmholtz equation

$$\nabla^2 \mathbf{E} + \kappa^2 \mathbf{E} = \imath\omega\mu_0 \mathbf{J} - \frac{1}{\imath\omega\varepsilon_0} \nabla(\nabla \cdot \mathbf{J}) \tag{2.2}$$

The Green's function of the scalar Helmholtz equation can now be used in a convolution with the right hand side of (2.2) to define an integral operator acting on $\mathbf{J}$. The radiation formula for the electric field is then

$$\mathbf{E}(\mathbf{r}) = -\imath\omega\mu_0 \int_V G(\mathbf{r}, \mathbf{r}') \left[\mathbf{J}(\mathbf{r}') + \frac{1}{\kappa^2} \nabla' \nabla' \cdot \mathbf{J}(\mathbf{r}') \right] d\mathbf{r}'. \tag{2.3}$$

Following a similar procedure for the magnetic field, one can derive the integral

$$\mathbf{H}(\mathbf{r}) = \nabla \times \int_V G(\mathbf{r}, \mathbf{r}') \mathbf{J}(\mathbf{r}') d\mathbf{r}'. \tag{2.4}$$

Alternatively, one can form these radiation integrals through vector and scalar potentials. By using the Helmholtz decomposition theorem, it is assumed that **E** and **H** take the form

$$\begin{aligned} \mathbf{E} &= -\imath\omega\mathbf{A} - \nabla\Phi_e & (2.5) \\ \mathbf{H} &= \frac{1}{\mu_0}\nabla\times\mathbf{A}. & (2.6) \end{aligned}$$

for some vector potential **A** and scalar potential Φ_e. Choosing the Lorentz gauge

$$\nabla\cdot\mathbf{A} = -\imath\omega\mu_0\varepsilon_0\Phi_e, \tag{2.7}$$

one can obtain the inhomogeneous Helmholtz equations for **A** and Φ_e,

$$\begin{aligned} \nabla^2\mathbf{A} + \kappa^2\mathbf{A} &= -\mu_0\mathbf{J} & (2.8) \\ \nabla^2\Phi_e + \kappa^2\Phi_e &= -\frac{q_e}{\varepsilon_0}. & (2.9) \end{aligned}$$

Using the Helmholtz Green's function again, the integrals

$$\begin{aligned} \mathbf{A}(\mathbf{r}) &= \mu_0\int_V G(\mathbf{r},\mathbf{r}')\mathbf{J}(\mathbf{r}')d\mathbf{r}' & (2.10) \\ \Phi_e &= \frac{1}{\varepsilon_0}\int_V G(\mathbf{r},\mathbf{r}')q_e(\mathbf{r}')d\mathbf{r}' & (2.11) \end{aligned}$$

give the corresponding radiation formulas; the only difference is that the gradient operator in (2.5) is outside the integral acting on unprimed coordinates. The equivalent formula can be obtained by using the symmetry of the Green's function and integration by parts, redistributing the operator onto the current **J**. The details of this derivation are contained in [42].

To form the boundary integral equation, one must use equivalence principles to define "equivalent sources" which, when radiated using the integral formulas derived previously, will produce the same scattered fields as the original problem. Huygen's principle, also known as the surface equivalence theorem [21], states that every point on an advancing wavefront is a source of radiated waves. For a perfectly conducting object, it is postulated that the scattered field in the exterior of the object is generated by an induced current on the surface. Enforcing the tangential boundary conditions for the PEC along with the incident field-scattered field decompositions $\mathbf{E} = \mathbf{E}_{\text{inc}} + \mathbf{E}_{\text{sca}}$ and $\mathbf{H} = \mathbf{H}_{\text{inc}} + \mathbf{H}_{\text{sca}}$ gives

$$\hat{\mathbf{n}} \times (\mathbf{E}_{\text{inc}} + \mathbf{E}_{\text{sca}}) = 0 \qquad (2.12)$$

$$\hat{\mathbf{n}} \times (\mathbf{H}_{\text{inc}} + \mathbf{H}_{\text{sca}}) = \mathbf{J}_s, \qquad (2.13)$$

where $\mathbf{J}_s$ is the surface current. For a scatterer which occupies the region D with the boundary surface ∂D, the scattered fields $(\mathbf{E}_{\text{sca}}, \mathbf{H}_{\text{sca}})$ generated at location $\mathbf{r}$ by $\mathbf{J}_s$ are

$$\mathbf{E}_{\text{sca}}(\mathbf{r}) = -\imath\omega\mu_0 \int_{\partial D} G(\mathbf{r}, \mathbf{r}') \left[\mathbf{J}_s(\mathbf{r}') + \frac{1}{\kappa^2} \nabla' \nabla' \cdot \mathbf{J}_s(\mathbf{r}') \right] d\mathbf{r}' \qquad (2.14)$$

$$\mathbf{H}_{\text{sca}}(\mathbf{r}) = \nabla \times \int_{\partial D} G(\mathbf{r}, \mathbf{r}') \mathbf{J}_s(\mathbf{r}') d\mathbf{r}'. \qquad (2.15)$$

By crossing with the unit normal vector $\hat{\mathbf{n}}$, taking the limit as $\mathbf{r}$ approaches ∂D, and substituting the scattered fields with the incident fields in (2.13), the resulting integral equations are

$$\begin{aligned}\hat{\mathbf{n}}(\mathbf{r}) \times \mathbf{E}_{\text{inc}}(\mathbf{r}) &= \imath\omega\mu_0 \hat{\mathbf{n}}(\mathbf{r}) \times \int_{\partial D} G(\mathbf{r}, \mathbf{r}') \left[\mathbf{J}_s(\mathbf{r}') + \frac{1}{\kappa^2} \nabla' \nabla' \cdot \mathbf{J}_s(\mathbf{r}') \right] d\mathbf{r}' \\ \hat{\mathbf{n}}(\mathbf{r}) \times \mathbf{H}_{\text{inc}}(\mathbf{r}) &= \frac{1}{2} \mathbf{J}_s(\mathbf{r}) - \hat{\mathbf{n}}(\mathbf{r}) \times \int_{\partial D} \mathbf{J}_s(\mathbf{r}') \times \nabla' G(\mathbf{r}, \mathbf{r}') d\mathbf{r}', \qquad (2.16)\end{aligned}$$

where the curl operator has been distributed inside the integral in the second equation; the $\frac{1}{2}\mathbf{J}_s$ term is a result of the hypersingularity in the gradient of the Green's function. Since the incident fields $(\mathbf{E}_{\text{inc}}, \mathbf{H}_{\text{inc}})$ are known, one can solve for the current $\mathbf{J}_s$, then compute the scattered fields at any point $\mathbf{r}$ using the radiation formulas. Equations (2.16) are known as the electric field integral equation (EFIE) and magnetic field integral equation (MFIE), respectively.

For closed surface scattering problems, both the EFIE and MFIE suffer from spurious solutions due to internal resonant modes; at these frequencies, there exists a solution to the interior cavity problem. To eliminate these solutions, a common formulation is the combined field integral equation (CFIE). Because the null spaces of the EFIE and MFIE operators differ, taking a linear combination of the two ensures that there is a unique solution for each frequency. For a constant α between 0 and 1, the CFIE in electromagnetics is

$$\hat{\mathbf{n}}(\mathbf{r}) \times \left[\alpha \mathbf{E}_{\text{inc}}(\mathbf{r}) + (1-\alpha) \mathbf{H}_{\text{inc}}(\mathbf{r}) \right] = \alpha \mathcal{L}_{EFIE}[\mathbf{J}_s](\mathbf{r}) + (1-\alpha) \mathcal{L}_{MFIE}[\mathbf{J}_s](\mathbf{r}),$$

where $\mathcal{L}_{EFIE}[\mathbf{J}_s](\mathbf{r})$ and $\mathcal{L}_{MFIE}[\mathbf{J}_s](\mathbf{r})$ are the operators

$$\begin{aligned}\mathcal{L}_{EFIE}[\mathbf{J}_s](\mathbf{r}) &= \imath\omega\mu_0 \hat{\mathbf{n}}(\mathbf{r}) \times \int_{\partial D} G(\mathbf{r}, \mathbf{r}') \left[\mathbf{J}_s(\mathbf{r}') + \frac{1}{\kappa^2} \nabla' \nabla' \cdot \mathbf{J}_s(\mathbf{r}') \right] d\mathbf{r}' \\ \mathcal{L}_{MFIE}[\mathbf{J}_s](\mathbf{r}) &= \frac{1}{2} \mathbf{J}_s(\mathbf{r}) - \hat{\mathbf{n}}(\mathbf{r}) \times \int_{\partial D} \mathbf{J}_s(\mathbf{r}') \times \nabla' G(\mathbf{r}, \mathbf{r}') d\mathbf{r}'.\end{aligned}$$

In many scattering applications, such as radar or sonar, the quantity of interest is the far field pattern; this quantity describes the visibility of an object to an observer from different angles. Once the boundary integral equation is solved, the scattered field can be computed; if the field is being observed at a location $\mathbf{r}$ such that $|\mathbf{r}| \gg \lambda$, then the leading order approximation of the radiation integral can be taken. In electromagnetics, the far field pattern of the electric field is

$$\mathbf{E}_{\text{sca}}(\mathbf{r}) \approx -\imath\omega\mu_0 \frac{e^{-\imath\kappa|\mathbf{r}|}}{4\pi|\mathbf{r}|} \int_{\partial D} \mathbf{J}(\mathbf{r}') e^{\imath\kappa\hat{\mathbf{r}}\cdot\mathbf{r}'} d\mathbf{r}', \tag{2.17}$$

where $\hat{\mathbf{r}} = \frac{\mathbf{r}}{|\mathbf{r}|}$. The radar cross section (RCS) in the direction $\hat{\mathbf{r}}$ is defined by

$$\sigma(\hat{\mathbf{r}}) = \lim_{\rho\to\infty} 4\pi\rho^2 \frac{|\mathbf{E}_{\text{sca}}(\rho\hat{\mathbf{r}})|^2}{|\mathbf{E}_{\text{inc}}(\rho\hat{\mathbf{r}})|^2} = \frac{(\omega\mu)^2}{4\pi} \left| \int_{\partial D} \mathbf{J}(\mathbf{r}') e^{\imath\kappa\hat{\mathbf{r}}\cdot\mathbf{r}'} d\mathbf{r}' \right|^2 \tag{2.18}$$

2.2 Method of Moments

For electromagnetics, using a triangular mesh to represent the surface of the scatterer and expanding the current in terms of local basis functions such as the popular Rao-Wilton-Glisson (RWG) elements [80] is most common. This technique is known as the “method of moments” [53]. Consider the RWG basis function $\mathbf{f}_n$ defined on edge n of the triangular mesh of ∂D; an illustration of the RWG function is in figure 2.1. The support of each $\mathbf{f}_n$ consists of a pair of triangles $T_n^+ \cup T_n^- = \partial D_n$ sharing edge n. These functions satisfy

$$\mathbf{f}_n(\mathbf{r}) = \begin{cases} \frac{L_n}{2A_n^+}\boldsymbol{\rho}_n^+(\mathbf{r}) & \mathbf{r} \in T_n^+ \\ \frac{L_n}{2A_n^-}\boldsymbol{\rho}_n^-(\mathbf{r}) & \mathbf{r} \in T_n^- \\ 0 & \text{otherwise} \end{cases}, \tag{2.19}$$

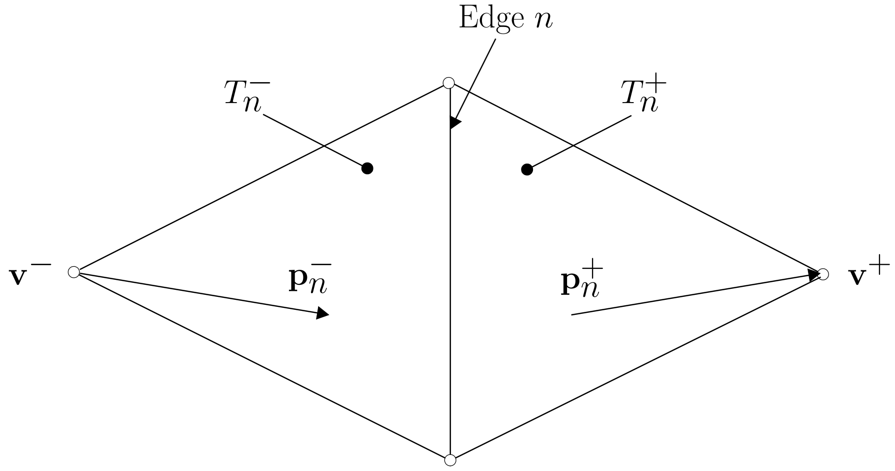


Figure 2.1: An RWG basis function defined on edge n.

where L_n is the length of edge n, $A_n^\pm$ is the area of $T_n^\pm$, and the vectors $\boldsymbol{\rho}_n^\pm$ are defined by

$$\begin{aligned} \boldsymbol{\rho}_n^+(\mathbf{r}) &= \mathbf{v}^+ - \mathbf{r} \qquad \mathbf{r} \in T_n^+ && (2.20) \\ \boldsymbol{\rho}_n^-(\mathbf{r}) &= \mathbf{r} - \mathbf{v}^- \qquad \mathbf{r} \in T_n^-. && (2.21) \end{aligned}$$

Here, $\mathbf{v}^\pm$ are the vertices of $T_n^\pm$ which are not vertices of edge n. Taking the surface divergence of the function yields

$$\nabla \cdot \mathbf{f}_n(\mathbf{r}) = \begin{cases} -\frac{L_n}{A_n^+} & \mathbf{r} \in T_n^+ \\ \frac{L_n}{A_n^-} & \mathbf{r} \in T_n^- \\ 0 & \text{otherwise} \end{cases} . \qquad (2.22)$$

This computation illustrates that the RWG is divergence conforming; the result is normal continuity of the current over triangle edges.

For a triangular surface mesh with N interior edges, the current $\mathbf{J}(\mathbf{r})$

can be expanded in terms of N basis functions,

$$\mathbf{J}(\mathbf{r}) = \sum_{n=1}^{N} c_n \mathbf{f}_n(\mathbf{r}), \tag{2.23}$$

where $\{c_n\}_{n=1}^{N} \in \mathbb{C}$ are the unknown coefficients. Plugging this expansion into the CFIE gives a semi-discrete version of the integral equation; at this stage, a test space can be chosen. Popular choices are the Galerkin method or pointwise collocation. For this work, I chose the "razor-blade" testing functions for the EFIE and point-matching for the MFIE because of their simplicity and ease of implementation; the details of these testing functions can be found in [75]. With this formulation, the $N \times N$ linear system of equations obtained is

$$ZI = V, \tag{2.24}$$

where the entries of Z, I, and V are

$$Z_{mn} = \alpha A_{mn} + (1-\alpha)\eta \Delta t_m B_{mn}, \tag{2.25}$$

$$A_{mn} = \imath\kappa\eta \int_{C_m} \int_{\partial D_n} G(\mathbf{r},\mathbf{r}')\mathbf{f}_n(\mathbf{r}')d\mathbf{r}' \cdot d\mathbf{r} - \tag{2.26}$$

$$\frac{\eta}{\imath\kappa} \int_{C_m} \nabla \left[\int_{\partial D_n} G(\mathbf{r},\mathbf{r}')\nabla' \cdot \mathbf{f}_n(\mathbf{r}')d\mathbf{r}' \right] \cdot d\mathbf{r}, \tag{2.27}$$

$$B_{mn} = \hat{\mathbf{e}}_m \cdot \int_{\partial D_n} \mathbf{f}_n(\mathbf{r}') \times \nabla G(\mathbf{r},\mathbf{r}')d\mathbf{r}' \Big|_{\mathbf{r}=\mathbf{r}_m}, \qquad m \neq n \tag{2.28}$$

$$B_{mm} = \frac{2\pi - \Omega_m}{2\pi} \tag{2.29}$$

$$I_n = c_n, \tag{2.30}$$

$$V_m = \alpha \int_{C_m} \mathbf{E}_{\text{inc}}(\mathbf{r}) \cdot d\mathbf{r} + (1-\alpha)\eta\Delta t_m \hat{\mathbf{e}}_m \cdot \mathbf{H}_{\text{inc}}(\mathbf{r}_m). \tag{2.31}$$

Here, C_m is the path of the razor-blade function along the surface of the m-th RWG, $\hat{\mathbf{e}}_m$ is the unit vector in the direction of the adjoining edge of the m-th

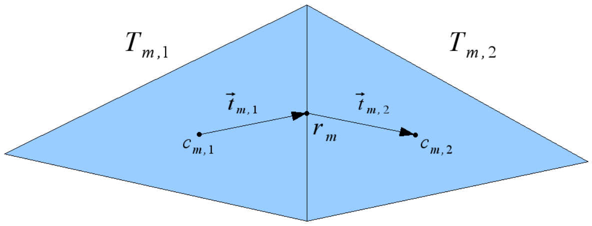


Figure 2.2: Illustration of a razor-blade function defined on the m-th edge.

RWG (oriented so that $\hat{\mathbf{n}} \times \hat{\mathbf{e}}_m$ points in the direction of the basis function, for either triangle), $\mathbf{r}_m$ is the center of the same adjoining edge, and Ω_m is the interior angle created by the two triangles $(T_{m,1}, T_{m,2})$ (see Figure 2.2 for an illustration of these notations). For the outer integrals in the matrix entries of A, the formulas

$$\int_{C_m} \nabla\phi(\mathbf{r})\cdot d\mathbf{r} = \phi(\mathbf{c}_{m,2}) - \phi(\mathbf{c}_{m,1}), \qquad \int_{C_m} \mathbf{F}(\mathbf{r})\cdot d\mathbf{r} \approx \mathbf{F}(\mathbf{c}_{m,1})\cdot\mathbf{t}_{m,1} + \mathbf{F}(\mathbf{c}_{m,2})\cdot\mathbf{t}_{m,2}$$

are used, where $\mathbf{c}_{m,1}$ and $\mathbf{c}_{m,2}$ are the centroids of $T_{m,1}$ and $T_{m,2}$, respectively, and $\mathbf{t}_{m,1}$, $\mathbf{t}_{m,2}$ are the vectors prescribed by the razor-blade function at the centroids. If $\phi(\mathbf{r}) = \int_{\partial D_n} G(\mathbf{r},\mathbf{r}')\nabla'\cdot\mathbf{f}_n(\mathbf{r}')d\mathbf{r}'$ and $\mathbf{F}(\mathbf{r}) = \int_{\partial D_n} G(\mathbf{r},\mathbf{r}')\mathbf{f}_n(\mathbf{r}')d\mathbf{r}'$, the matrix entries of A now become

$$\begin{aligned} A_{mn} &= \imath\kappa\eta\biggl(\mathbf{t}_{m,1}\cdot\int_{\partial D_n} G(\mathbf{c}_{m,1},\mathbf{r}')\mathbf{f}_n(\mathbf{r}')d\mathbf{r}' + \mathbf{t}_{m,2}\cdot\int_{\partial D_n} G(\mathbf{c}_{m,2},\mathbf{r}')\mathbf{f}_n(\mathbf{r}')d\mathbf{r}'\biggr) \\ &\quad -\frac{\eta}{\imath\kappa}\biggl(\int_{\partial D_n} G(\mathbf{c}_{m,2},\mathbf{r}')\nabla'\cdot\mathbf{f}_n(\mathbf{r}')d\mathbf{r}' - \int_{\partial D_n} G(\mathbf{c}_{m,1},\mathbf{r}')\nabla'\cdot\mathbf{f}_n(\mathbf{r}')d\mathbf{r}'\biggr). \end{aligned}$$

2.3 Directional FMM

The matrix-vector product performed at each iteration takes the form of the N-body problem,

$$u_i = \sum_{\substack{j=1\\ i\neq j}}^{N} G(x_i, x_j) f_j \tag{2.32}$$

for the point set $\{x_i\}_{i=1}^N \subset \mathbb{R}^3$, where $f_j \in \mathbb{C}$ is the source located at point x_j and u_i is the potential computed at x_i. Very recently, Engquist and Ying proposed an algorithm [32, 33] which computes this summation for each x_i in $O(N \log N)$ time; this algorithm has been named as the fast directional multilevel algorithm, or directional fast multipole method, due to its structural similarity to the FMM.

The main idea behind the directional FMM is the directional low rank property of the Green's function. That is, consider a box B of width $w\lambda$ with $w \geq 1$ and a wedge $W^{B,\ell}$ as illustrated in Figure 2.3 (a); if $W^{B,\ell}$ spans an angle of size $O(1/w)$ and the distance between B and $W^{B,\ell}$ is at least $O(w^2\lambda)$, it is said that $W^{B,\ell}$ and B satisfy the *directional parabolic separation condition.* In [32], it was proven that for any arbitrary accuracy ε, there exists a t_ε-term separated approximation of $G(x, y)$ for any $y \in B$ and $x \in W^{B,\ell}$. In practice, one can find sets $\{y_q^{B,\ell}\}_{q=1}^{t_\varepsilon}$, $\{x_p^{B,\ell}\}_{p=1}^{t_\varepsilon}$ and a matrix $D^{B,\ell} = (d_{qp}^{B,\ell})_{1\leq p,q\leq t_\varepsilon}$ such that

$$\left| G(x, y) - \sum_{q=1}^{t_\varepsilon} G(x, y_q^{B,\ell}) \sum_{p=1}^{t_\varepsilon} d_{qp}^{B,\ell} G(x_p^{B,\ell}, y) \right| \leq \varepsilon, \tag{2.33}$$

for $y \in B$ and $x \in W^{B,\ell}$. This is called the *directional separated approximation* of B in direction ℓ. The locations $\{y_q^{B,\ell}\}$, $\{x_p^{B,\ell}\}$, and the matrix D can be

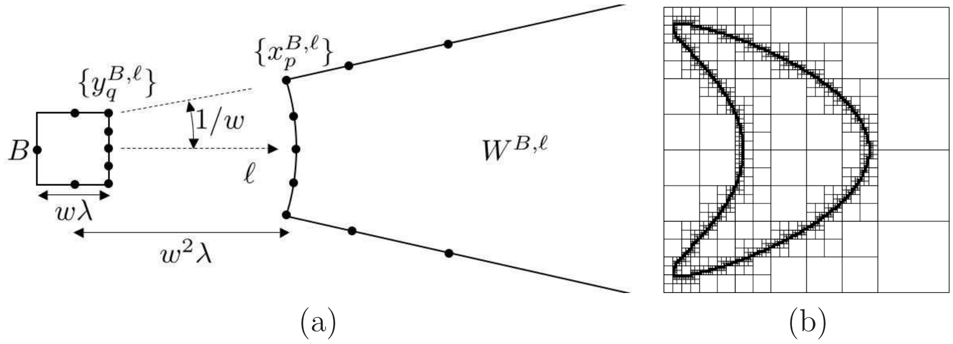


Figure 2.3: 2-D illustrations for components of the 3-D algorithm. (a) B and $W^{B,\ell}$ follow the directional parabolic separation condition. (b) A cross-section of the octree of a kite-shaped scatterer.

computed easily using the low-rank factorization scheme detailed in [33]; it is important to emphasize that the separation rank t_ε is independent of the size of B.

Consider the sources $\{f_i\}$ located at $\{z_i\}$ in B. The directional separated approximation allows the computation of the potentials in $W^{B,\ell}$ generated by $\{f_i\}$ with a smaller set of t_ε sources. More precisely, after applying (2.33) to each z_i and summing them up over all the sources, one has the estimate

$$\left| \sum_{z_i \in B} G(x, z_i) f_i - \sum_{q=1}^{t_\varepsilon} G(x, y_q^{B,\ell}) \left(\sum_{p=1}^{t_\varepsilon} d_{qp}^{B,\ell} \sum_{z_i \in B} G(x_p^{B,\ell}, z_i) f_i \right) \right| = O(\varepsilon). \quad (2.34)$$

This states that a set of sources

$$\left\{ f_q^{B,\ell} := \sum_{p=1}^{t_\varepsilon} d_{qp}^{B,\ell} \sum_{z_i \in B} G(x_p^{B,\ell}, z_i) f_i \right\} \quad (2.35)$$

can be placed at points $\{y_q^{B,\ell}\}$ in order to reproduce the potential at $x \in W^{B,\ell}$ generated by the sources $\{f_i\}$ located at points $\{z_i\}$ in B. These sources are called the *directional equivalent sources* of B in direction ℓ, and they play the role of the multipole expansions in the original FMM [46]. It is obvious from (2.34) that the computation of $\{f_q^{B,\ell}\}$ essentially requires only the potentials $\{\sum_{z_i \in B} G(x_p^{B,\ell}, z_i) f_i\}$ at $\{x_p^{B,\ell}\}$.

Because of the symmetry of the Green's function, the role of the source and target can be reversed to get a similar result for the analogous local expansions. Consider now the sources $\{f_i\}$ located at points $\{z_i\}$ in $W^{B,\ell}$. Applying (2.33) to each z_i and summing over all the sources gives

$$\left| \sum_{z_i \in W^{B,\ell}} G(y, z_i) f_i - \sum_{p=1}^{t_\varepsilon} G(y, x_p^{B,\ell}) \left(\sum_{q=1}^{t_\varepsilon} d_{qp}^{B,\ell} \sum_{z_i \in W^{B,\ell}} G(y_q^{B,\ell}, z_i) f_i \right) \right| = O(\varepsilon).$$

This means that from the potentials $\{u_q^{B,\ell} := \sum_{z_i \in W^{B,\ell}} G(y_q^{B,\ell}, z_i) f_i\}$, one can reproduce the potential at any $y \in B$ generated by $\{f_i\}$ at $\{z_i\} \subset W^{B,\ell}$. These potentials are called the *directional check potentials* of B in direction ℓ; they play the role of the local expansions in the original FMM.

For a box B of width $w\lambda$ with $w < 1$, the wedges are replaced with a single annulus W^B with an outer radius that extends to infinity. The equivalent sources, the check potentials, the point sets $\{x_p^B\}$ and $\{y_q^B\}$, and the transform matrices can be defined similarly, but they are non-directional now (see [101] for details). For example, the equivalent sources for B can be computed by

$$\left\{ f_q^B := \sum_{p=1}^{t_\varepsilon} d_{qp}^B \sum_{z_i \in B} G(x_p^B, z_i) f_i \right\}, \tag{2.36}$$

and it is clear that the computation again requires only the potentials at x^B, which are $\sum_{z_i \in B} G(x_p^B, z_i) f_i$.

The directional multilevel algorithm begins by constructing an octree that contains the entire scatterer (see Figure 2.3(b)). Similarly to [22], the tree is split into two regimes: the low-frequency regime and high-frequency regime. A box B of width $w\lambda$ is considered to be in the low frequency regime if $w < 1$ and in the high-frequency regime if $w \geq 1$. In the low-frequency regime, a box B is partitioned as long as the number of points in B is greater than a fixed constant P; in the high-frequency regime, the domain is partitioned uniformly without any adaptivity, with empty boxes being discarded. The algorithm now uses the translation operators introduced in [101] in the low-frequency regime. In order to use the directional separated approximation in (2.33), the *far field* F^B of a box B is defined as the region that is at least $w^2\lambda$ away in the high-frequency regime; conversely, the *near field* N^B is defined as the union of boxes which are less than $w^2\lambda$ away. A box A is said to be in the *interaction list* of B if A is in B's far field but not in the far field of B's parent. F^B is further partitioned into a group of directional wedges $\{W^{B,\ell}\}$, where each wedge is contained in a cone with spanning angle $O(1/w)$.

A crucial point is that the wedges of the parent box and the child box are *nested*, so that the construction of M2M, M2L, and L2L translations are of $O(1)$ complexity as in the original FMM algorithm [46]. However, these translations in the high frequency regime are now *directional.* Combining these parts gives the following *high-frequency directional multilevel algorithm.*

1. Construct the octree. In the high-frequency regime, the boxes are partitioned uniformly. In the low-frequency regime, the boxes are partitioned

adaptively until each leaf level box contains at most P points.

2. Travel up the low-frequency regime. For each box B, compute the non-directional equivalent sources following [101].

3. Travel up the high-frequency regime. For each box B and each $W^{B,\ell}$, compute $\{f_q^{B,\ell}\}$ using the directional M2M translation. The boxes with width greater than $\sqrt{K}\lambda$ are skipped since their interaction lists are empty.

4. Travel down the high-frequency regime. For each box B and each $W^{B,\ell}$
   (a) Transform $\{f_q^{A,\ell}\}$ of the boxes $\{A\}$ in B's interaction list and in direction ℓ using the directional M2L translation. Next, add the result to $\{u_q^{B,\ell}\}$.
   (b) Transform $\{u_q^{B,\ell}\}$ into the directional check potentials for B's children using the directional L2L translation.

5. Travel down the low-frequency regime. For each box B:
   (a) Transform the non-directional equivalent sources of the boxes $\{A\}$ in B's interaction list using the non-directional M2L translation. Next, add the result to the non-directional check potentials.
   (b) Perform the non-directional L2L translation. If B is a leaf box, add the result to the potentials at the original points inside B. If not, then add the result to the non-directional check potentials of B's children.

6. Compute the near-field interactions. For each leaf box B, add contributions from sources in N^B directly to the potentials at points in B.

For a point set $\{z_i\}_{i=1}^N$ obtained from discretizing the surface of a scatterer, it is shown in [32, 33] that the overall cost of this algorithm is $O(N \log N)$.

2.4 DFMM for Maxwell's Equations

To solve the linear system (2.24) with GMRES, at each iteration the updated vector of coefficients I is given and a summation of the form

$$\sum_{n=1}^{N} Z_{mn} I_n \tag{2.37}$$

for $m = 1, 2, ..., N$ must be computed. If the matrix entries are done explicitly and each summation is performed directly, this obviously yields a complexity of $O(N^2)$, with an $O(N^2)$ memory requirement. Instead, the approach is modified by considering the evaluation of the potential integrals at the test points first, before applying the test vectors. That is, define the following integral operators

$$\begin{aligned} U[\mathbf{J}](\mathbf{r}) &= \sum_{i=1}^{N_t} \int_{T_i} G(\mathbf{r}, \mathbf{r}')\mathbf{J}(\mathbf{r}')d\mathbf{r}', && (2.38) \\ V[\mathbf{J}](\mathbf{r}) &= \sum_{i=1}^{N_t} \int_{T_i} G(\mathbf{r}, \mathbf{r}')\nabla' \cdot \mathbf{J}(\mathbf{r}')d\mathbf{r}', && (2.39) \\ W[\mathbf{J}](\mathbf{r}) &= \sum_{i=1}^{N_t} \int_{T_i} \mathbf{J}(\mathbf{r}') \times \nabla G(\mathbf{r}, \mathbf{r}')d\mathbf{r}', && (2.40) \end{aligned}$$

where the integrals are now over each triangle instead of each RWG; N_t is the number of triangles in the mesh and T_i is the i-th triangle. Using these operators and (2.31), the summation in (2.37) is simply

$$\begin{aligned}\sum_{n=1}^{N} Z_{mn} I_n &= \alpha\Big\{\imath\kappa\eta\Big(\mathbf{t}_{m,1}\cdot U[\mathbf{J}](\mathbf{c}_{m,1}) + \mathbf{t}_{m,2}\cdot U[\mathbf{J}](\mathbf{c}_{m,2})\Big)\\ &\quad -\frac{\eta}{\imath\kappa}\Big(V[\mathbf{J}](\mathbf{c}_{m,2}) - V[\mathbf{J}](\mathbf{c}_{m,1})\Big)\Big\}\\ &\quad +(1-\alpha)\eta\Delta t_m\Big\{\hat{\mathbf{e}}_m\cdot W[\mathbf{J}](\mathbf{r}_m) + \frac{2\pi-\Omega_m}{2\pi} j_m\Big\}. \qquad (2.41)\end{aligned}$$

At each iteration, the expansion (2.23) can be used to interpolate the current on every triangle given the input $\{j_n\}_{n=1}^{N}$. It can be observed from the matrix entries that the necessary task is to evaluate $U[\mathbf{J}](\mathbf{r})$ and $V[\mathbf{J}](\mathbf{r})$ at points $\{\mathbf{c}_\ell\}_{\ell=1}^{N_t}$ and $W[\mathbf{J}](\mathbf{r})$ at points $\{\mathbf{r}_m\}_{m=1}^{N}$. Since these integrals cannot be analytically evaluated in most cases, it is necessary to introduce a numerical quadrature scheme. First, consider the evaluation of $U[\mathbf{J}](\mathbf{c}_\ell)$ and $V[\mathbf{J}](\mathbf{c}_\ell)$. If $\mathbf{c}_\ell \notin T_i$, then symmetric Gaussian quadrature in [91] can be used over T_i; if $\mathbf{c}_\ell \in T_i$, however, a quadrature rule especially constructed to handle the $\frac{1}{r}$ singularity in the integral is employed. To handle this, the triangle T_i is subdivided into three new triangles which share $\mathbf{c}_\ell$ as a vertex, and the Duffy quadrature rule proposed in [29] is used over each of the new triangles (see Figure 2.4).

For a triangle T_i, let $\{\mathbf{p}_{i,q}\}_{q=1}^{Q}$ and $\{\alpha_{i,q}\}_{q=1}^{Q}$ be the nodes and weights of the Gaussian quadrature rule and let $\{\mathbf{t}_{i,s}\}_{s=1}^{S}$ and $\{\beta_{i,s}\}_{s=1}^{S}$ be the unions of the three sets of Duffy quadrature nodes and weights (one for each of the three new triangles). It is important to note here that the weights take into account the area when integrating over T_i, i.e. that the Jacobian is already built into

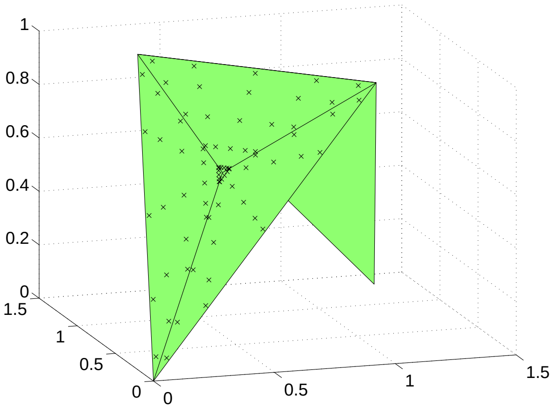


Figure 2.4: Integrating over an RWG basis function with a singularity at the centroid of one triangle. The triangle is subdivided into 3 new triangles. The ×'s mark the location of the Duffy quadrature nodes.

each set of weights. The integrals for each $\mathbf{c}_\ell$ can then be approximated as

$$\begin{aligned} U[\mathbf{J}](\mathbf{c}_\ell) &\approx \sum_{\substack{i=1 \\ i\neq\ell}}^{N_t} \sum_{q=1}^{Q} G(\mathbf{c}_\ell, \mathbf{p}_{i,q}) \mathbf{J}(\mathbf{p}_{i,q}) \alpha_{i,q} + \sum_{s=1}^{S} G(\mathbf{c}_\ell, \mathbf{t}_{\ell,s}) \mathbf{J}(\mathbf{t}_{\ell,s}) \beta_{\ell,s}, \\ V[\mathbf{J}](\mathbf{c}_\ell) &\approx \sum_{\substack{i=1 \\ i\neq\ell}}^{N_t} \sum_{q=1}^{Q} G(\mathbf{c}_\ell, \mathbf{p}_{i,q}) \nabla\cdot\mathbf{J}(\mathbf{p}_{i,q}) \alpha_{i,q} + \sum_{s=1}^{S} G(\mathbf{c}_\ell, \mathbf{t}_{\ell,s}) \nabla\cdot\mathbf{J}(\mathbf{t}_{\ell,s}) \beta_{\ell,s}. \end{aligned}$$

The main computation is the first double sum in each formula, since for each $\mathbf{c}_\ell$ one needs to sum over $O(qN_t)$ terms. It is not easy to apply fast summation algorithms directly to this sum, since the summation is performed over a set that depends on $\mathbf{c}_\ell$, i.e., the constraint $i \neq \ell$. In order to facilitate the fast summation, formulas are rewritten as

$$U[\mathbf{J}](\mathbf{c}_\ell) \approx \sum_{i=1}^{N_t} \sum_{q:\mathbf{c}_\ell \neq \mathbf{p}_{i,q}} G(\mathbf{c}_\ell, \mathbf{p}_{i,q})\mathbf{J}(\mathbf{p}_{i,q})\alpha_{i,q} -$$

$$\sum_{q:\mathbf{c}_\ell \neq \mathbf{p}_{\ell,q}} G(\mathbf{c}_\ell, \mathbf{p}_{\ell,q})\mathbf{J}(\mathbf{p}_{\ell,q})\alpha_{\ell,q} + \sum_{s=1}^{S} G(\mathbf{c}_\ell, \mathbf{t}_{\ell,s})\mathbf{J}(\mathbf{t}_{\ell,s})\beta_{\ell,s}, \tag{2.42}$$

$$V[\mathbf{J}](\mathbf{c}_\ell) \approx \sum_{i=1}^{N_t} \sum_{q:\mathbf{c}_\ell \neq \mathbf{p}_{i,q}} G(\mathbf{c}_\ell, \mathbf{p}_{i,q})\nabla \cdot \mathbf{J}(\mathbf{p}_{i,q})\alpha_{i,q} -$$

$$\sum_{q:\mathbf{c}_\ell \neq \mathbf{p}_{\ell,q}} G(\mathbf{c}_\ell, \mathbf{p}_{\ell,q})\nabla \cdot \mathbf{J}(\mathbf{p}_{\ell,q})\alpha_{\ell,q} + \sum_{s=1}^{S} G(\mathbf{c}_\ell, \mathbf{t}_{\ell,s})\nabla \cdot \mathbf{J}(\mathbf{t}_{\ell,s})\beta_{\ell,s}. \tag{2.43}$$

The advantage of this form is that now the summation is essentially over all possible pairs (i, q), except the case $\mathbf{c}_\ell = \mathbf{p}_{i,q}$, which can be handled easily by fast algorithms. For the evaluation of $W[\mathbf{J}](\mathbf{r}_m)$, Gaussian quadrature nodes are again used when $\mathbf{r}_m \notin T_i$. For $\mathbf{r}_m \in T_i$, since $\nabla G(\mathbf{r}_m, \mathbf{r})$ lies in the plane of T_i, $\mathbf{f}_n(\mathbf{r}) \times \nabla G(\mathbf{r}_m, \mathbf{r})$ is perpendicular to T_i; thus, when dotted with $\hat{e}_m$, the contribution from T_i is zero. There is no need for singularity correction quadrature, so the sum is just

$$W[\mathbf{J}](\mathbf{r}_m) \approx \sum_{i=1}^{N_t} \sum_{q} \alpha_{i,q}\mathbf{J}(\mathbf{p}_{i,q}) \times \nabla G(\mathbf{r}_m, \mathbf{p}_{i,q}). \tag{2.44}$$

It is clear that the double sums for U, V, and W result in $O(N^2)$ complexity, and their computation will be accelerated using the directional FMM.

It is first observed that in (2.32), there is only one set of points that serve both as "sources" and "targets," while in (2.42), (2.43), and (2.44), the source points and target points are different. To address this difference, the source points and target points are combined into a single set of points. If the target points are assigned zero weights, then the FMM can be applied to the combined set of points and the potentials at the target points can be extracted. Clearly, this does not change the complexity of the algorithm; from now on, it is safely assumed that the source and target points can be different. If $\mathbf{u}_\ell$ denotes the double sum in (2.42), $j(i,q) = (i-1)Q + q$, $N_y = N_t Q$, $\mathbf{y}_j = \mathbf{p}_{i,q}$, and $\mathbf{f}_j = \alpha_{i,q}\mathbf{J}(\mathbf{p}_{i,q})$, then the double sum for U can be rewritten as

$$\mathbf{u}_\ell = \sum_{\substack{j=1 \\ \mathbf{c}_\ell \neq \mathbf{y}_j}}^{N_y} G(\mathbf{c}_\ell, \mathbf{y}_j)\mathbf{f}_j, \tag{2.45}$$

where $\mathbf{u}_\ell = (u_{\ell,1}, u_{\ell,2}, u_{\ell,3})$ and $\mathbf{f}_j = (f_{j,1}, f_{j,2}, f_{j,3})$. Similarly, if v_ℓ is the double sum of (2.43) and $g_j = \alpha_{i,q}\nabla \cdot \mathbf{J}(\mathbf{p}_{i,q})$, the double sum for V is of the form

$$v_\ell = \sum_{\substack{j=1 \\ \mathbf{c}_\ell \neq \mathbf{y}_j}}^{N_y} G(\mathbf{c}_\ell, \mathbf{y}_j) g_j. \tag{2.46}$$

Thus, these are the forms in which one is able to apply the directional multilevel algorithm. The summation for v_ℓ is in the exact same form as (2.32), i.e. the product of an $N_t \times N_y$ matrix and an $N_y \times 1$ vector. The summation for $\mathbf{u}_\ell$ is also of the same form, only now there are three components to sum over; instead of a matrix-vector operation, it is a multiplication of an $N_t \times N_y$ matrix with an $N_y \times 3$ matrix. In practice, since the discretization points are the same for both sums, the computations are aggregated by combining the three components from $\mathbf{f}_j$ with g_j. By defining $\boldsymbol{\alpha}_\ell = (u_{\ell,1}, u_{\ell,2}, u_{\ell,3}, v_\ell)$ and

$\boldsymbol{\beta}_j = (f_{j,1}, f_{j,2}, f_{j,3}, g_j)$, the sum is

$$\boldsymbol{\alpha}_\ell = \sum_{\substack{j=1 \\ \mathbf{c}_\ell \neq \mathbf{y}_j}}^{N_y} G(\mathbf{c}_\ell, \mathbf{y}_j)\boldsymbol{\beta}_j. \tag{2.47}$$

Here, instead of running the directional multilevel algorithm four times (once for each component), the algorithm is performed once but with all of the translation operators vectorized to handle multiple columns. This process is fairly simple, since the equivalent source and potential locations remain the same; the only difference is that the sources and potentials take on vector values. When computing the equivalent sources or any of the translation operators, matrix-matrix products are used instead of matrix-vector products. The resulting complexity of this computation is $O(N_y \log N_y)$ where $N_y = N_t Q$. Since Q is constant and the number of triangles N_t is always less than the number of unknowns, this complexity is $O(N \log N)$.

Using the same notation for j, $\mathbf{y}_j$, N_y, and $\mathbf{f}_j$, the double sum of (2.44) is of the form

$$\mathbf{w}_m = \sum_{j=1}^{N_y} \nabla G(\mathbf{r}_m, \mathbf{y}_j) \times \mathbf{f}_j, \tag{2.48}$$

where $\mathbf{w}_m = W[\mathbf{J}](\mathbf{r}_m) = (w_{m,1}, w_{m,2}, w_{m,3})$ and the gradient operator $\nabla = (\partial_x, \partial_y, \partial_z)$ acts on the first argument of G. Rewriting the cross product as a matrix-vector operation, it is clear that

$$\begin{pmatrix} w_{m,1} \\ w_{m,2} \\ w_{m,3} \end{pmatrix} = \sum_{j=1}^{N_y} \begin{pmatrix} 0 & -G_z(\mathbf{r}_m, \mathbf{y}_j) & G_y(\mathbf{r}_m, \mathbf{y}_j) \\ G_z(\mathbf{r}_m, \mathbf{y}_j) & 0 & -G_x(\mathbf{r}_m, \mathbf{y}_j) \\ -G_y(\mathbf{r}_m, \mathbf{y}_j) & G_x(\mathbf{r}_m, \mathbf{y}_j) & 0 \end{pmatrix} \begin{pmatrix} f_{j,1} \\ f_{j,2} \\ f_{j,3} \end{pmatrix}. \tag{2.49}$$

At first, it may seem as if the directional multilevel algorithm cannot be applied to this situation. If a minor modification is made, however, it will become apparent that the algorithm can work for (2.49). More specifically, consider the first component,

$$w_{m,1} = \sum_{j=1}^{N_y} \left(-G_z(\mathbf{r}_m, \mathbf{y}_j) f_{j,2} + G_y(\mathbf{r}_m, \mathbf{y}_j) f_{j,3} \right). \tag{2.50}$$

The essential step of the directional multilevel algorithm is the construction of the equivalent sources. For a leaf level box B in the low-frequency regime, (2.36) states that the potential field produced by an arbitrary set of sources inside B can be well approximated by a small set of equivalent sources in B, when observing the far field of B. As G_z and G_y are derivatives of the Green's function, the sources in (2.50) are essentially dipoles in the z and y directions. Since the field generated by a dipole can be well approximated by a group or a distribution of monopoles (i.e., sources determined by the Green's function) in its vicinity, the field generated by points in B with the kernels of (2.50) can also be approximated by a small set of equivalent sources, when observing the far field of B. Moreover, notice that in (2.36) the construction of the equivalent sources for a box B requires only the potentials at $\{x_p^B\}$; clearly, these potentials can be evaluated directly using the formulas of the kernels $G_z(\cdot, \mathbf{y}_j)$ and $G_y(\cdot, \mathbf{y}_j)$. Two points need to be emphasized here: first, the equivalent sources are always computed using the Green's function even though the kernels of (2.50) are the derivatives; second, once the true sources of (2.50) are transformed into equivalent sources at the leaf level, the computation in the high-frequency regime only involves the Green's function $G(x, y)$, and hence requires no modification at all. Compared with the algorithm for the

Green's function $G(x, y)$, the algorithm for the kernels of (2.50) requires only the following modifications:

- For the computation of the equivalent sources for the leaf boxes in the low-frequency regime, use the kernels of (2.50) to compute the potentials at $\{x_p^B\}$.
- Use the kernels of (2.50) for direct near-field calculations in the low-frequency regime.

The computation of the second and third components $w_{m,2}$ and $w_{m,3}$ can be handled in essentially the same way. In fact, since the algorithm for $w_{m,1}$, $w_{m,2}$, and $w_{m,3}$ above the leaf level is exactly the same, the computations can be aggregated by vectorizing the translation operations for all three components, allowing one to perform the necessary calculations by running the algorithm once. The resulting algorithm for the MFIE kernel has $O(N_y \log N_y) = O(N \log N)$ complexity.

2.5 Numerical Results for Maxwell DFMM

The first test of the directional multilevel algorithm is to show the scaling properties and attainable accuracy. The tests are performed on two commonly-used examples: the sphere and the NASA almond (see Figure 2.6). Here, the triangular meshes use roughly 5 elements per wavelength. In these tables, K is the diameter of the object in terms of wavelength, ε is the prescribed order of accuracy, T_e is the time of the EFIE summation, T_m is the time of the MFIE summation, ε_e is the relative error for the EFIE summation, and ε_m is the relative error for the MFIE summation. N_p is the total number of discretization points in the directional multilevel algorithm, including the Gaussian quadrature nodes over each triangle, the centroid of each triangle, and the center of each edge. To estimate the relative error between direct calculation and the algorithm, the true values of the summations are computed at a subset R of 200 points sampled from the total number of points. The relative error is computed via

$$\sqrt{\frac{\sum_{i\in R}|\mathbf{u}_i - \mathbf{u}_i^c|^2}{\sum_{i\in R}|\mathbf{u}_i|^2}} \tag{2.51}$$

where $\mathbf{u}_i$ is the value by direct computation and $\mathbf{u}_i^c$ is the value by the directional algorithm.

Table 2.1 shows data for the sphere, while table 2.2 shows data for the NASA almond. It is observed that the computational time grows roughly by a factor of 3 or 4 when increasing the accuracy by two digits. It is also noted that the algorithm for the MFIE summation is slightly less accurate; nevertheless, the same order of accuracy is retained. To illustrate the $O(N \log N)$ scaling of the algorithm, a plot of the computational time versus the number of unknowns for the sphere is in figure 2.5. Here, the size of the sphere ranges from $K = 3$ to $K = 48$, and the error tolerance is 1e-4.

For electromagnetic scattering examples, there are a few test geometries that have been well established in the electromagnetics community. In these tests, the residual tolerance for GMRES iteration is set at 1e-3, while the error

(K,ε)	N_p	T_e(sec)	T_m(sec)	ε_e	ε_m
(12,1e-4)	4.669e+5	1.080e+2	1.250e+2	3.771e-4	6.428e-4
(24,1e-4)	1.867e+6	4.830e+2	5.470e+2	4.027e-4	6.594e-4
(48,1e-4)	7.471e+6	2.150e+3	2.384e+3	3.760e-4	6.991e-4
(12,1e-6)	4.669e+5	4.140e+2	3.690e+2	1.500e-6	3.639e-6
(24,1e-6)	1.867e+6	1.790e+3	1.601e+3	1.646e-6	3.794e-6
(48,1e-6)	7.471e+6	7.665e+3	6.773e+3	2.227e-6	3.472e-6
(12,1e-8)	4.669e+5	1.217e+3	1.000e+3	8.648e-9	6.371e-8
(24,1e-8)	1.867e+6	5.148e+3	4.198e+3	1.637e-8	6.625e-8
(48,1e-8)	7.471e+6	2.165e+4	1.742e+4	1.349e-8	5.284e-8

Table 2.1: Computational times and relative error 2-norms for the directional algorithm with discretization points on the surface of a sphere.

(K,ε)	N_p	T_e(sec)	T_m(sec)	ε_e	ε_m
(32,1e-4)	3.164e+5	8.400e+1	8.800e+1	4.083e-4	5.291e-4
(64,1e-4)	1.265e+6	3.860e+2	3.860e+2	3.463e-4	5.042e-4
(128,1e-4)	5.059e+6	1.785e+3	1.745e+3	4.350e-4	4.824e-4
(32,1e-6)	3.164e+5	3.130e+2	2.700e+2	2.386e-6	2.860e-6
(64,1e-6)	1.265e+6	1.372e+3	1.171e+3	1.363e-6	2.814e-6
(128,1e-6)	5.059e+6	6.082e+3	5.177e+3	1.977e-6	2.579e-6
(32,1e-8)	3.164e+5	9.310e+2	7.460e+2	1.396e-8	3.300e-8
(64,1e-8)	1.265e+6	3.978e+3	3.226e+3	1.361e-8	3.139e-8
(128,1e-8)	5.059e+6	1.719e+4	1.363e+4	1.573e-8	3.099e-8

Table 2.2: Computational times and relative error 2-norms for the directional algorithm with discretization points on the surface of the NASA almond.

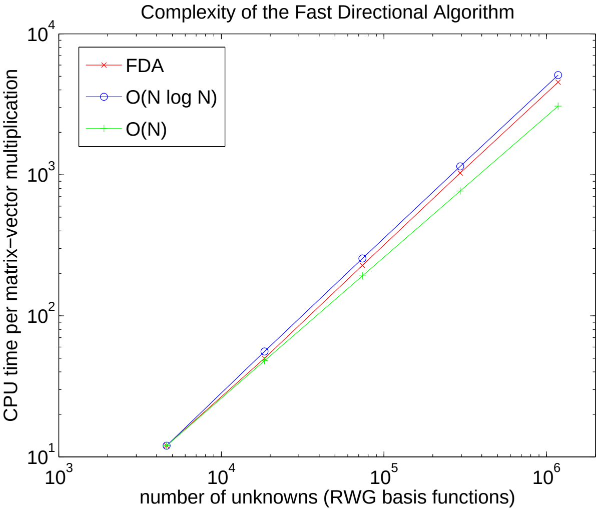


Figure 2.5: CPU time per matrix-vector multiplication vs. the number of unknowns for the sphere.

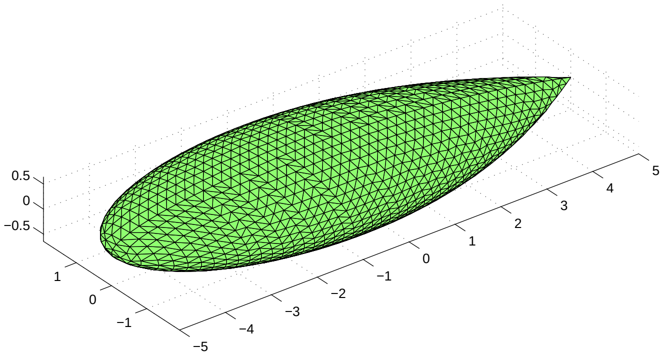
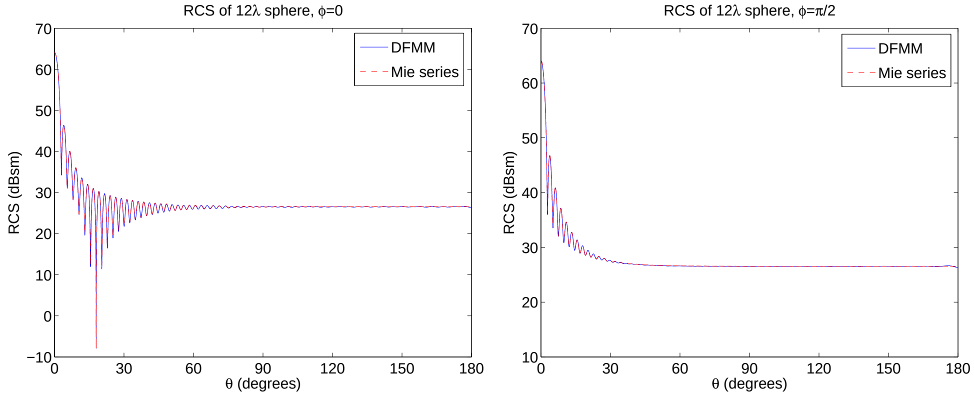


Figure 2.6: The NASA almond geometry.

tolerance of the directional algorithm and butterfly algorithm are set at 1e-4. The constant α, which determines the balance between the EFIE and MFIE, is set at 0.3. Since the goal is to produce sufficiently accurate RCS plots, more refined meshes with 10 elements per wavelength are utilized. To show the accuracy of the butterfly algorithm, the radar cross section computations are compared to either analytical results or measured experiments. For analytical results, the relative error of the radar cross section is measured; that is, if $\sigma(\hat{\mathbf{r}})$ is the analytical solution of the RCS and $\sigma^c(\hat{\mathbf{r}})$ is the numerical result, given the point set $\{\hat{\mathbf{r}}_s\}_{s=1}^{N_{\hat{\mathbf{r}}}}$ on the unit sphere, this error is defined as

$$\frac{\sqrt{\sum_{s=1}^{N_{\hat{\mathbf{r}}}} |\sigma(\hat{\mathbf{r}}_s) - \sigma^c(\hat{\mathbf{r}}_s)|^2}}{\sqrt{\sum_{s=1}^{N_{\hat{\mathbf{r}}}} |\sigma(\hat{\mathbf{r}}_s)|^2}}. \tag{2.52}$$

Three commonly-used surfaces were used for these tests. The first example is a conducting sphere, which has well-documented analytical solutions [52]. In this setup, the incident plane wave is propagating in the $-\hat{\mathbf{z}}$ direction, with the E-field in the $+\hat{\mathbf{x}}$ direction and H-field in the $-\hat{\mathbf{y}}$ direction. For each simulation, the bistatic RCS was calculated along θ for $\phi = 0$ and $\phi = \frac{\pi}{2}$; here, θ and ϕ take on their usual definitions in spherical coordinates. Table 2.3(top) shows the bistatic RCS of a 12λ-radius sphere. The Mie series solution is compared to the boundary element method using the directional multilevel algorithm in Table 2.3(bottom) for diameters K=6, 12, and 24. Both the solver times and number of unknowns for each example are listed to show the $O(N \log N)$ scaling, where N is the number of RWG basis functions. Here, T_{solve} is the total run-time of the GMRES iterative solver, and N_{iter} is the



(K,ε)	N	T_{solve}(sec)	N_{iter}	RCS error
(6,1e-4)	7.373e+4	3.439e+3	18	1.659e-02
(12,1e-4)	2.949e+5	1.856e+4	23	1.136e-02
(24,1e-4)	1.180e+6	9.602e+4	28	8.851e-03

Table 2.3: Top: Bistatic RCS of a 12λ radius sphere ($K = 24$). Bottom: Solver times and RCS relative error 2-norms for the boundary element code with the fast directional algorithm for the sphere.

number of iterations necessary for convergence. It is important to note that no preconditioning techniques were used, leading to an increased number of iterations for higher frequency problems.

The second example is a conducting cube with the same incoming field as the previous example. Table 2.4(top) shows the bistatic RCS of a cube with sides which are 15λ long; although there is no analytical solution for the RCS or measured data on the cube, two figures from the MLFMA paper by Song and Chew [84] are placed side-by-side as a comparison. The running time and iteration numbers are reported in Table 2.4(bottom). It is clear that the RCS for the $\phi = 0$ cut is not exactly symmetric, but by no fault of the butterfly

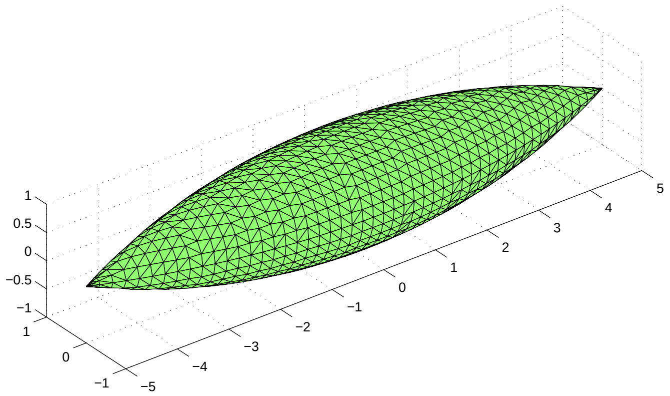
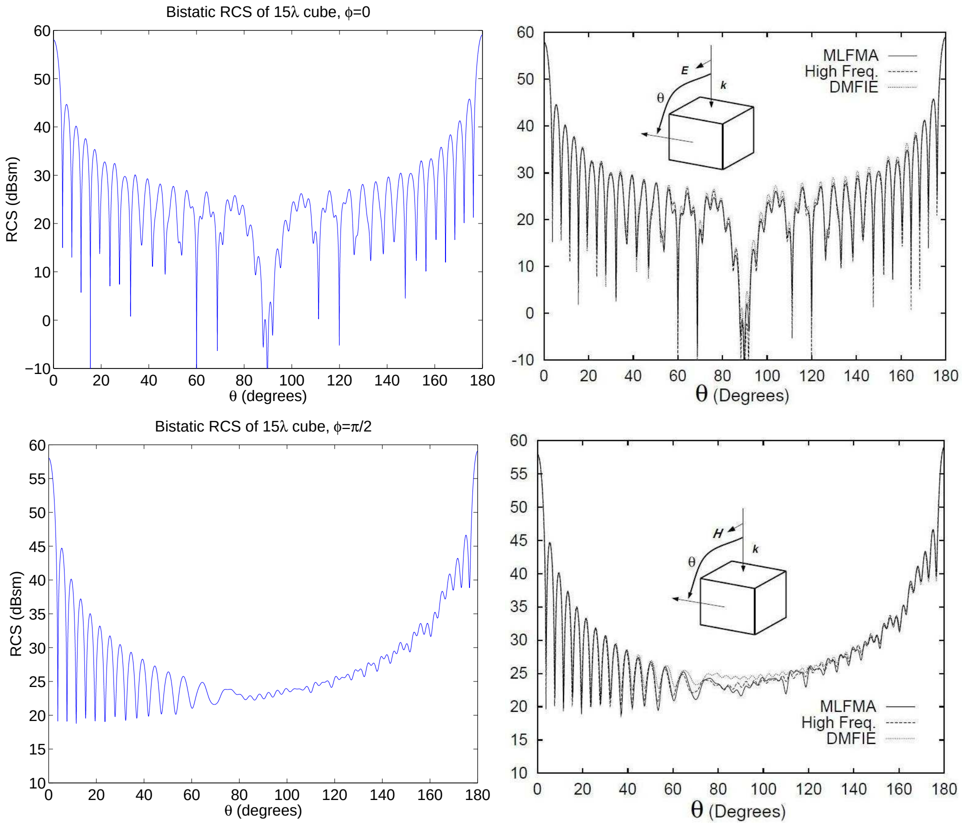


Figure 2.7: The ogive geometry.

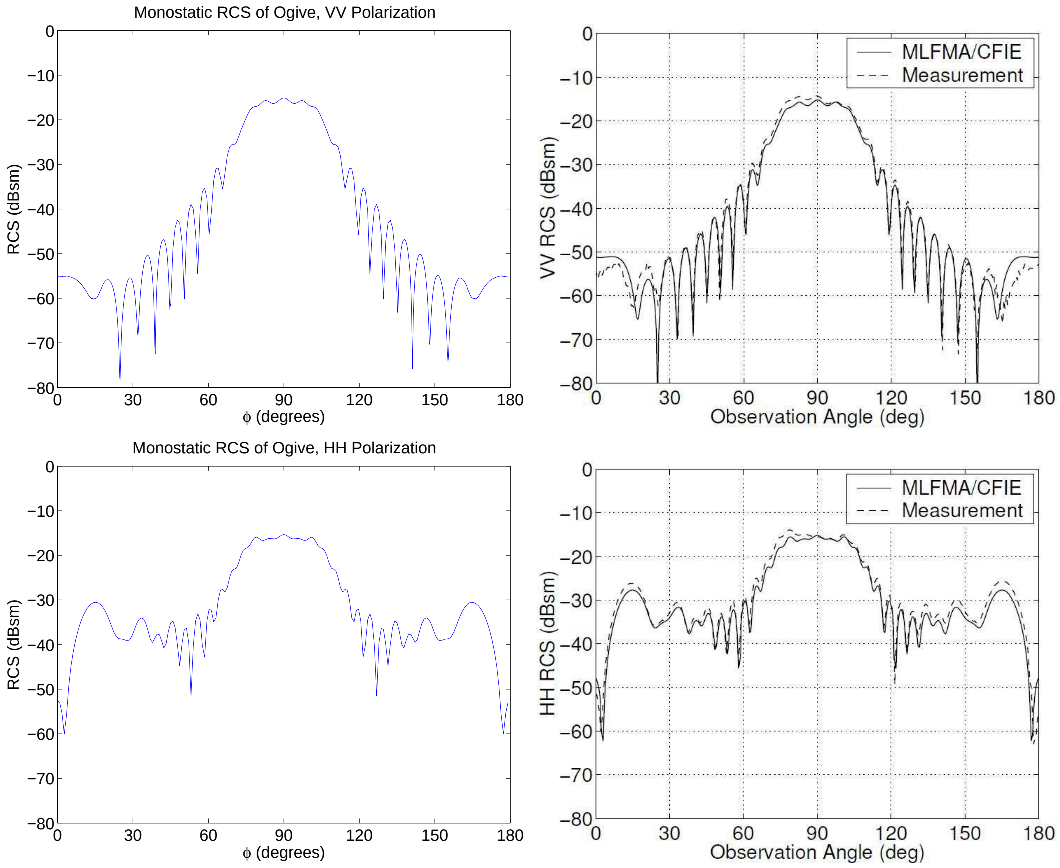
algorithm; this is merely the result of using a crude testing scheme and not employing accurate near-field techniques in the regions of corners and edges.

The third example, the ogive, is a popular benchmark target for testing electromagnetics codes [94] (see Figure 2.7). Here, the specific example is of the 10 inch ogive at a frequency of 9.0 GHz. Since the wavelength is assumed to be 1 at all times, the geometry is scaled appropriately so that the object is still of the same diameter K in terms of wavelength. Table 2.5(top) shows the monostatic RCS about the observation angle ϕ. In these plots, VV polarization signifies the E-field oriented in the $+\hat{\mathbf{z}}$ direction, while HH polarization signifies the H-field oriented in the $-\hat{\mathbf{z}}$ direction. In the comparison plots provided by Gibson in [42], the RCS curves are not normalized by wavelength; thus, in order to scale the RCS calculations down to their actual levels, one must compute $10\log_{10}(RCS \cdot \lambda^2)$, where λ is the free-space wavelength at the chosen frequency. Table 2.5(bottom) illustrates the complexity in solving larger problems; that is, for K=16, 32, and 64.



(K, ε)	N	T_{solve}(sec)	N_{iter}
(3.75,1e-4)	1.843e+4	9.620e+2	14
(7.5,1e-4)	7.373e+4	2.887e+3	16
(15,1e-4)	2.949e+5	1.466e+4	18

Table 2.4: Top: Bistatic RCS of a 15λ cube. On the left are figures for the boundary element code using the fast directional algorithm and butterfly algorithm. On the right are figures done by Song and Chew in [84] using the MLFMA. Bottom : Solver times for the boundary element code with the fast directional algorithm for the cube.



(K, ε)	N	T_{solve}(sec)	N_{iter}
(16,1e-4)	3.246e+4	1.012e+3	14
(32,1e-4)	1.297e+5	5.183e+3	16
(64,1e-4)	5.188e+5	2.529e+4	18

Table 2.5: Top: Monostatic RCS of the ogive at 9.0 GHz. On the left are figures for the boundary element code using the fast directional algorithm and butterfly algorithm. On the right are figures done by Gibson in [42] using the MLFMA. Bottom: Solver times for the boundary element code with the fast directional algorithm for the ogive.

Chapter 3

Uncertainty Quantification for Acoustic Scattering

In many practical situations, the shape and properties of the scattering object may be slightly perturbed from the specifications of the original geometry. This may occur if a vehicle has manufacturing defects, or if it has suffered damage after combat use. As a result, there is a level of uncertainty when observing physical quantities that are dependent on the characteristics of the scatterer. This chapter will go over my work on random surface scattering for the Helmholtz equation.

3.1 Boundary Integral Equations for Acoustics

For the scalar Helmholtz equation (1.2), if a is the identity tensor and c is constant, the exterior Helmholtz problem reads as

$$\begin{aligned} \Delta u + \kappa^2 u &= 0 \qquad \text{in } \mathbb{R}^d \backslash D & (3.1) \\ u &= -u_{\text{inc}} \quad \text{on } \partial D & (3.2) \\ \lim_{r\to\infty} r\left(\frac{\partial u}{\partial r} - \imath\kappa u\right) &= 0. & (3.3) \end{aligned}$$

Using Green's identities, it has been proven that the scattered field u can be represented as a combination of single and double layer potentials

$$u(\mathbf{x}) = \int_{\partial D} \left[\frac{\partial G(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{y})} - \imath\kappa G(\mathbf{x},\mathbf{y}) \right] \varphi(\mathbf{y}) ds(\mathbf{y}), \tag{3.4}$$

for surface density φ, where G is the free-space Green's function for the Helmholtz equation,

$$G(\mathbf{r},\mathbf{r}') = \begin{cases} \frac{\imath}{4} H_0^1(\kappa|\mathbf{r}-\mathbf{r}'|) & \text{in 2D} \\ \frac{e^{-\imath\kappa|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} & \text{in 3D} \end{cases}. \tag{3.5}$$

Taking the limit as $\mathbf{x}$ approaches ∂D gives us the combined field integral equation

$$-u_{\text{inc}}(\mathbf{x}) = \frac{1}{2}\varphi(\mathbf{x}) + \int_{\partial D} \left[\frac{\partial G(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{y})} - \imath\kappa G(\mathbf{x},\mathbf{y}) \right] \varphi(\mathbf{y}) ds(\mathbf{y}). \tag{3.6}$$

The solution process for computing u is as follows: first, solve the boundary integral equation for the density φ. After this density is found, the 2D far field pattern F is computed by

$$F(s) = \frac{e^{-\imath\frac{\pi}{4}}}{\sqrt{8\pi\kappa}} \int_{\partial D} \{\kappa\left(\mathbf{n}(\mathbf{y})\cdot\mathbf{s}\right) + \eta\} e^{-\imath\kappa\mathbf{s}\cdot\mathbf{y}} \varphi(\mathbf{y}) d\mathbf{y}. \tag{3.7}$$

3.2 Nystöm Method

Various discretization techniques exist for the integral equations above. For acoustics, the most popular method is Nyström discretization [71, 61, 4], where the field takes a pointwise representation by way of quadrature rule applied to the boundary surface. This is especially attractive in two dimensions, since the quadrature scheme can be defined on the unit circle. Consider the quadrature rule with weights $\{w_j\}_{j=1}^N$ and points $\{\mathbf{x}_j\}_{j=1}^N$. Starting with (3.6), the equation is enforced at each node $\mathbf{x}_i$, i.e.

$$-u_{\text{inc}}(\mathbf{x}_i) = \frac{1}{2}\varphi(\mathbf{x}_i) + \int_{\partial D} \left[\frac{\partial G(\mathbf{x}_i, \mathbf{y})}{\partial n(\mathbf{y})} - \imath\kappa G(\mathbf{x}_i, \mathbf{y})\right]\varphi(\mathbf{y})ds(\mathbf{y}). \tag{3.8}$$

To discretize the integral, a combination of the original quadrature rule and a singularity correction rule must be used in a small neighborhood of $\mathbf{x}_i$. The fully discretized equation takes the form

$$-u_{\text{inc}}(\mathbf{x}_i) = \frac{1}{2}\varphi(\mathbf{x}_i) + \sum_{i\neq j} w_j \left[\frac{\partial G(\mathbf{x}_i, \mathbf{x}_j)}{\partial n(\mathbf{x}_j)} - \imath\kappa G(\mathbf{x}_i, \mathbf{x}_j)\right]\varphi(\mathbf{x}_j) \tag{3.9}$$

$$+ \sum_{j} \tilde{w}_j \left[\frac{\partial G(\mathbf{x}_i, \tilde{\mathbf{x}}_j)}{\partial n(\tilde{\mathbf{x}}_j)} - \imath\kappa G(\mathbf{x}_i, \tilde{\mathbf{x}}_j)\right]\varphi(\tilde{\mathbf{x}}_j). \tag{3.10}$$

where the singularity correction weights and nodes for the integral near $\mathbf{x}_i$ are $\{\tilde{w}_j\}$ and $\{\tilde{\mathbf{x}}_j\}$.

3.3 Stochastic Methods

To quantify the far-field and radar cross section of a randomly perturbed scatterer, stochastic methods are necessary. The traditional approach is the Monte Carlo method, but this usually results in long computational times due to the slow $O(1/\sqrt{N})$ convergence with respect to the number of realizations N. Very recently, a class of methods based on generalized polynomial chaos (gPC) have been developed and become popular in many practical applications. Most notable is the stochastic collocation method using Smolyak sparse grids [96], which may offer much better convergence properties than the Monte Carlo method while keeping the same ease of implementation. For wave scattering with random shapes, the gPC method was applied in [97] and found to be effective in low-frequency scattering. However, for the high-frequency scattering problem considered here, the sparse grid collocation method does not offer a big advantage over other methods. In order to resolve the highly oscillatory solution, a higher order method is required in the random space; in addition, to properly model the rough physical domain, the random space needs to be parametrized by a larger set of random variables. Therefore, for gPC-based methods, the problem would require a high-order implementation in a large number of dimensions. To alleviate this computational difficulty, quasi-Monte Carlo (QMC) methods based on low discrepancy sequences are introduced. The QMC methods [70, 18] are in fact deterministic approaches based on pseudo-random numbers; they have much faster convergence rates ($O(1/N)$ up to logarithmic factors) without sacrificing the generality of the Monte Carlo method, and their dependence on dimensionality is much weaker than for stochastic collocation methods.

To incorporate the uncertainty of the scatterer D, a probabilistic setting is adopted and the surface is modeled as a random process. In this section only, N is denoted as the number of random samples and s_ℓ, $\ell = 0, 1, ..., N_s$ are the far-field observation directions

$$s_\ell = (s_{\ell,1}, s_{\ell,2}) = \left(\cos\left(\frac{2\pi\ell}{N_s}\right), \sin\left(\frac{2\pi\ell}{N_s}\right)\right).$$

The 2D boundary takes the form

$$\partial D_{z(\omega)} = \{x(t,\omega) = b(t) \cdot (1 + p(t,\omega)), t \in [0, 2\pi), \omega \in \Omega\},$$

where $b(t) = (b_1(t), b_2(t))$ is the base geometry, Ω is the event space in a properly defined probability space, and $p(t, \omega)$ is the perturbation. For a fixed ω, $p(t, \omega)$ is a deterministic function representing how the base geometry $b(t)$ is scaled, while for a fixed location t, $p(t, \omega)$ is a random variable representing the uncertainty of the surface at the location associated with t. The perturbation $p(t, \omega)$ is also assumed to be sufficiently regular so that the scattering problem is well posed almost everywhere in Ω.

A critical step in modeling the random surface is to properly parametrize the random process by a finite number of independent random variables. Let $Z(\omega) = (Z_1(\omega), \ldots, Z_M(\omega))$, $M \geq 1$, be such a set of independent random variables, whose probability distribution is $F_Z(z) = \text{Prob}(Z \leq z)$, where $z \in \mathbb{R}^M$. Without loss of generality, I focus on the continuous random variables, where a probability density function $\rho(z) = dF_Z(z)/dz$ exists. The random surface can now be expressed in terms of Z in the following manner:

$$\partial D_z = \{b(t) \cdot (1 + p(t, Z)), t \in [0, 2\pi), Z \in \mathbb{R}^M\}.$$

Now, the integral formulations given in (3.6) and (3.7) depend on z. The density $\varphi_z(\mathbf{x})$ for $\mathbf{x} \in \partial D_z$ satisfies

$$-u_{\text{inc}}(\mathbf{x}) = \frac{1}{2}\varphi(\mathbf{x}) + \int_{\partial D_z} \left[\frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial n(\mathbf{y})} - \imath\kappa G(\mathbf{x}, \mathbf{y})\right] \varphi_z(\mathbf{y}) ds(\mathbf{y}). \tag{3.11}$$

The far field pattern and the radar cross sections are equal to

$$\begin{aligned} F_z(s) &= \frac{e^{-\imath\frac{\pi}{4}}}{\sqrt{8\pi\kappa}} \int_{\partial D_z} \{\kappa\left(\mathbf{n}(\mathbf{y}) \cdot \mathbf{s}\right) + \eta\} e^{-\imath\kappa\mathbf{s}\cdot\mathbf{y}} \varphi_z(\mathbf{y}) d\mathbf{y} & (3.12) \\ R_z(s) &= |F_z(s)|^2 & (3.13) \end{aligned}$$

Finally, the mean and the variance of the observable $R(s)$ are given by

$$\begin{aligned} E[R(s)] &= \int R_z(s)\rho(z)dz, && (3.14)\\ Var[R(s)] &= \int \left(R_z(s) - E[R(s)]\right)^2 \rho(z)dz. && (3.15) \end{aligned}$$

One can follow [70] for a short description of the quasi-Monte Carlo methods. The main idea of the quasi-Monte Carlo method is the construction of low discrepancy sequences. For any integer $b \geq 2$, define $\mathbb{Z}_b = \{0, 1, \ldots, b-1\}$. For any integer $n \geq 1$, one can write the unique b-ary representation of n as

$$n = \sum_{j=0}^{\infty} a_j(n)b^j, \quad a_j(n) \in \mathbb{Z}_b.$$

The *radical inverse function* $\phi_b(n)$ is defined to be

$$\phi_b(n) = \sum_{j=0}^{\infty} a_j(n)b^{-j-1}, \quad \forall n \geq 0.$$

Clearly, $0 \leq \phi_b(n) \leq 1$. Two of the most commonly used low discrepancy sequences are defined based on the radical inverse functions. Let M be an arbitrary dimension and $b_1, \ldots, b_M$ coprime to each other. The *Halton sequence* is defined for each integer $n > 0$ as

$$z^{(n)} = (\phi_{b_1}(n), \ldots, \phi_{b_M}(n)) \in [0,1]^M$$

The definition of the *Hammersley sequence* is similar. Let M be the dimension, N be the length of the sequence, and $b_1, \ldots, b_{M-1}$ coprime to each other. The *Hammersley sequence* is defined for $n = 1, \ldots, N$ as

$$z^{(n)} = \Big(\frac{n}{N}, \phi_{b_1}(n), \ldots, \phi_{b_{M-1}}(n)\Big) \in [0,1]^M.$$

For a fixed sample size N, the samples $z^{(1)}, z^{(2)}, \ldots, z^{(N)}$ are generated using a low discrepancy sequence (in the numerical examples, the Hammersley sequence is chosen due to its lower discrepancy). For each sample $z^{(i)}$, the directional FMM is used to solve the Helmholtz problem and compute the RCS $R_{z^{(i)}}(s_\ell)$ for $\ell = 0, 1, \ldots, N_s - 1$. Once they are ready, the statistical estimations of the mean and variance are given, respectively, by

$$\begin{aligned}\bar{R}_N(s_\ell) &= \frac{1}{N}\sum_{i=1}^{N} R_{z^{(i)}}(s_\ell)\\ \bar{V}_N(s_\ell) &= \frac{1}{N-1}\sum_{i=1}^{N}\left(R_{z^{(i)}}(s_\ell) - \bar{R}_N(s_\ell)\right)^2.\end{aligned}$$

3.4 Numerical Results for Random Surface Scattering

In this section, the results of some numerical experiments are presented. The two base shapes that were tested are the cylinder and the kite; these objects were chosen because they are smooth and have a simple parametrization in the two-dimensional plane.

- Cylinder. $b(t) = (b_1(t), b_2(t)) = \frac{K}{2}(\cos(t), \sin(t))$.
- Kite. $b(t) = (b_1(t), b_2(t)) = \frac{K}{2}\left(\frac{\cos(t)+0.65\cos(2t)-0.65}{1.5}, \sin(t)\right)$.

The perturbation $p(t, z)$ is modeled as follows. First, choose a set number of frequencies or modes $\{\xi_i\}_{i=1}^{M/2}$. For simplicity, it is assumed that each component Z_i of the random parameter $Z = (Z_1, \ldots, Z_M)$ has a uniform probability density function over the unit interval $[0, 1]$ (this assumption

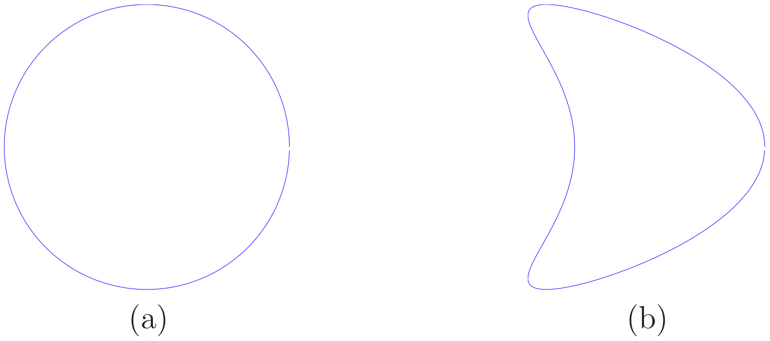


Figure 3.1: The base shape of the scatterers used in the test. (a) Circle. (b) Kite.

can certainly be removed by performing appropriate re-parametrization to each Z_i). As a result, the joint probability density function for z is the constant one function over the M-dimensional cube $[0,1]^M$. For a given sample $Z = (Z_1, \ldots, Z_M)$, the perturbation $p(t, Z)$ is defined as

$$p(t,Z) = \frac{\mu}{K} \sum_{i=1}^{M/2} \left(\left(Z_{2i-1} - \frac{1}{2} \right) \cos(\xi_i t) + \left(Z_{2i} - \frac{1}{2} \right) \sin(\xi_i t) \right). \qquad (3.16)$$

Depending on the choice of the frequencies $\{\xi_i\}_{i=1}^{M/2}$, $p(t, Z)$ can model both low frequency and high frequency perturbations.

- Low frequency perturbation sets $\xi_i = i$ for $i = 1, 2, \ldots, M/2$; thus, the perturbation function does not have many oscillations and the resulting boundary ∂D does not have rough edges.

- High frequency perturbation sets $\xi_i = \frac{iK}{M}$ for $i = 1, 2, \ldots, M/2$. Here, the high frequency range extends to modes which are comparable to the size of the scattering object in terms of wavelength and the resulting boundary ∂D exhibits small-scale oscillations.

In each case, the uncertainty quantification results of the Monte Carlo method and the quasi-Monte Carlo method are compared for a fixed sample size N. In Monte-Carlo, the random parameter sample $Z^{(i)} = (Z_1^{(i)}, \ldots, Z_M^{(i)})$ for $i = 1, 2, \ldots, N$ is generated randomly for each entry $Z_j^{(i)}$ for $j = 1, 2, \ldots, M$. In quasi-Monte Carlo, the random parameter $Z^{(i)} = (Z_1^{(i)}, \ldots, Z_M^{(i)})$ for each i is constructed using the Hammersley sequence. For the simulations, the goal is to see how the estimations of the expected value and the variance of the radar cross section converge for the Monte Carlo and quasi-Monte Carlo. In each case, the statistical estimations of the mean and variance for a fixed sample size N are given, respectively, by

$$\bar{R}_N(s_\ell) = \frac{1}{N}\sum_{i=1}^{N} R_{z^{(i)}}(s_\ell)$$

$$\bar{V}_N(s_\ell) = \frac{1}{N-1}\sum_{i=1}^{N} \left(R_{z^{(i)}}(s_\ell) - \bar{R}_N(s_\ell)\right)^2.$$

In order to measure the convergence rate depending on the sample size N, the error is estimated using the relative ℓ_2 norm. Suppose that N_{max} is the largest sample size used in the tests. Then for each fixed N, the errors $\varepsilon_{\bar{R},N}$ and $\varepsilon_{\bar{V},N}$ are defined as

$$\varepsilon_{\bar{R},N} = \sqrt{\frac{\sum_{\ell=0}^{N_s-1} |\bar{R}_N(s_\ell) - \bar{R}_{N_{max}}(s_\ell)|^2}{\sum_{\ell=0}^{N_s-1} |\bar{R}_{N_{max}}(s_\ell)|^2}}$$

$$\varepsilon_{\bar{V},N} = \sqrt{\frac{\sum_{\ell=0}^{N_s-1} |\bar{V}_N(s_\ell) - \bar{V}_{N_{max}}(s_\ell)|^2}{\sum_{\ell=0}^{N_s-1} |\bar{V}_{N_{max}}(s_\ell)|^2}}$$

In these tests, the diameter of the scatterer K is set to be 512, the number of random modes $M = 8$ and the perturbation amplitude in (3.16)

$\mu = 0.1$. The incident field is chosen to be a plane wave propagating in the x_1-direction, i.e.

$$u_{\text{inc}}(x) = e^{2\pi\imath x_1}, \tag{3.17}$$

where $x = (x_1, x_2)$; once again, the wavenumber is 2π and the wavelength λ is 1. Both the cylinder and kite geometries were tested, using both low frequency and high frequency perturbations mentioned earlier. First, the results of stochastic collocation using Smolyak sparse grids for the random parameter are given, with accuracy level of 1, 2, and 3. Figure 3.2 shows close-ups of the variance curve for the low frequency and high frequency perturbations of the kite, respectively. It is clear that stochastic collocation produces somewhat nonsensical results, as it should be impossible to have a negative value for the variance of the RCS. This artifact is purely a result of utilizing negative weights in the quadrature of the random space; for this reason, stochastic collocation does not work well when the solution is highly oscillatory.

Next, the results of both the Monte Carlo and the quasi-Monte Carlo methods are presented. In order to measure the convergence, different sample sizes of $N = 64, 256, 1024$ are used with $N_{max} = 1024$ for the highest order accuracy. Figures 3.3 and 3.4 summarize the results of the cylinder for the low frequency and high frequency perturbations, respectively. The errors in both cases are tabulated in Table 3.1. For low frequency perturbations, the expectation and variance converge significantly faster for quasi-Monte Carlo when the sample size N increases. However, for the high frequency perturbations, it is observed that the improvement in error for both quantities is modest at best;

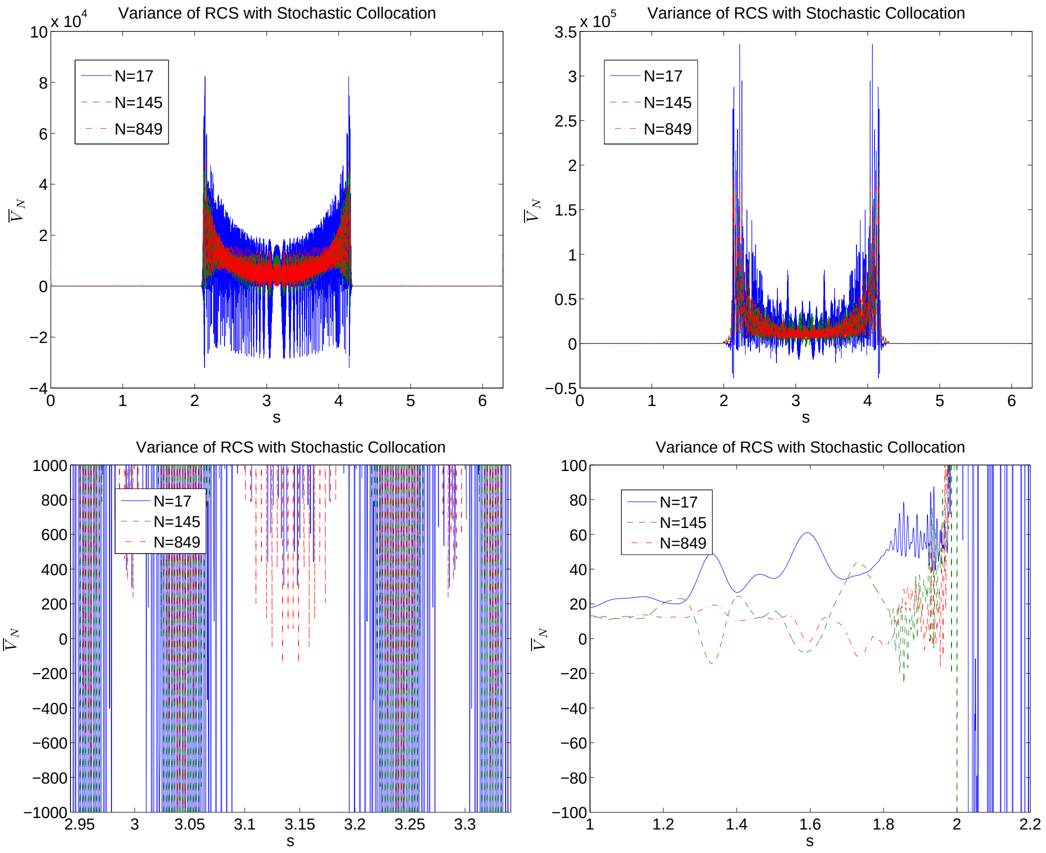


Figure 3.2: The variance of the radar cross section for the kite, using stochastic collocation. The figures on the left are for low frequency perturbations, while the figures on the right are for high frequency perturbations. For each type of perturbation, the top figure shows the full plot, while the bottom figure shows a close-up where the curves show negative variance.

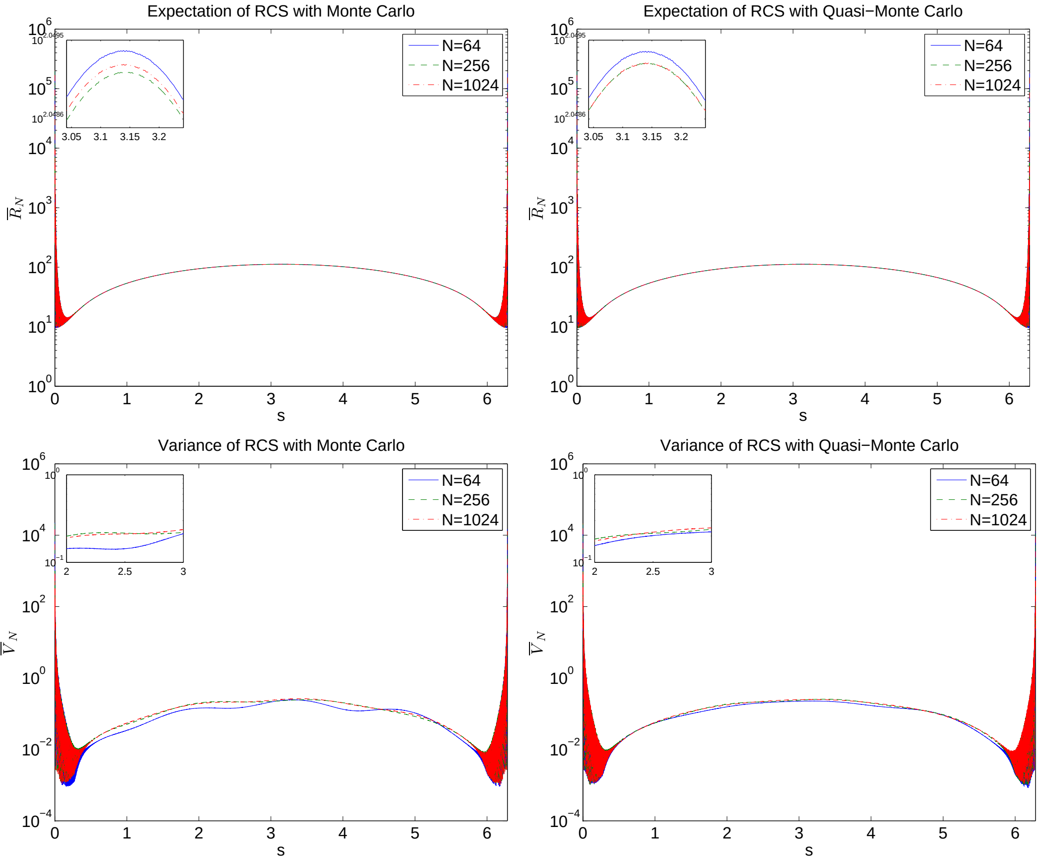


Figure 3.3: The expectation and variance of the radar cross section for low frequency perturbations on the cylinder. The figures on the left are regular Monte Carlo, while the figures on the right use the Hammersley low-discrepancy sequence.

that is, the rougher the surface of the cylinder, the more difficult it is to quantify the RCS accurately. Tests for larger values of M were also performed and gave similar results for both low-frequency and high-frequency perturbations.

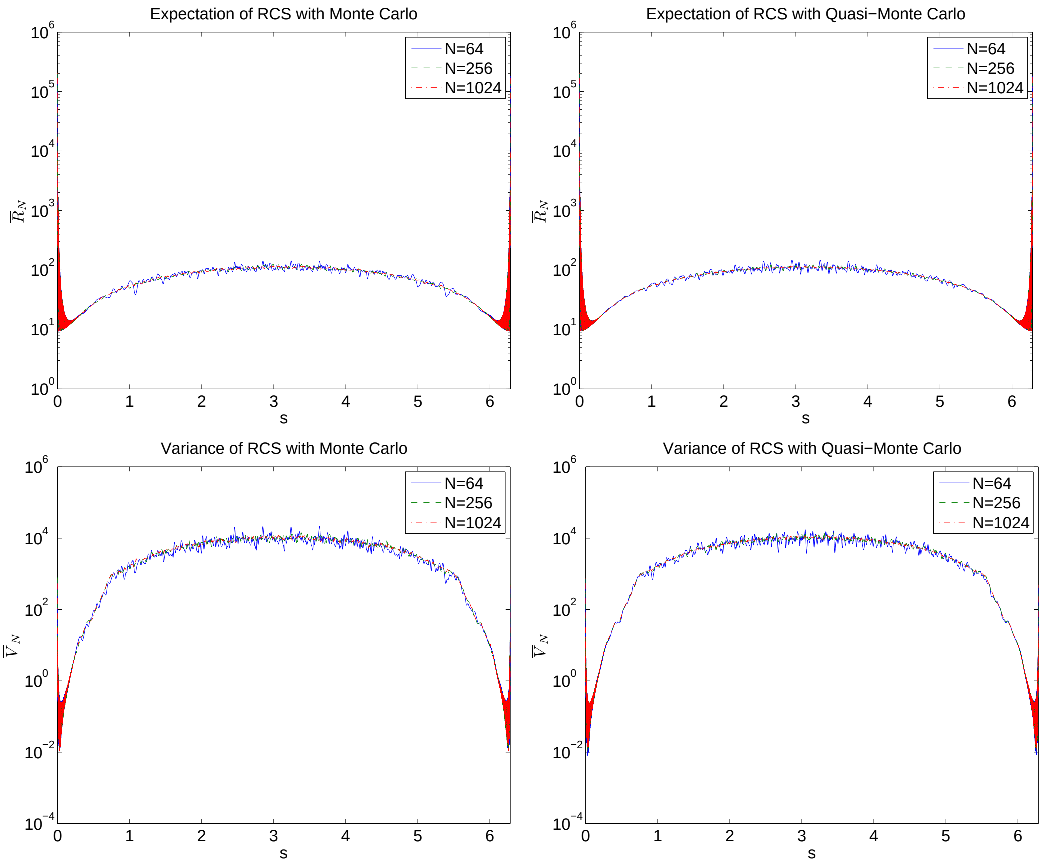
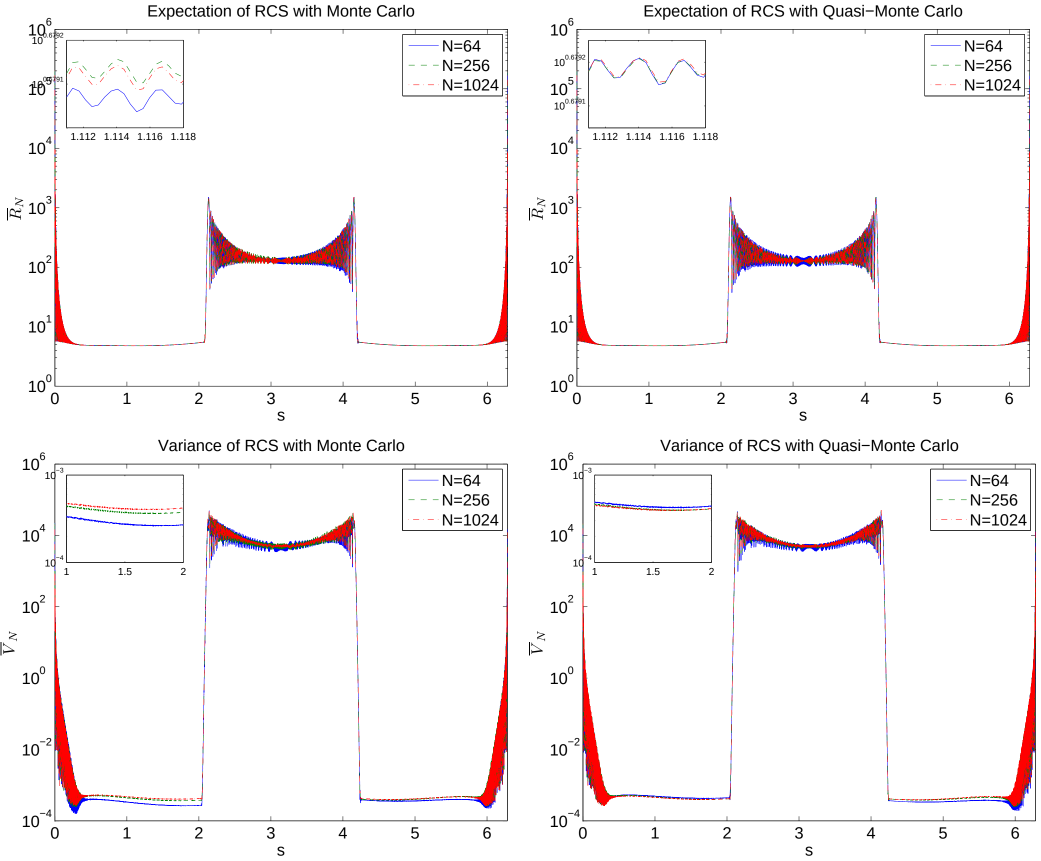


Figure 3.4: The expectation and variance of the radar cross section for high frequency perturbations on the cylinder.

	low frequency perturbations		high frequency perturbations	
(method,N)	$\varepsilon_{\bar{R},N}$	$\varepsilon_{\bar{V},N}$	$\varepsilon_{\bar{R},N}$	$\varepsilon_{\bar{V},N}$
MC, 64	6.17e-05	1.33e-01	2.35e-03	2.69e-01
MC, 256	1.50e-05	9.68e-02	1.28e-03	1.62e-01
QMC, 64	2.71e-05	8.06e-02	2.25e-03	2.27e-01
QMC, 256	4.80e-06	3.48e-03	9.32e-04	1.11e-01

Table 3.1: 2-norm errors for the RCS of the cylinder geometry.


Expectation of RCS with Monte Carlo

Expectation of RCS with Quasi−Monte Carlo

Variance of RCS with Monte Carlo

Variance of RCS with Quasi−Monte Carlo

Figure 3.5: The expectation and variance of the radar cross section for low frequency perturbations on the kite scatterer.

	low frequency perturbations		high frequency perturbations	
(method,N)	$\varepsilon_{\bar{R},N}$	$\varepsilon_{\bar{V},N}$	$\varepsilon_{\bar{R},N}$	$\varepsilon_{\bar{V},N}$
MC, 64	1.91e-03	1.27e-01	3.39e-03	2.15e-01
MC, 256	1.07e-03	7.08e-02	1.84e-03	9.87e-02
QMC, 64	1.60e-03	9.90e-02	2.01e-03	1.29e-01
QMC, 256	5.01e-04	5.08e-02	6.45e-04	6.01e-02

Table 3.2: 2-norm errors for the RCS of the kite geometry.

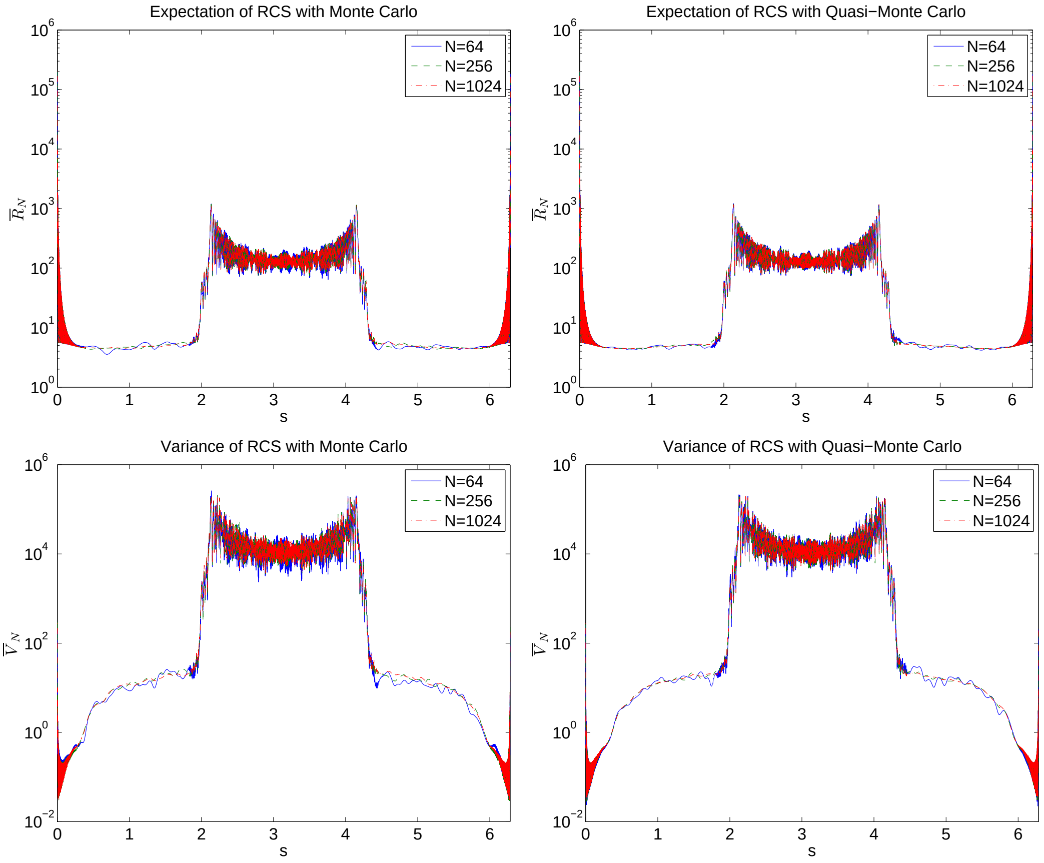


Figure 3.6: The expectation and variance of the radar cross section for high frequency perturbations on the kite scatterer.

Figures 3.5 and 3.6 summarize the results of the kite for the low frequency and high frequency perturbations, respectively, using quasi-Monte Carlo. The errors in both cases are tabulated in Table 3.2. The results suggests that, when the sample size N is quadrupled, the expectation for both the low frequency and high frequency perturbations converge by a factor of 3 for the quasi Monte-Carlo method and by a factor of 2 for the standard Monte-Carlo method. On the other hand, the convergence rates for the variance seem to be comparable for the two methods.

One difficulty that was encountered in the numerical tests is the sensitivity of the RCS calculation varying with the size of the perturbation μ. For larger perturbations approaching the size of the operating wavelength, such convergence to the actual mean or variance proved to be quite difficult without having an inordinate number of samples. In order to achieve something sensible, especially for high frequency problems, it was deduced computationally that the perturbation size must satisfy $\mu \leq \frac{\lambda}{5}$.

Part II: Sweeping Preconditioners for Time-Harmonic Wave Equations

The next few chapters will be dedicated to work I have done on developing the moving PML sweeping preconditioner [35] for a variety of physics and discretization techniques. I will proceed first with two chapters on Maxwell's equations, then go on to parallel preconditioners for the Helmholtz and elastic wave equation with higher order elements. I will conclude with a proof of the approximability of the free-space Green's function using the perfectly matched layer boundary condition.

Chapter 4

Preconditioners for Maxwell's Equations: the Yee Grid

4.1 PMLs for Maxwell's Equations

Recall the time-harmonic Maxwell equations (1.9) for general media on an infinite domain, i.e.

$$\begin{aligned}
\nabla \times \mathbf{E} &= -\imath\omega\mu_0\mu_r\mathbf{H} \\
\nabla \times \mathbf{H} &= \imath\omega\varepsilon_0\varepsilon_r\mathbf{E} + \mathbf{J} \\
\nabla \cdot (\varepsilon_0\varepsilon_r\mathbf{E}) &= q_e \\
\nabla \cdot (\mu_0\mu_r\mathbf{H}) &= 0 \\
\lim_{|\mathbf{r}|\to\infty} (\mathbf{H} \times \mathbf{r} - |\mathbf{r}|\mathbf{E}) &= 0 \\
\lim_{|\mathbf{r}|\to\infty} (\mathbf{E} \times \mathbf{r} + |\mathbf{r}|\mathbf{H}) &= 0.
\end{aligned}$$

It should be noted that the current and charge satisfy the continuity equation $\nabla \cdot \mathbf{J} = -\imath\omega q_e$. The last two limits are the Silver-Müller radiation conditions, which enforce the fields to radiate away from the current source and dissipate as $|\mathbf{r}|$ goes to infinity. The relative permittivity and permeability tensors ε_r and μ_r are measurable with respect to the spatial variable $\mathbf{r} \in \mathbb{R}^3$; as a reminder, the entries of these matrices are

$$\varepsilon_r = \begin{pmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{pmatrix}, \qquad \mu_r = \begin{pmatrix} \mu_{xx} & \mu_{xy} & \mu_{xz} \\ \mu_{yx} & \mu_{yy} & \mu_{yz} \\ \mu_{zx} & \mu_{zy} & \mu_{zz} \end{pmatrix}.$$

Since it is impossible to numerically solve Maxwell's equations in all of $\mathbb{R}^3$, the computational domain is truncated and a boundary condition which emulates the radiation condition is introduced; for this setting, the domain of interest is the unit cube $\Omega = [0, 1]^3$. Absorbing boundary conditions [68, 31] have been very popular for the wave equation. Here, the perfectly matched layer derived by complex-stretched coordinates [23] is chosen because of its ubiquity in computational electromagnetics.

Define the width of the PML as ℓ, so that the non-PML region in Ω is $[\ell, 1-\ell]^3$. The complex stretching variables s_ξ for $\xi = x, y, z$ are of the form

$$s_\xi(\xi) = a(\xi) + \imath\sigma(\xi), \tag{4.1}$$

with $a \geq 1$ and $\sigma \geq 0$; in the physical space outside the PML, $s_\xi = 1$. Typically, a is chosen to be 1 everywhere, and σ is the ramp-like function

$$\sigma(\xi) = \begin{cases} \theta\left(\frac{\xi-\ell}{\ell}\right)^2, & \xi \in [0, \ell] \\ 0, & \xi \in [\ell, 1-\ell] \\ \theta\left(\frac{\xi-1+\ell}{\ell}\right)^2, & \xi \in [1-\ell, 1] \end{cases}, \tag{4.2}$$

where θ is an optimal constant inversely proportional to frequency [58]. Now define the matrix

$$\mathbf{S} = \begin{pmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & \frac{1}{s_z} \end{pmatrix}, \tag{4.3}$$

so $\det\big(\mathbf{S}\big) = (s_x s_y s_z)^{-1}$. It has been shown in [24] that Maxwell's equations in complex stretched coordinates can be written in terms of this operator; when these equations are recast in the non-stretched coordinates, the material tensors take on the form

$$\tilde{\varepsilon}_r = \big(\det\big(\mathbf{S}\big)\big)^{-1}\big(\mathbf{S}\varepsilon_r\mathbf{S}\big), \quad \tilde{\mu}_r = \big(\det\big(\mathbf{S}\big)\big)^{-1}\big(\mathbf{S}\mu_r\mathbf{S}\big).$$

Explicitly, these matrix entries are

$$\tilde{\varepsilon}_r = \begin{pmatrix} \varepsilon_{xx}\frac{s_y s_z}{s_x} & \varepsilon_{xy}s_z & \varepsilon_{xz}s_y \\ \varepsilon_{yx}s_z & \varepsilon_{yy}\frac{s_x s_z}{s_y} & \varepsilon_{yz}s_x \\ \varepsilon_{zx}s_y & \varepsilon_{zy}s_x & \varepsilon_{zz}\frac{s_x s_y}{s_z} \end{pmatrix}, \quad \tilde{\mu}_r = \begin{pmatrix} \mu_{xx}\frac{s_y s_z}{s_x} & \mu_{xy}s_z & \mu_{xz}s_y \\ \mu_{yx}s_z & \mu_{yy}\frac{s_x s_z}{s_y} & \mu_{yz}s_x \\ \mu_{zx}s_y & \mu_{zy}s_x & \mu_{zz}\frac{s_x s_y}{s_z} \end{pmatrix}.$$

Now, $\tilde{\varepsilon}_r$ and $\tilde{\mu}_r$ can be used as the material tensors in the whole computational domain, since $s_\xi = 1$ in $[\ell, 1-\ell]^3$ and the PML reduces to the actual material. At the boundary of the domain $\partial\Omega$, a PEC boundary condition is artificially placed to close the system; because of the exponential decay of any plane wave entering the perfectly matched layer, the field is so small that the reflections off of the PEC outside of the PML are deemed insignificant. The infinite domain problem is now reduced to the truncated problem,

$$\left.\begin{aligned} \nabla\times\mathbf{E} &= -\imath\omega\mu_0\tilde{\mu}_r\mathbf{H} \\ \nabla\times\mathbf{H} &= \imath\omega\varepsilon_0\tilde{\varepsilon}_r\mathbf{E} + \mathbf{J} \\ \nabla\cdot\big(\varepsilon_0\tilde{\varepsilon}_r\mathbf{E}\big) &= q_e \\ \nabla\cdot\big(\mu_0\tilde{\mu}_r\mathbf{H}\big) &= 0 \end{aligned}\right. \quad \text{in } \Omega$$
$$\left.\begin{aligned} \hat{\mathbf{n}}\times\mathbf{E} &= 0 \\ \hat{\mathbf{n}}\times\mathbf{H} &= 0 \end{aligned}\right. \quad \text{on } \partial\Omega. \tag{4.4}$$

4.2 Yee Grid Discretization

For the scalar Helmholtz equation, standard central differencing methods are sufficient and produce reasonably accurate results. In the case of Maxwell's equations, however, dispersion becomes a major problem if the components of the electric field **E** and magnetic field **H** are all defined at the same locations. The celebrated Yee grid, which was originally used for finite-difference time-domain simulations [86], is also applicable to the frequency domain. In this scheme, the components of **E** and **H** are defined on a staggered grid. The advantages to using the Yee grid are two-fold: the divergence equations in (1.9) are implicitly satisfied, and boundary conditions between materials are naturally handled.

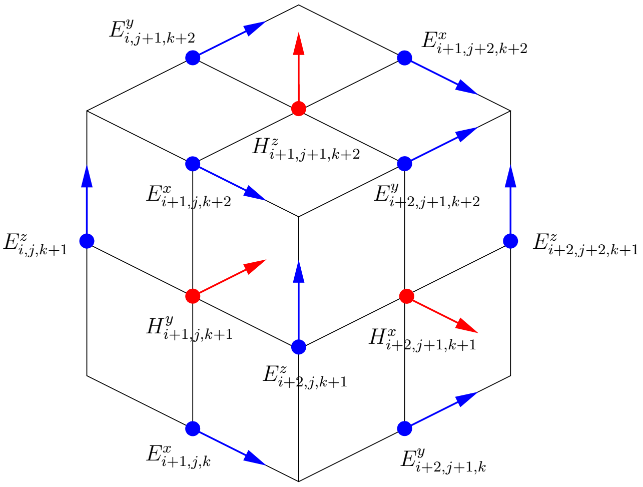


Figure 4.1: The Yee grid on a cubic cell. The red vectors are components of **H** and the blue vectors are components of **E**.

Figure 4.1 illustrates the locations of the field components in Yee's scheme for an orthogonal Cartesian grid; with this structure, the finite difference formulas are

$$
\begin{aligned}
E^{y}_{i,j+1,k} - E^{y}_{i,j+1,k+2} + E^{z}_{i,j+2,k+1} - E^{z}_{i,j,k+1} + & \\
2\imath\omega h(\mu_0\tilde{\mu}_r\mathbf{H})^{x}_{i,j+1,k+1} &= 0 \\
E^{z}_{i,j,k+1} - E^{z}_{i+2,j,k+1} + E^{x}_{i+1,j,k+2} - E^{x}_{i+1,j,k} + & \\
2\imath\omega h(\mu_0\tilde{\mu}_r\mathbf{H})^{y}_{i+1,j,k+1} &= 0 \\
E^{x}_{i+1,j,k} - E^{x}_{i+1,j+2,k} + E^{y}_{i+2,j+1,k} - E^{y}_{i,j+1,k} + & \\
2\imath\omega h(\mu_0\tilde{\mu}_r\mathbf{H})^{z}_{i+1,j+1,k} &= 0 \\
H^{z}_{i+1,j+1,k} - H^{z}_{i+1,j-1,k} + H^{y}_{i+1,j,k-1} - H^{y}_{i+1,j,k+1} - & \\
2h\big(\imath\omega(\varepsilon_0\tilde{\varepsilon}_r\mathbf{E})^{x}_{i+1,j,k}\big) &= 2h\big(J^{x}_{i+1,j,k}\big) \\
H^{x}_{i,j+1,k+1} - H^{x}_{i,j+1,k-1} + H^{z}_{i-1,j+1,k} - H^{z}_{i+1,j+1,k} - & \\
2h\big(\imath\omega(\varepsilon_0\tilde{\varepsilon}_r\mathbf{E})^{y}_{i,j+1,k}\big) &= 2h\big(J^{y}_{i,j+1,k}\big) \\
H^{y}_{i+1,j,k+1} - H^{y}_{i-1,j,k+1} + H^{x}_{i,j-1,k+1} - H^{x}_{i,j+1,k+1} - & \\
2h\big(\imath\omega(\varepsilon_0\tilde{\varepsilon}_r\mathbf{E})^{z}_{i,j,k+1}\big) &= 2h\big(J^{z}_{i,j,k+1}\big),
\end{aligned}
\tag{4.5}
$$

where $h = \frac{1}{n+1}$ is the distance between two nodes in the grid, and the subscript notation for each component follows the convention $E^{x}_{i,j,k} \approx E_x(ih, jh, kh)$.

In the derivation of the finite difference formulas, the tensor product terms are the result of using the midpoint rule on the integrals $\int_{D_1}(\tilde{\mu}_r\mathbf{H})\cdot\hat{n}dA$ and $\int_{D_2}(\tilde{\varepsilon}_r\mathbf{E})\cdot\hat{n}dA$, where D_1 is a square face orthogonal to $\mathbf{H}$, D_2 is a square face orthogonal to $\mathbf{E}$, and $\hat{n}$ is the unit normal vector. Observing the diagram in Figure 4.1, it is clear that only one component of the electric and magnetic fields is defined at each point. This creates a problem when $\tilde{\mu}_r$ and $\tilde{\varepsilon}_r$ have non-zero off-diagonal entries, as the individual components of $\tilde{\mu}_r\mathbf{H}$ and $\tilde{\varepsilon}_r\mathbf{E}$ will be summations of these terms; a local approximation must be constructed for the field components which are not defined at the midpoints of D_1 and D_2. To this end, the simple schemes introduced in previous works on FDTD methods [92] are used, which take the average of the nearest four field values. The products $\tilde{\mu}_r\mathbf{H}$ and $\tilde{\varepsilon}_r\mathbf{E}$ on the grids are then

$$\begin{aligned}
&(\tilde{\mu}_r\mathbf{H})^x_{i,j+1,k+1} \approx \tilde{\mu}^{xx}_{i,j+1,k+1}H^x_{i,j+1,k+1}\\
&+\tilde{\mu}^{xy}_{i,j+1,k+1}\frac{H^y_{i+1,j,k+1}+H^y_{i+1,j+2,k+1}+H^y_{i-1,j,k+1}+H^y_{i-1,j+2,k+1}}{4}\\
&+\tilde{\mu}^{xz}_{i,j+1,k+1}\frac{H^z_{i+1,j+1,k}+H^z_{i+1,j+1,k+2}+H^z_{i-1,j+1,k}+H^z_{i-1,j+1,k+2}}{4}\\
&(\tilde{\varepsilon}_r\mathbf{E})^x_{i+1,j,k} \approx \tilde{\varepsilon}^{xx}_{i+1,j,k}E^x_{i+1,j,k}\\
&+\tilde{\varepsilon}^{xy}_{i+1,j,k}\frac{E^y_{i,j+1,k}+E^y_{i+2,j+1,k}+E^y_{i,j-1,k}+E^y_{i+2,j-1,k}}{4}\\
&+\tilde{\varepsilon}^{xz}_{i+1,j,k}\frac{E^z_{i,j,k+1}+E^z_{i+2,j,k+1}+E^z_{i,j,k-1}+E^z_{i+2,j,k-1}}{4}
\end{aligned} \tag{4.6}$$

for the x-components; the other components can be defined similarly. Now, these approximations can be inserted into (4.5) to get the finite difference formulas for fully anisotropic media.

4.3 Block tridiagonal structure and node ordering

Observe the finite difference equations given by (4.5) and (4.6), and consider the unknowns on the layer $z = kh$. It is clear that each node on this layer of the grid only interacts with nodes that satisfy $(k-1)h \leq z \leq (k+1)h$. Denoting u_k as the vector of unknowns on the layer $z = kh$, the unknowns within each layer can be ordered lexicographically by the x-coordinate first and y-coordinate second. The full vector of unknowns can be written as $u = (u_1^t, u_2^t, \ldots, u_n^t)^t$, where

$$\begin{aligned}
u_1 &= (H^x_{2,1,1}, H^x_{4,1,1}, ..., H^y_{1,2,1}, E^z_{2,2,1}, ..., H^x_{n-3,n,1}, H^x_{n-1,n,1})^t \\
u_2 &= (H^z_{1,1,2}, E^y_{2,1,2}, ..., E^x_{1,2,2}, E^x_{3,2,2}, ..., E^y_{n-1,n,2}, H^z_{n,n,2})^t \\
&\vdots \\
u_{n-1} &= (H^z_{1,1,n-1}, E^y_{2,1,n-1}, ..., E^x_{1,2,n-1}, E^x_{3,2,n-1}, ..., E^y_{n-1,n,n-1}, H^z_{n,n,n-1})^t \\
u_n &= (H^x_{2,1,n}, H^x_{4,1,n}, ..., H^y_{1,2,n}, E^z_{2,2,n}, ..., H^x_{n-3,n,n}, H^x_{n-1,n,n})^t.
\end{aligned}$$

Similarly, the right hand side f contains the information on the current source and can be written as $f = (f_1^t, f_2^t, \ldots, f_n^t)^t$ following the same ordering.

Given the full vector of unknowns $u = (u_1^t, ..., u_n^t)^t$ and the right hand side $f = (f_1^t, ..., f_n^t)^t$, the linear system $Au = f$ takes the block tridiagonal form

$$\begin{pmatrix} A_{1,1} & A_{1,2} & & \\ A_{2,1} & A_{2,2} & \ddots & \\ & \ddots & \ddots & A_{n-1,n} \\ & & A_{n,n-1} & A_{n,n} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}. \tag{4.7}$$

It is important to note that the off-diagonal blocks here are not square; because of the staggered grid, there is a slightly different number of unknowns in each layer.

4.4 Preconditioners for the Yee Grid

In [35], the sweeping preconditioner with moving PML was developed for the scalar Helmholtz equation with variable media. Here, the preconditioner is reviewed and adapted to Maxwell's equations and the Yee grid.

After arriving at (4.7), the discussion of the sweeping factorization can begin. Let P_k be the unknowns on the k-th layer. By eliminating the unknowns layer by layer, the block LDL^t factorization for the matrix A can be written

$$A = L_1 ... L_{n-1} \begin{pmatrix} S_1 & & & \\ & S_2 & & \\ & & \ddots & \\ & & & S_n \end{pmatrix} L_{n-1}^t ... L_1^t, \tag{4.8}$$

where $S_1 = A_{1,1}$, $S_m = A_{m,m} - A_{m,m-1} S_{m-1}^{-1} A_{m-1,m}$ for $m = 2, ..., n$, and L_k are the block lower triangular matrices given by

$$L_k(P_{k+1}, P_k) = A_{k+1,k} S_k^{-1}, \quad L_k(P_i, P_i) = I \quad (1 \le i \le n), \quad \text{zero otherwise.}$$

Inverting this factorization and applying it to the right hand side f, the solution is

$$u = (L_1^t)^{-1} ... (L_{n-1}^t)^{-1} \begin{pmatrix} S_1^{-1} & & & \\ & S_2^{-1} & & \\ & & \ddots & \\ & & & S_n^{-1} \end{pmatrix} L_{n-1}^{-1} ... L_1^{-1} f. \tag{4.9}$$

The goal is to find an approximate inverse M^{-1} efficiently and solve the preconditioned system $M^{-1}Au = M^{-1}f$ iteratively. Here, the main computational task is constructing the inverse operators of $S_1, ..., S_n$, as these matrices are dense; this problem will be addressed shortly.

Before an accurate approximation for the inversion of the Schur complement matrices can be made, it is important to gain some physical intuition by restricting the problem to the first m layers. Consider the upper $m \times m$ blocks of the block tridiagonal matrix A; if only the degrees of freedom for layers $1, ..., m$ are considered, where layer m is outside the PML, then the relevant linear system is still block tridiagonal, i.e.

$$\begin{pmatrix} A_{1,1} & A_{1,2} & & \\ A_{2,1} & A_{2,2} & \ddots & \\ & \ddots & \ddots & A_{m-1,m} \\ & & A_{m,m-1} & A_{m,m} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{pmatrix}. \tag{4.10}$$

Removing the degrees of freedom for layers $m + 1$ to n essentially strips the domain of these layers and enforces PEC boundary conditions at layer $m + 1$; that is, the matrix equation (4.10) corresponds to the discretization of the half-space PEC plane problem for Maxwell's equations on layers 1 to m, where the PEC plane is located at layer $m + 1$. If the inverse of the operator on the left hand side is taken, then it takes the form

$$A^{-1} = (L_1^t)^{-1} ... (L_{m-1}^t)^{-1} \begin{pmatrix} S_1^{-1} & & \\ & \ddots & \\ & & S_m^{-1} \end{pmatrix} L_{m-1}^{-1} ... L_1^{-1}. \tag{4.11}$$

The inversion formula above is similar in structure to the inverse of the full matrix A; this time, however, only the lower triangular matrices $L_1, \dots, L_{m-1}$ and Schur complements $S_1, \dots, S_m$ are necessary. The left hand side in equation (4.11) is the Green's function for the half-space problem, a dense matrix which can be written in block form as

$$\begin{pmatrix} A_{1,1} & A_{1,2} & & \\ A_{2,1} & A_{2,2} & \ddots & \\ & \ddots & \ddots & A_{m-1,m} \\ & & A_{m,m-1} & A_{m,m} \end{pmatrix}^{-1} = \begin{pmatrix} G_{1,1} & G_{1,2} & \dots & G_{1,m} \\ G_{2,1} & G_{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ G_{m,1} & \dots & \dots & G_{m,m} \end{pmatrix}, \quad (4.12)$$

where the entries in each block $G_{i,j}$ give the fields in layer i due to sources in layer j. The crucial observation here is that S_m^{-1} remains untouched by the left operator $(L_1^t)^{-1}...\ (L_{m-1}^t)^{-1}$ and right operator $L_{m-1}^{-1}...L_1^{-1}$, due to the definition of $L_1, ..., L_{m-1}$; that is, if the matrix multiplications on the right hand side of (4.11) are carried out, then the matrix in the (m, m)-th block is just S_m^{-1}. Algebraically, this gives the result $G_{m,m} = S_m^{-1}$. The physical knowledge gained here is that S_m^{-1} *is approximately the discrete half-space Green's function for Maxwell's equations with PEC boundary conditions on layer* $m + 1$*, restricted to degrees of freedom on the* m*-th layer*. By solving the half-space problem on a grid of the first m layers, an operator which reproduces S_m^{-1} can be constructed. However, this becomes very costly as m approaches n, so the goal is to approximate S_m^{-1} efficiently. The authors of [34, 35] have proposed two methods for this approximation: the hierarchical matrix approach and the moving PML approach. Here, the latter approach is taken.

4.5 Moving PML method

The application of S_m^{-1} only involves the degrees of freedom on the m-th layer; that is, the half space Green's function matrix which will be approximated ($G_{m,m}$) only maps the right hand side on layer m to the solution on layer m. Therefore, the solution of the half-space problem only needs to be accurate in a small neighborhood of $z = mh$. Recall that the purpose of the PML is to be an absorbing boundary; in the original problem, the fields outside the cube $[\ell, 1-\ell]^3$ are of little interest, so a PML was placed at the boundary to make the computational domain to be $[0,1]^3$. The same exact reasoning can be used to truncate each half-space problem; if only the layers in the immediate vicinity of the m-th layer are of importance, then layers 1 to $m-1$ can be treated as buffer layers or "white space." Thus, the PML can be pushed up to the edge of the domain of interest and still reproduce a good approximation of the solution on layer m. The computational advantage here is that the subproblem for each S_m^{-1} is much smaller than the full half-space problem. The technique of truncating the domain and pushing the PML closer is called the *moving PML method.*

To be more precise, let $b = \frac{\ell}{h}$ be the number of PML layers and consider the domain $\Omega_m = [0,1] \times [0,1] \times [(m-b)h, (m+1)h]$. Define also the shifted PML function

$$s_z^m(z) = 1 + \imath\sigma(z - (m-b)h), \tag{4.13}$$

and shifted material tensors

$$\tilde{\varepsilon}_{r,m} = \begin{pmatrix} \varepsilon_{xx}\frac{s_y s_z^m}{s_x} & \varepsilon_{xy}s_z^m & \varepsilon_{xz}s_y \\ \varepsilon_{yx}s_z^m & \varepsilon_{yy}\frac{s_x s_z^m}{s_y} & \varepsilon_{yz}s_x \\ \varepsilon_{zx}s_y & \varepsilon_{zy}s_x & \varepsilon_{zz}\frac{s_x s_y}{s_z^m} \end{pmatrix}, \quad \tilde{\mu}_{r,m} = \begin{pmatrix} \mu_{xx}\frac{s_y s_z^m}{s_x} & \mu_{xy}s_z^m & \mu_{xz}s_y \\ \mu_{yx}s_z^m & \mu_{yy}\frac{s_x s_z^m}{s_y} & \mu_{yz}s_x \\ \mu_{zx}s_y & \mu_{zy}s_x & \mu_{zz}\frac{s_x s_y}{s_z^m} \end{pmatrix}.$$

With the PML pushed to the edge, the subproblem to be solved for each S_m^{-1}

is

$$\left.\begin{aligned} \nabla \times \mathbf{E} &= -\imath\omega\mu_0\tilde{\mu}_{r,m}\mathbf{H} \\ \nabla \times \mathbf{H} &= \imath\omega\varepsilon_0\tilde{\varepsilon}_{r,m}\mathbf{E} + \mathbf{J} \\ \nabla \cdot \big(\varepsilon_0\tilde{\varepsilon}_{r,m}\mathbf{E}\big) &= q_e \\ \nabla \cdot \big(\mu_0\tilde{\mu}_{r,m}\mathbf{H}\big) &= 0 \end{aligned}\right\} \quad \text{in } \Omega_m \tag{4.14}$$

$$\left.\begin{aligned} \hat{\mathbf{n}} \times \mathbf{E} &= 0 \\ \hat{\mathbf{n}} \times \mathbf{H} &= 0 \end{aligned}\right\} \quad \text{on } \partial\Omega_m \tag{4.15}$$

and the subgrid on which the above equations are discretized is

$$G_m = \{(ih, jh, kh) \quad | \quad 1 \leq i, j \leq n, m - b + 1 \leq k \leq m\}. \tag{4.16}$$

To solve each discretized subproblem, a version of the multifrontal method [62, 35] is utilized, with a modification that allows the Yee grid to be handled naturally. Consider the matrix H_m resulting from the discretization of (4.14) on G_m. Essentially, each subproblem can be viewed as a quasi-2D problem, since the number of PML and buffer layers is small. The first step of the algorithm is to partition the nodes of G_m hierarchically in the x-y plane, i.e. nodes with the same x and y indices remain in the same group. In the Helmholtz case, every node is associated with a variable; in the Maxwell case, however, the Yee grid does not have an unknown assigned to every node. To keep the same efficiency of the method with the Helmholtz grid, these empty nodes are left in at the hierarchical partitioning stage; after each cluster is set, any empty nodes are removed, as they contain no information relevant to the factorization. The unknowns associated with the full nodes are then reordered according to their hierarchical groups to minimize the number of fill-ins in the LDL^t factorization of H_m. The costs of computing the factorization and applying to a vector are $O(b^3n^3)$ and $O(b^2n^2 \log n)$, respectively; the previous works for can be referred to for details.

Once the multifrontal factorization of H_m is constructed, the approximate application of S_m^{-1} to a vector is as follows. Given a vector of values g_m defined on the grid points of layer m, a longer vector padded with zeros is constructed, which will correspond to the grid points on layers $m - b$ to $m - 1$; the zeros are necessary to ensure that the solution is not corrupted by the Green's function on these layers. After the vector is made long enough to match the dimensions of H_m, the matrix-vector product

$$H_m^{-1} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_m \end{pmatrix} = \begin{pmatrix} * \\ \vdots \\ * \\ v_m \end{pmatrix} \tag{4.17}$$

is computed and the vector v_m can be extracted; this will give the approximation of $S_m^{-1} g_m = v_m$ necessary. This combined process of concatenation, application of H_m^{-1}, and extraction is denoted as the operator $\tilde{T}_m : g_m \to v_m$.

The moving PML preconditioner can now be summarized in two stages: the construction of the approximate sweeping factorization, and the application to an arbitrary vector. Staying consistent to [35], the vector u_F is defined as

$$u_F = (u_1^t, ..., u_b^t), \quad f_F = (f_1^t, ..., f_b^t), \tag{4.18}$$

while A is rewritten as

$$\begin{pmatrix} A_{F,F} & A_{F,b+1} & & \\ A_{b+1,F} & A_{b+1,b+1} & \ddots & \\ & \ddots & \ddots & A_{n-1,n} \\ & & A_{n,n-1} & A_{n,n} \end{pmatrix} \begin{pmatrix} u_F \\ u_{b+1} \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} f_F \\ f_{b+1} \\ \vdots \\ f_n \end{pmatrix}, \tag{4.19}$$

where $A_{F,F}$ is the upper left block of A for the first b layers.

Algorithm 4.5.1. *Construct the approximate sweeping factorization of A.*

```
1. Let $G_F$ be the subgrid of the first $b$ layers and $H_F = A_{F,F}$; construct the
   multifrontal factorization of $H_F$.
2.   for $m = b+1, ..., n$ do
         Let $G_m$ be as defined in (4.16) and $H_m$ be the matrix resulting
         from the finite difference discretization of (4.14). Construct the
         multifrontal factorization of $H_m$.
     end for
```

Algorithm 4.5.2. *Apply the approximate inverse to get $u \approx A^{-1}f$.*

```
1. $u_F = f_F$ and $u_m = f_m$ for $m = b+1, ..., n$.
2. $u_{b+1} = u_{b+1} - A_{b+1,F}(H_F^{-1}u_F)$, using the multifrontal factorization of $H_F$.
3.   for $m = b+1, ..., n-1$ do
         $u_{m+1} = u_{m+1} - A_{m+1,m}(\tilde{T}_m u_m)$, where $\tilde{T}_m u_m$ is computed by the
         process described earlier in section 4.5.
     end for
```

```
4. $u_F = H_F^{-1} u_F$, using the multifrontal factorization of $H_F$.
5.    for $m = b+1, ..., n$ do
          $u_m = \tilde{T}_m u_m$. See previous steps for the application of $\tilde{T}_m$.
      end for
6.    for $m = n-1, ..., b+1$ do
          $u_m = u_m - \tilde{T}_m(A_{m,m+1} u_{m+1})$. See previous steps for the application
          of $\tilde{T}_m$.
      end for
7. $u_F = u_F - H_F^{-1}(A_{F,b+1} u_{b+1})$, using the multifrontal factorization of $H_F$.
```

Because the total number of degrees of freedom is $N = n^3$ and there are n layers, it is clear that the cost of algorithms 4.5.1 and 4.5.2 are $O(b^3 n^4) = O(b^3 N^{4/3})$ and $O(b^2 n^3 \log n) = O(b^2 N \log N)$, respectively.

In practice, the approach is slightly different. First, the block LDL^t factorization is constructed so that multiple layers can be preconditioned for each subproblem; that is, S_m^{-1} is not just applied to layer m, but layers $m-d+1$ to m, where d is a specified number of buffer layers. The subproblem for each S_m is instead defined on an expanded domain $\Omega'_m = [0,1] \times [0,1] \times [(m-b-d+1)h, (m+1)h]$. Secondly, the perturbed matrix associated with the curl equations

$$\nabla \times \mathbf{E} = -\imath(\omega + \imath\alpha)\mu_0\tilde{\mu}_r\mathbf{H}, \qquad \nabla \times \mathbf{H} = \imath(\omega + \imath\alpha)\varepsilon_0\tilde{\varepsilon}_r\mathbf{E} + \mathbf{J} \tag{4.20}$$

is used instead of the true matrix A, for some small damping constant α; the preconditioner produced from this matrix is much more stable and effective. The approximate inverse operator constructed by algorithm 4.5.1 with constant α is denoted as M_α^{-1}; this approximate inverse is used as a left preconditioner, and the preconditioned linear system to be solved is

$$M_\alpha^{-1} A u = M_\alpha^{-1} f. \tag{4.21}$$

4.6 Numerical results

In this section, some preliminary results are provided illustrating the effectiveness of the sweeping preconditioner for the Yee grid. The general setup of the problems is as follows. A point source oriented in the z-direction with current magnitude $|\mathbf{J}|$=1 is embedded in the material domain at $(0.5, 0.25, 0.5)$. As mentioned before, the domain is the unit cube $[0,1]^3$, and has a PEC boundary. K is the size of the domain in terms of wavelengths, so $\lambda = \frac{1}{K}$. For the Yee grid, the number of points per wavelength is set at 6 to attain reasonably accurate results; this implies that field components of the same kind are separated by $\frac{\lambda}{6}$, but the distance between two nodes in the physical grid is $h = \frac{\lambda}{12}$. The PML width is chosent to be $\ell = \frac{\lambda}{2}$, and the number of buffer layers is $d = 12$. For the damping constant, α is set to 1. As for the iterative solver, GMRES iteration with a relative residual tolerance of $1e-3$ is used. For ease of implementation and generating figures, all of the code is run serially in MATLAB.

Three example mediums were chosen. The first example is the converging lens with a Gaussian profile centered at $\mathbf{r}_c = (0.5, 0.5, 0.5)$ for both permittivity and permeability; that is, the material is isotropic, and

$$\varepsilon_r(\mathbf{r}) = \frac{1}{\frac{4}{3}\left(1 - \frac{1}{2}e^{-32|\mathbf{r}-\mathbf{r}_c|^2}\right)}, \quad \mu_r(\mathbf{r}) = \frac{1}{\frac{4}{3}\left(1 - \frac{1}{2}e^{-32|\mathbf{r}-\mathbf{r}_c|^2}\right)}.$$

The second example is a random isotropic medium, which is formed with a random smooth perturbation function $\delta(\mathbf{r})$ satisfying $0 < \delta < 1$; once again, the material tensors are formed by multiplying the identity tensor with scalar functions

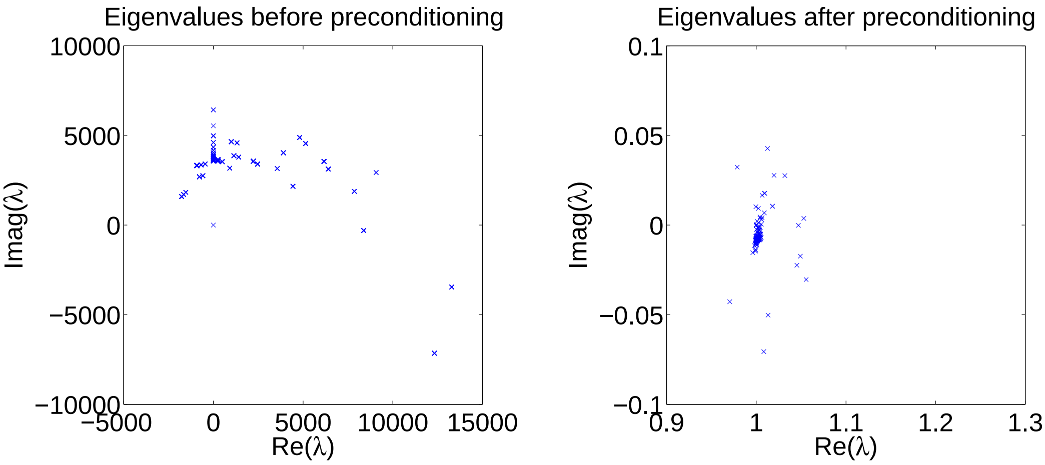
$$\varepsilon_r(\mathbf{r}) = \frac{4}{3}\left(1 - \frac{1}{2}\delta(\mathbf{r})\right), \quad \mu_r(\mathbf{r}) = \frac{4}{3}\left(1 - \frac{1}{2}\delta(\mathbf{r})\right).$$

Finally, for an anisotropic, inhomogeneous example, the medium

$$\varepsilon_r(\mathbf{r}) = \begin{pmatrix} 1.1 & 0.1\imath & 0 \\ -0.1\imath & 0.9 & 0 \\ 0 & 0 & \gamma(\mathbf{r}) \end{pmatrix}, \quad \mu_r(\mathbf{r}) = \begin{pmatrix} 1.1 & 0.1\imath & 0 \\ -0.1\imath & 0.9 & 0 \\ 0 & 0 & \gamma(\mathbf{r}) \end{pmatrix}.$$

is constructed, where γ is a smooth random perturbation function satisfying $0.7 < \gamma < 1.3$. This medium is similar to gyrotropic mediums found in crystals, except the z-component of the tensor is randomized.

To start, the preconditioner is tested to see how it affects the eigenvalues and condition numbers of the original matrix. Figure 4.2 shows that the original problem is indefinite and very ill-conditioned; a plot of the eigenvalues for the converging lens media illustrates that the original matrix A has large eigenvalues which can have a negative real part, as well as eigenvalues close to the origin. After preconditioning, the spectrum is clustered around $(1, 0)$ in the positive real half of the complex plane, which is conducive for the convergence of GMRES. Using MATLAB's condition number estimator `condest`, the 1-norm

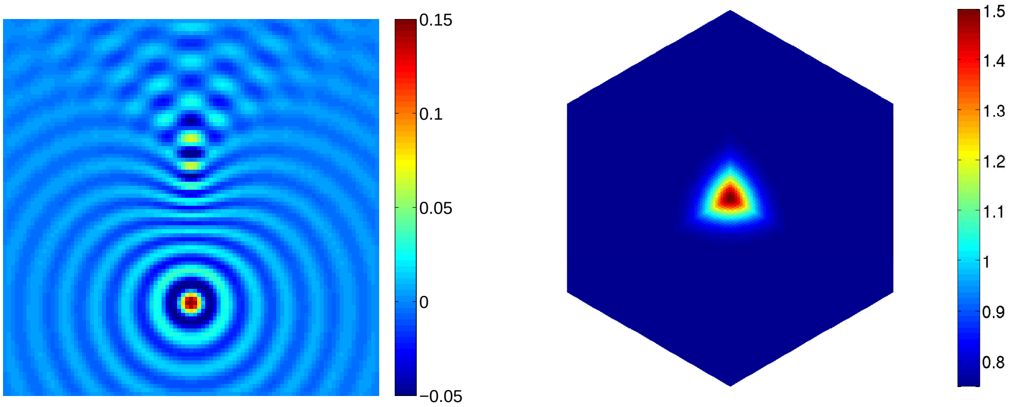


Medium	cond(A)	cond$(M_\alpha^{-1}A)$
Converging lens	1.875e+07	42.407
Random isotropic	2.959e+07	22.729
Random anisotropic	3.305e+07	11.841

Figure 4.2: Eigenvalues of the coefficient matrix A in the converging lens problem, before and after preconditioning, along with condition number estimates for different media.

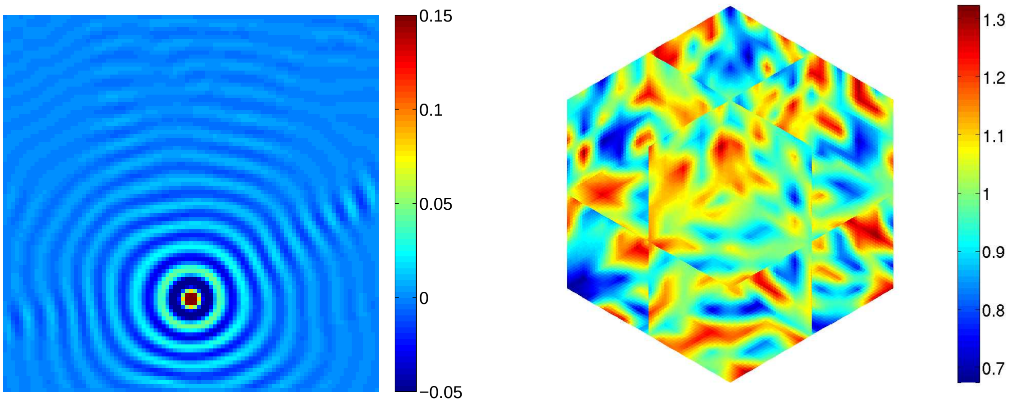
condition numbers for small problems $(K = 2)$ are estimated before and after preconditioning; the method reduces the condition number by several orders of magnitude.

For each medium, the E_z component in the x-y plane at $z = 0.5$ is plotted; because the source is aligned in the z-direction, the most significant behavior in the isotropic case happens in this component. Figures 4.3, 4.4, and 4.5 give the numerical results for each particular example. To illustrate the characteristics of the material, the relative material parameters for the isotropic cases and ε_{zz} for the anisotropic example are shown. The tables under each plot list the size of the problem K, number of unknowns N, preconditioner



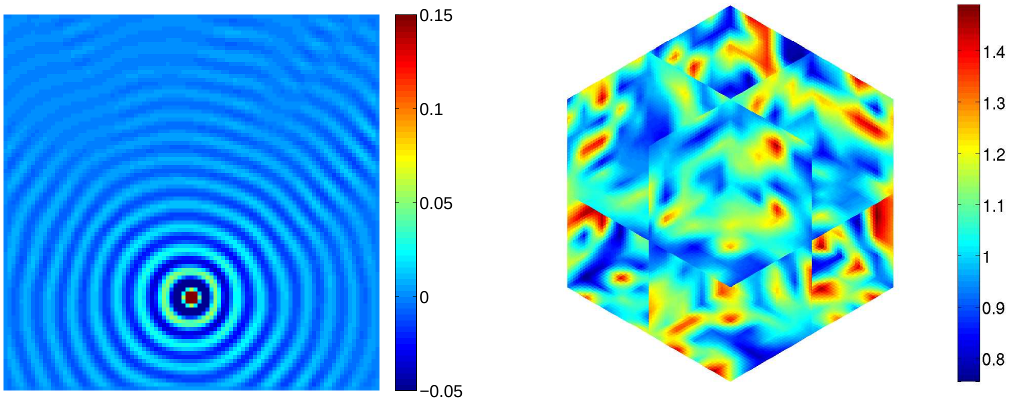
K	N	T_{setup}	T_{solve}	N_{iter}
4	7.783e+4	24	8	4
8	6.429e+5	303	84	4
16	5.225e+6	3938	851	4

Figure 4.3: Converging lens example. Left: $Re(E_z)$ in the x-y plane at $z = 0.5$. Right: ε_r and μ_r in the medium.



K	N	T_{setup}	T_{solve}	N_{iter}
4	7.783e+4	23	8	4
8	6.429e+5	301	105	5
16	5.225e+6	3946	1064	5

Figure 4.4: Random isotropic media example. Left: $Re(E_z)$ in the x-y plane at $z = 0.5$. Right: ε_r and μ_r in the medium.



K	N	T_{setup}	T_{solve}	N_{iter}
4	7.783e+4	24	8	4
8	6.429e+5	305	85	4
16	5.225e+6	3942	859	4

Figure 4.5: Random anisotropic media example. Left: $Re(E_z)$ in the x-y plane at $z = 0.5$. Right: ε_{zz} in the medium.

setup time T_{setup}, iterative solver time T_{solve}, and number of iterations N_{iter}. It is observed that even if the size of the problem is increased, the number of iterations remains almost constant. The preconditioner takes the same time for the anisotropic medium as it does for the isotropic mediums; thus, there is no dependency on the type of medium for the method to work.

Next, the accuracy of the preconditioning method is tested by running GMRES to a smaller residual tolerance and comparing with the numerical solution of the linear system using MATLAB's backslash operator; in these examples, the tolerance is set at $1e-6$. The relative 2-norm error between the direct solver and iterative solver result is ϵ. For the converging lens problem, table 4.6 shows that the preconditioner agrees with numerically stable direct

K	N	N_{iter}	ϵ
4	7.783e+4	7	2.4572e-7
8	6.429e+5	7	3.5560e-7
16	5.225e+6	8	4.3398e-7

Figure 4.6: Accuracy of the preconditioning method for the converging lens problem.

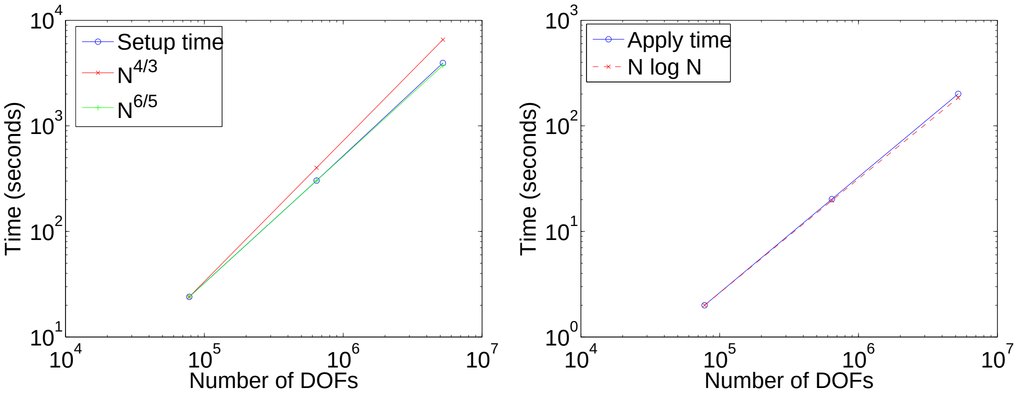


Figure 4.7: Setup and apply times plotted against the number of DOFs.

solvers, as the relative error does not exceed the given residual tolerance. It is significant that although the residual tolerance has decreased, the number of iterations does not grow significantly.

Finally, to show the complexities for the setup and apply stages, the setup time and apply time have been plotted against the total number of degrees of freedom. Figure 4.7 illustrates the almost linear complexity of the sweeping preconditioner. Since the number of gridpoints per wavelength is kept constant, each time the frequency is doubled, the total number of DOFs should increase by a factor of 8; this implies that the setup time should increase by a factor of $8^{4/3} = 16$. However, an increase by a factor of 12 or 13 is usually observed; this trends with an $O(N^{6/5})$ complexity instead.

Chapter 5

Preconditioners for Maxwell's Equations: Nédélec elements

5.1 Variational Formulations

Consider the PML-truncated Maxwell problem (4.5). By multiplying the first equation with μ_r^{-1} and then operating with $\nabla\times$, the magnetic field variable can be eliminated; the boundary value problem now is to solve

$$\begin{aligned} \nabla \times \tilde{\mu}_r^{-1} \nabla \times \mathbf{E} - \kappa^2 \tilde{\varepsilon}_r \mathbf{E} &= -\imath\omega\mu_0 \mathbf{J} \qquad \text{in } \Omega, & (5.1)\\ \hat{\mathbf{n}} \times \mathbf{E} &= 0 & (5.2)\\ \hat{\mathbf{n}} \times (\tilde{\mu}_r^{-1} \nabla \times \mathbf{E}) &= 0 \qquad \text{on } \partial\Omega. & (5.3) \end{aligned}$$

By multiplying with a test function $\boldsymbol{\phi} \in H_0(\text{curl}, \Omega)$ and integrating by parts, the weak form of the PDE is obtained; that is,

$$(\tilde{\mu}_r^{-1} \nabla \times \mathbf{E}, \nabla \times \boldsymbol{\phi})_\Omega - \kappa^2 (\tilde{\varepsilon}_r \mathbf{E}, \boldsymbol{\phi})_\Omega = -\imath\omega\mu_0 (\mathbf{J}, \boldsymbol{\phi})_\Omega, \tag{5.4}$$

with the standard inner product being $(\mathbf{u}, \mathbf{v}) = \int_\Omega \mathbf{u} \cdot \bar{\mathbf{v}} dV$. For the space $X = H_0(\text{curl}, \Omega)$, define the sesquilinear form $B : X \times X \to \mathbb{C}$ and linear functional $F : X \to \mathbb{C}$ as

$$\begin{aligned} B(\mathbf{E}, \boldsymbol{\phi}) &= (\tilde{\mu}_r^{-1} \nabla \times \mathbf{E}, \nabla \times \boldsymbol{\phi})_\Omega - \kappa^2 (\tilde{\varepsilon}_r \mathbf{E}, \boldsymbol{\phi})_\Omega & (5.5)\\ F(\boldsymbol{\phi}) &= -\imath\omega\mu_0 (\mathbf{J}, \boldsymbol{\phi})_\Omega. & (5.6) \end{aligned}$$

The problem of solving Maxwell's equations can be rephrased as the following: find the electric field $\mathbf{E} \in X$ such that

$$B(\mathbf{E}, \boldsymbol{\phi}) = F(\boldsymbol{\phi}), \qquad \forall \boldsymbol{\phi} \in X. \tag{5.7}$$

Proving existence and uniqueness for the indefinite Maxwell problem is more difficult than for positive-definite elliptic PDEs; because of the $-\omega^2$ term, the sesquilinear form is not coercive, which is clear for large wavenumbers. Thus, the Lax-Milgram lemma is not directly applicable. In addition, the curl operator contains a large null space in $H(\mathrm{curl}, \Omega)$, which needs to be removed using the Helmholtz decomposition. The reader can refer to [67] for the details.

5.2 Nédélec elements

For Maxwell's equations, curl-conforming basis functions are necessary so that the resulting fields satisfy the divergence conditions, which negate the problem of "spurious solutions" [58]. To this end, the low-order edge elements introduced by Whitney [93] and Nédélec [69] are most popular in the CEM community. Consider the standard $p = 1$ nodal basis functions ϕ_i defined at the vertices $\mathbf{v}_i$ for $i = 1, \ldots, d+1$ of each triangular/tetrahedral element; these functions satisfy $\phi_i(\mathbf{v}_j) = \delta_{ij}$. The first order Whitney form is defined as

$$\boldsymbol{\psi}_{i,j} = (\phi_i \nabla \phi_j - \phi_j \nabla \phi_i)\ell_{i,j}, \tag{5.8}$$

where $\ell_{i,j}$ is the length of the edge between vertex i and vertex j. This term is necessary to normalize the function and make it dimensionless. The resulting function $\boldsymbol{\psi}_{i,j}$ has the properties

$$\begin{aligned} \nabla \cdot \boldsymbol{\psi}_{i,j} &= \nabla \cdot (\phi_i \nabla \phi_j)\ell_{i,j} - \nabla \cdot (\phi_j \nabla \phi_i)\ell_{i,j} = 0, & (5.9) \\ \nabla \times \boldsymbol{\psi}_{i,j} &= 2\nabla \phi_i \times \nabla \phi_j \ell_{i,j}. & (5.10) \end{aligned}$$

If $\mathbf{e}_{i,j}$ is the unit vector in the direction from vertex i to vertex j, then $\mathbf{e}_{i,j} \cdot \boldsymbol{\psi}_{i,j} = 1$; that is, the tangential component of the basis function along its prescribed edge is constant. On the contrary, the tangential component of the function along its complementary edges is 0.

Given a quasi-uniform tetrahedral mesh of Ω with edge lengths bounded by h, with a little abuse of notation the finite element approximation can be written as

$$\mathbf{E}_h = \sum_{i=1}^{N} c_i \boldsymbol{\psi}_i, \tag{5.11}$$

where $\boldsymbol{\psi}_i$ is the edge function defined on edge i, N is the total number of degrees of freedom (interior edges in this case), and $c_i \in \mathbb{C}$ are the undetermined coefficients. Using the same space for both trial and test functions, the Galerkin method yields the linear system

$$Ax = b, \tag{5.12}$$

with the matrix and vector entries $A_{ij} = B(\boldsymbol{\psi}_i, \boldsymbol{\psi}_j)$, $x_j = c_j$, and $b_i = F(\boldsymbol{\psi}_i)$.

5.3 Preconditioners for Unstructured Meshes

The preconditioner for finite elements on an unstructured mesh is similar to the uniform finite difference case; however, there are some key differences which need to be addressed, so some parts of the overall algorithm will be reviewed to highlight their context.

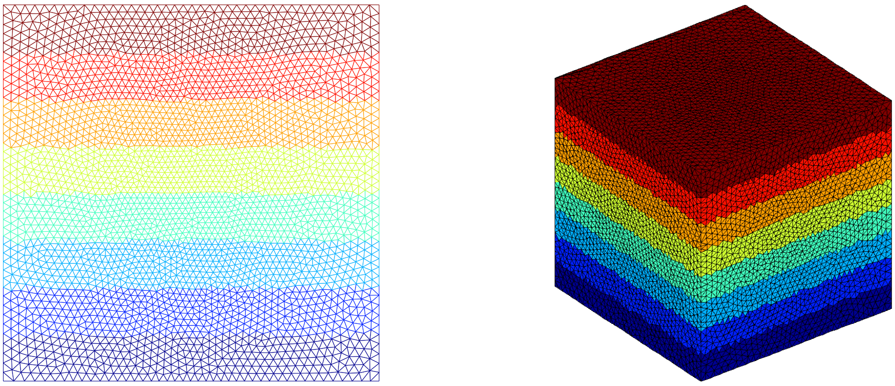


Figure 5.1: Left: triangular mesh partitioned into eight layers. Right: tetrahedral mesh partitioned into eight layers.

In the unstructured mesh case, the first step to setting up the preconditioner is to divide the mesh into layers or slabs. Consider the problem for a particular wavenumber κ such that the number $K := \frac{\kappa}{\pi} = \frac{\omega\sqrt{\mu_0\varepsilon_0}}{\pi}$ is an integer; here, K is the width of the domain in wavelengths. Let it be assumed for the sake of simplicity that the PML width is $\ell = \lambda$. The domain can then be divided into the subdomains Ω_i, $i = 1, \dots, K$ as

$$\begin{aligned} \Omega_i &= [-1,1]^{d-1} \times [-1+(i-1)\lambda, -1+i\lambda), \quad \text{for} \quad i = 1, \dots, K-1 \\ \Omega_K &= [-1,1]^{d-1} \times [-1+(K-1)\lambda, 1]. \end{aligned}$$

The partition occurs in the y-direction for 2D problems and in the z-direction for 3D problems. It is clear that $\bar{\Omega} = \cup_{i=1}^{K} \Omega_i$ and $\Omega_i \cap \Omega_j = \emptyset$ if $i \neq j$.

For a tetrahedral mesh $\mathcal{T} = \{t_1, \dots, t_{N_{\mathcal{T}}}\}$ with edges denoted by e_j,

$j = 1, \dots, N$, define $\mathbf{v}_{j_k}$ for $k = 1, \dots, d+1$ to be the vertices of tetrahedron t_j in d dimensions and denote the centroid of t_j with $\mathbf{c}_j = \frac{1}{n+1} \sum_{k=1}^{n+1} \mathbf{v}_{j_k}$. Next, define $\mathcal{T}_i$ as the union of tetrahedra whose centroids are in Ω_i, i.e.,

$$\mathcal{T}_i = \cup\{t_j : \mathbf{c}_j \in \Omega_i\}. \tag{5.13}$$

Clearly, $\bar{\Omega} = \cup_{i=1}^{K} \mathcal{T}_i$. Figure 5.1 shows an unstructured triangular mesh and tetrahedral mesh partitioned into eight layers, with a different color for each layer; in general, the boundary of each layer is not a smooth surface.

Once the task of partitioning the mesh is completed, the integer sets $\mathcal{E}_i$ can be constructed; these sets will point to the degrees of freedom which are associated with layer Ω_i. This organizational structure is necessary to obtain the block LDL^t factorization for the sweeping factorization. At first, this may seem trivially similar to the previous algorithm, but a conflict occurs at the boundary between two layers; specifically, when a simplex in $\mathcal{T}_i$ and a simplex in $\mathcal{T}_{i-1}$ share an edge. This problem can be remedied by always choosing to associate boundary edges with the upper layer. If $\partial\mathcal{T}_i$ is the boundary and $\mathcal{T}_i^{\text{int}}$ is the interior of the domain defined by elements in $\mathcal{T}_i$, such that $\mathcal{T}_i = \partial\mathcal{T}_i \cup \mathcal{T}_i^{\text{int}}$, the edges e_j, $j = 1, \dots, N$ can be categorized in the following manner:

$$\begin{aligned} \mathcal{E}_1 &= \{j : e_j \text{ is an interior edge of } T_1\} \\ \mathcal{E}_i &= \{j : e_j \text{ is an interior edge of } T_i \text{ or } e_j \in \partial\mathcal{T}_i \cap \partial\mathcal{T}_{i-1}\} \text{ for } i = 2, \dots, K. \end{aligned}$$

With the degrees of freedom in each layer defined by the integer sets $\mathcal{E}_i$ for $i = 1, \dots, K$, the sparse linear system can be written in block tridiagonal form. Using MATLAB-style notation for indexing matrices and vectors, the system can be reordered as

$$\begin{pmatrix} A(\mathcal{E}_1,\mathcal{E}_1) & A(\mathcal{E}_1,\mathcal{E}_2) & & \\ A(\mathcal{E}_2,\mathcal{E}_1) & A(\mathcal{E}_2,\mathcal{E}_2) & \ddots & \\ & \ddots & \ddots & A(\mathcal{E}_{K-1},\mathcal{E}_K) \\ & & A(\mathcal{E}_K,\mathcal{E}_{K-1}) & A(\mathcal{E}_K,\mathcal{E}_K) \end{pmatrix} \begin{pmatrix} x(\mathcal{E}_1) \\ x(\mathcal{E}_2) \\ \vdots \\ x(\mathcal{E}_K) \end{pmatrix} = \begin{pmatrix} b(\mathcal{E}_1) \\ b(\mathcal{E}_2) \\ \vdots \\ b(\mathcal{E}_K) \end{pmatrix}, \tag{5.14}$$

where $x(\mathcal{E}_i)$ are the unknown coefficients associated with degrees of freedom in layer i, $b(\mathcal{E}_i)$ is the right hand side computed from basis functions defined in layer i, and $A(\mathcal{E}_i,\mathcal{E}_j)$ are the blocks of the stiffness matrix corresponding to the degrees of freedom in layer i and layer j. This permits the block LDL^t factorization

$$L_1 \dots L_{K-1} \begin{pmatrix} S_1 & & & \\ & S_2 & & \\ & & \ddots & \\ & & & S_K \end{pmatrix} L_{K-1}^t \dots L_1^t, \tag{5.15}$$

where the Schur complement matrices take the form $S_1 = A(\mathcal{E}_1,\mathcal{E}_1)$, $S_i = A(\mathcal{E}_i,\mathcal{E}_i) - A(\mathcal{E}_i,\mathcal{E}_{i-1})S_{i-1}^{-1}A(\mathcal{E}_{i-1},\mathcal{E}_i)$ for $i = 2,\dots,K$. Define the index sets $\mathcal{P}_i$ for $i = 1,\dots,K$ as

$$\mathcal{P}_i = \left\{ \sum_{s=1}^{i-1} |\mathcal{E}_s| + 1, \dots, \sum_{s=1}^{i} |\mathcal{E}_s| \right\}, \tag{5.16}$$

where $|\mathcal{E}_i|$ is the cardinality of set $\mathcal{E}_i$. The block lower triangular matrices L_i are then

$$L_i(\mathcal{P}_{i+1},\mathcal{P}_i) = A(\mathcal{E}_{i+1},\mathcal{E}_i)S_i^{-1},\ L_i(\mathcal{P}_i,\mathcal{P}_i) = I\ (1 \le i \le K),\ \text{zero otherwise.}$$

Explicitly inverting the factorization yields the solution

$$\begin{pmatrix} x(\mathcal{E}_1) \\ x(\mathcal{E}_2) \\ \vdots \\ x(\mathcal{E}_K) \end{pmatrix} = (L_1^t)^{-1} \dots (L_{K-1}^t)^{-1} \begin{pmatrix} S_1^{-1} & & & \\ & S_2^{-1} & & \\ & & \ddots & \\ & & & S_K^{-1} \end{pmatrix} L_{K-1}^{-1} \dots L_1^{-1} \begin{pmatrix} b(\mathcal{E}_1) \\ b(\mathcal{E}_2) \\ \vdots \\ b(\mathcal{E}_K) \end{pmatrix}.$$

The factorization and inversion process are similar to the previous chapter, and can be summarized in the following algorithms.

Algorithm 5.3.1. *Construct the sweeping factorization of A.*

```
Set $S_1 = A(\mathcal{E}_1, \mathcal{E}_1)$ and compute $S_1^{-1}$.
for $i = 2, \dots, K$ do
    Set $S_i = A(\mathcal{E}_i, \mathcal{E}_i) - A(\mathcal{E}_i, \mathcal{E}_{i-1}) S_{i-1}^{-1} A(\mathcal{E}_{i-1}, \mathcal{E}_i)$ and compute $S_i^{-1}$.
end for
```

Algorithm 5.3.2. *Apply the inverse to get $x = A^{-1}b$.*

```
Set $x(\mathcal{E}_i) = b(\mathcal{E}_i)$, for $i = 1, \dots, K$.
Compute $x(\mathcal{E}_2) = x(\mathcal{E}_2) - A(\mathcal{E}_2, \mathcal{E}_1) S_1^{-1} x(\mathcal{E}_1)$.
for $i = 2, \dots, K-1$ do
    Compute $x(\mathcal{E}_{i+1}) = x(\mathcal{E}_{i+1}) - A(\mathcal{E}_{i+1}, \mathcal{E}_i) S_i^{-1} x(\mathcal{E}_i)$.
end for
Compute $x(\mathcal{E}_1) = S_1^{-1} x(\mathcal{E}_1)$.
for $i = 2, \dots, K$ do
    Compute $x(\mathcal{E}_i) = S_i^{-1} x(\mathcal{E}_i)$.
end for
```

```
for $i = K-1, \ldots, 2$ do
    Compute $x(\mathcal{E}_i) = x(\mathcal{E}_i) - S_i^{-1} A(\mathcal{E}_i, \mathcal{E}_{i+1}) x(\mathcal{E}_{i+1})$.
end for
Compute $x(\mathcal{E}_1) = x(\mathcal{E}_1) - S_1^{-1} A(\mathcal{E}_1, \mathcal{E}_2) x(\mathcal{E}_2)$.
```

Once again, the main computational cost comes from inverting the Schur complement blocks S_i. For Cartesian finite difference grids, the cost of computing the inversion of the above factorization was shown to be $O(N^2)$ in 2D and $O(N^{7/3})$ in 3D. A similar argument for finite element meshes can be made, as the number of edges varies approximately linearly with the number of simplex elements.

Just as in the finite difference case, an important physical observation for each S_i can be made. Specifically, restrict the full problem to the first m subdomains, for $m < K$; that is, instead of the whole system in (5.14), consider the smaller system of equations

$$\begin{pmatrix} A(\mathcal{E}_1,\mathcal{E}_1) & A(\mathcal{E}_1,\mathcal{E}_2) & & \\ A(\mathcal{E}_2,\mathcal{E}_1) & A(\mathcal{E}_2,\mathcal{E}_2) & \ddots & \\ & \ddots & \ddots & A(\mathcal{E}_{m-1},\mathcal{E}_m) \\ & & A(\mathcal{E}_m,\mathcal{E}_{m-1}) & A(\mathcal{E}_m,\mathcal{E}_m) \end{pmatrix} \begin{pmatrix} x(\mathcal{E}_1) \\ x(\mathcal{E}_2) \\ \vdots \\ x(\mathcal{E}_m) \end{pmatrix} = \begin{pmatrix} b(\mathcal{E}_1) \\ b(\mathcal{E}_2) \\ \vdots \\ b(\mathcal{E}_m) \end{pmatrix}.$$

This system corresponds to the discretization of the semi-infinite half-space Maxwell problem with a PEC boundary condition on the boundary of $\cup_{i=1}^m \mathcal{T}_i$,

$$\begin{aligned} \nabla \times \tilde{\mu}_r^{-1} \nabla \times \mathbf{E} - \kappa^2 \tilde{\varepsilon}_r \mathbf{E} &= -\imath\omega\mu_0 \mathbf{J} \qquad \text{in} \quad \text{int}\left(\cup_{i=1}^m \mathcal{T}_i\right), \qquad (5.17) \\ \hat{\mathbf{n}} \times \mathbf{E} &= 0 \qquad\qquad\quad \text{on} \quad \partial(\cup_{i=1}^m \mathcal{T}_i). \qquad (5.18) \end{aligned}$$

Note that this boundary is generally not a flat plane; it conforms to the faces of the elements, which results in a rough surface. If the upper m blocks are inverted, one obtains

$$\begin{pmatrix} x(\mathcal{E}_1) \\ \vdots \\ x(\mathcal{E}_m) \end{pmatrix} = (L_1^t)^{-1} \dots (L_{m-1}^t)^{-1} \begin{pmatrix} S_1^{-1} & & \\ & \ddots & \\ & & S_m^{-1} \end{pmatrix} L_{m-1}^{-1} \dots L_1^{-1} \begin{pmatrix} b(\mathcal{E}_1) \\ \vdots \\ b(\mathcal{E}_m) \end{pmatrix}. \tag{5.19}$$

However, due to the structure of L_i for $i = 1, \dots, m$, it is noticed that the S_m^{-1} is unaffected by the left and right operators in (5.19), i.e. the (m, m)-th block in the right hand side is exactly S_m^{-1}. Based on this fact, it can be concluded that S_m^{-1} is the discrete Green's function for degrees of freedom in the m-th layer for (5.18); solving the half-space problem above implicitly constructs an operator for S_m^{-1}. The full half-space problem is once again approximated by the moving PML method.

The process of approximating S_i^{-1} for $i = 2, \dots, K$ in operator form is as follows. Consider the shifted stretching function for subdomain Ω_i,

$$s_{\xi,i}(\xi) = 1 + \imath\sigma_i(\xi), \tag{5.20}$$

where σ_i is the ramp-like function

$$\sigma_i(\xi) = \begin{cases} \theta\left(\frac{-1+(i-1)\ell-\xi}{\ell}\right)^2, & \xi \in [-1 + (i-2)\ell, -1 + (i-1)\ell] \\ 0, & \xi \in [-1 + (i-1)\ell, 1 - \ell] \\ \theta\left(\frac{\xi-1+\ell}{\ell}\right)^2, & \xi \in [1 - \ell, 1] \end{cases}. \tag{5.21}$$

The truncated half-space problem for layer i is then

$$\begin{aligned} \nabla \times \tilde{\mu}_{r,i}^{-1} \nabla \times \mathbf{E} - \kappa^2 \tilde{\varepsilon}_{r,i} \mathbf{E} &= -\imath\omega\mu_0 \mathbf{J} && \text{in} \quad \operatorname{int}(\mathcal{T}_{i-1} \cup \mathcal{T}_i) && (5.22) \\ \hat{\mathbf{n}} \times \mathbf{E} &= 0 && \text{on} \quad \partial(\mathcal{T}_{i-1} \cup \mathcal{T}_i), && (5.23) \end{aligned}$$

where the material parameters in 2D are

$$\tilde{\varepsilon}_{r,i} = \begin{pmatrix} \varepsilon_{xx}\frac{s_{y,i}}{s_x} & \varepsilon_{xy} \\ \varepsilon_{yx} & \varepsilon_{yy}\frac{s_x}{s_{y,i}} \end{pmatrix}, \qquad \tilde{\mu}_{r,i} = s_x s_{y,i} \mu_{zz}, \tag{5.24}$$

and the tensors in 3D are

$$\tilde{\varepsilon}_{r,i} = \begin{pmatrix} \varepsilon_{xx}\frac{s_y s_{z,i}}{s_x} & \varepsilon_{xy} s_{z,i} & \varepsilon_{xz} s_y \\ \varepsilon_{yx} s_{z,i} & \varepsilon_{yy}\frac{s_x s_{z,i}}{s_y} & \varepsilon_{yz} s_x \\ \varepsilon_{zx} s_y & \varepsilon_{zy} s_x & \varepsilon_{zz}\frac{s_x s_y}{s_{z,i}} \end{pmatrix}, \quad \tilde{\mu}_{r,i} = \begin{pmatrix} \mu_{xx}\frac{s_y s_{z,i}}{s_x} & \mu_{xy} s_{z,i} & \mu_{xz} s_y \\ \mu_{yx} s_{z,i} & \mu_{yy}\frac{s_x s_{z,i}}{s_y} & \mu_{yz} s_x \\ \mu_{zx} s_y & \mu_{zy} s_x & \mu_{zz}\frac{s_x s_y}{s_{z,i}} \end{pmatrix}. \tag{5.25}$$

Clearly, subproblem (5.23) requires only the first two layers of the shifted PML function. Denote the stiffness matrix resulting from the discretization of (5.23) as H_i; it is crucial that the degrees of freedom in the local subproblem maintain the same order as in (5.14). Using the multifrontal method with nested dissection, the optimal sparse LU factorization of H_i can be constructed, and the inverse operator H_i^{-1} can be applied efficiently. Now consider the vector $v \in \mathbb{C}^{N_i}$, where N_i is the number of degrees of freedom in layer i. If a vector of zeros $0 \in \mathbb{C}^{N_{i-1}}$ is concatenated with v and H_i^{-1} is applied, the result is

$$H_i^{-1}\begin{pmatrix} 0 \\ v \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}. \tag{5.26}$$

for vectors $w_1 \in \mathbb{C}^{N_{i-1}}$, $w_2 \in \mathbb{C}^{N_i}$. The vector w_2 can then be extracted from the right hand side of (5.26) to obtain the approximation of $S_i^{-1}v$; the operator which performs this concatenation/extraction process is defined as $\tilde{S}_i^{-1} : \mathbb{C}^{N_i} \to \mathbb{C}^{N_i}$.

The setup and application algorithms 5.3.1 and 5.3.2 can be modified accordingly with the new operator $\tilde{S}_i^{-1}$; although it is not entirely obvious, they maintain the same complexity estimates as in the finite difference case.

Algorithm 5.3.3. *Setup the sweeping preconditioner of A.*

```
1: Let $H_1 = A(\mathcal{E}_1, \mathcal{E}_1)$; construct the sparse LU factorization of $H_1$.
2: for $i = 2, \dots, K$ do
3:     Let $H_i$ be the stiffness matrix of (5.23). Construct the optimal sparse LU factorization of $H_i$ using the multifrontal method with nested dissection.
4: end for
```

The setup of the preconditioner requires the solution of each subproblem in (5.23.) In the 2D case, each subproblem has $O(\sqrt{N})$ degrees of freedom. The multifrontal method constructs the solution of the quasi-1D problem with linear complexity, so each subproblem can be solved in $O(\sqrt{N})$ time. Since there are $O(K) = O(\sqrt{N})$ subproblems to be solved, the total setup time in 2D is $O(N)$. In the 3D case, each subproblem contains $O(N^{1/3}) \times O(N^{1/3}) = O(N^{2/3})$ degrees of freedom; consequently, the multifrontal method can solve the quasi-2D problem in $O((N^{2/3})^{3/2}) = O(N)$ time. Since there are $O(K) = O(N^{1/3})$ subproblems, the total complexity to setup the 3D preconditioner is $O(N^{4/3})$.

Algorithm 5.3.4. *Apply the approximate inverse to b to get $x \approx A^{-1}b$.*

```
1: Set $x(\mathcal{E}_i) = b(\mathcal{E}_i)$, for $i = 1, \dots, K$.
2: Compute $x(\mathcal{E}_2) = x(\mathcal{E}_2) - A(\mathcal{E}_2, \mathcal{E}_1) H_1^{-1} x(\mathcal{E}_1)$.
3: for $i = 2, \dots, K-1$ do
```

```
    Compute $x(\mathcal{E}_{i+1}) = x(\mathcal{E}_{i+1}) - A(\mathcal{E}_{i+1}, \mathcal{E}_i)\tilde{S}_i^{-1}x(\mathcal{E}_i)$, where the operator
    $\tilde{S}_i^{-1}$ is described above.
end for
Compute $x(\mathcal{E}_1) = H_1^{-1}x(\mathcal{E}_1)$.
for $i = 2, \ldots, K$ do
    Compute $x(\mathcal{E}_i) = \tilde{S}_i^{-1}x(\mathcal{E}_i)$.
end for
for $i = K-1, \ldots, 2$ do
    Compute $x(\mathcal{E}_i) = x(\mathcal{E}_i) - \tilde{S}_i^{-1}A(\mathcal{E}_i, \mathcal{E}_{i+1})x(\mathcal{E}_{i+1})$.
end for
Compute $x(\mathcal{E}_1) = x(\mathcal{E}_1) - H_1^{-1}A(\mathcal{E}_1, \mathcal{E}_2)x(\mathcal{E}_2)$.
```

The main cost in Algorithm 5.3.4 is the application of $\tilde{S}_i^{-1}$. In 2D, the cost of applying the inverse of the sparse LU factorization is linear; for each layer, this amounts to $O(\sqrt{N})$ time. The computation is done $O(\sqrt{N})$ times, which results in a total complexity of $O(N)$. In 3D, applying each inverse can be done in logarithmic linear time, i.e. $O(N^{2/3} \log N^{2/3})$. With $O(N^{1/3})$ layers, this results in a total complexity of $O(N \log N^{2/3}) = O(N \log N)$.

Algorithms 5.3.3 and 5.3.4 define an inverse operator M^{-1}, which is an approximation of A^{-1}. Once again, for stability reasons, it is more prudent to construct the approximate sweeping factorization for the equation

$$\nabla \times \tilde{\mu}_r^{-1} \nabla \times \mathbf{E} - (\kappa + \imath\alpha)^2 \tilde{\varepsilon}_r \mathbf{E} = -\imath\omega\mu_0 \mathbf{J}, \tag{5.27}$$

where α is a positive damping constant of $O(1)$. With M_α^{-1} being the approximate inverse operator for the discretization of (5.27), the preconditioned linear system

$$M_\alpha^{-1} A x = M_\alpha^{-1} b. \tag{5.28}$$

can be solved using a Krylov subspace iterative solver. As the numerical results will show, the rate of convergence of the iterative solver will be either independent or logarithmically dependent on frequency, with a low number of iterations.

A few remarks must be made about practical issues with the algorithm. First, the moving PML method is presented here with sweeping in the z-direction for 3D; this choice is arbitrary, as the mesh can also be partitioned into slabs orthogonal to the x or y axes. For these cases, the appropriate PML functions must be shifted for the material tensors in (5.25). Second, each subdomain is defined to have the same thickness as the PML; this is not a restriction, as one could configure the subdomains so that each slab is of a different thickness. This is particularly useful when computing on adaptive or locally refined meshes. In addition, the PML used to back each slab does not need to coincide with the adjacent slab; this simplification is chosen for ease of implementation. Finally, other absorbing boundary conditions (ABC) can be utilized in place of the PML. The use of an ABC would significantly reduce the memory and computational time, as each subproblem would not need to be paddded with a buffer layer.

5.4 Numerical Results in 2D

Several numerical results are presented to support the claims for the accuracy and linear complexity of the sweeping preconditioner. All of the 2D algorithms are implemented in sequential C++ code on a server equipped with Intel Xeon E7420 2.13 GHz processors. For the multifrontal method used to solve each subproblem, a sequential version of MUMPS [3] is employed; the iterative solver is GMRES with a residual tolerance set to 10^{-3}. With Cartesian PMLs, there are a few examples of heterogeneous media which are of importance to the optics and photonics community:

1. A converging lens profile. Here, consider the isotropic, heterogeneous material

$$\varepsilon_r = \left(1 + e^{-30(x^2+y^2)}\right)\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, \qquad \mu_r = \left(1 + e^{-30(x^2+y^2)}\right). \tag{5.29}$$

At the center of the lens, the wavespeed is $\frac{1}{2}c$, where $c = \frac{1}{\sqrt{\mu_0\varepsilon_0}}$ is the speed of light in free space.

2. A periodic medium. The 2D function

$$f(x,y) = 1 + \frac{1}{4}\cos\left(20\left(\frac{x}{\sqrt{2}} + \frac{y}{\sqrt{2}}\right)\right) + \frac{1}{4}\cos\left(20\left(\frac{x}{\sqrt{2}} - \frac{y}{\sqrt{2}}\right)\right) \tag{5.30}$$

is used to form the isotropic material

$$\varepsilon_r = \sqrt{f(x,y)}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, \qquad \mu_r = \sqrt{f(x,y)}. \tag{5.31}$$

In these examples, the current $\mathbf{J}$ is a solenoidal vector field in the x-y plane derived from a Gaussian point source oriented in the z-direction, i.e.

$$\mathbf{J}(x,y) = \nabla \times \left(\hat{\mathbf{z}} e^{-\kappa^2(x^2+(y-0.5)^2)}\right). \tag{5.32}$$

The preconditioner is constructed with $\alpha = 1$ and PML width $\ell = 2\lambda$. For each experiment, the domain is fixed while simultaneously increasing the wavenumber κ, keeping the same resolution for elements per wavelength; listed in each table are the width of the problem in wavelengths $K := \frac{\kappa}{\pi} = \frac{\omega\sqrt{\mu_0\varepsilon_0}}{\pi}$, the number of degrees of freedom N, the preconditioner setup time T_{setup}, the iterative solver time T_{solve}, and the number of iterations necessary for convergence N_{iter}.

From the Table in Figure 5.3, it is observed that when K doubles, the number of degrees of freedom increases by a factor of 4. At the same time, T_{setup} also increases approximately by a factor of 4, which shows the linear complexity of Algorithm 5.3.3. The time per iteration $\frac{T_{\text{solve}}}{N_{\text{iter}}}$ also grows roughly by a factor of 4; thus, it can be inferred that the application Algorithm 5.3.4 is also $O(N)$. As the number of iterations either remains constant or grows very weakly with frequency, the entire solver has $O(N)$ complexity. The complexity graphs in Figure 5.4 support these claims.

The sweeping preconditioner can also be used with a cylindrical PML; in this case, the computational domain is a circle of radius 1. Instead of partitioning the domain into equally sized horizontal or vertical layers, a series of concentric shells are introduced. This organization is natural because the cylindrical PML is a shell surrounding the domain; the sweeping direction

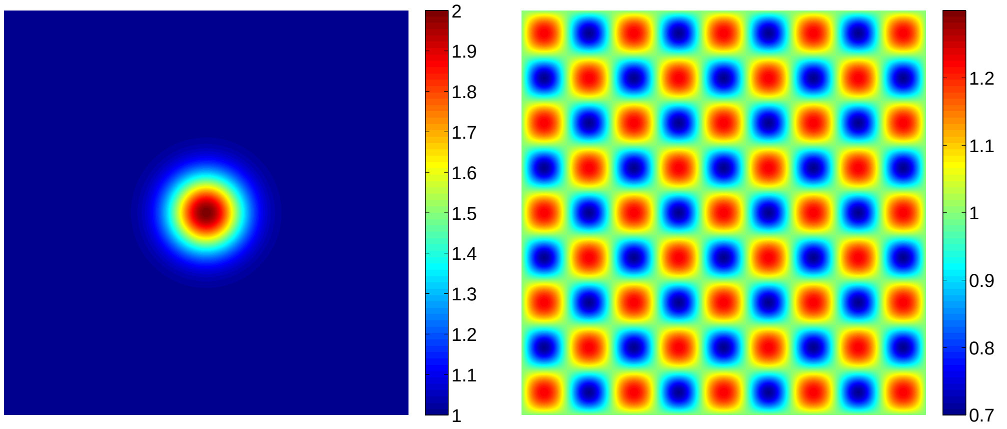
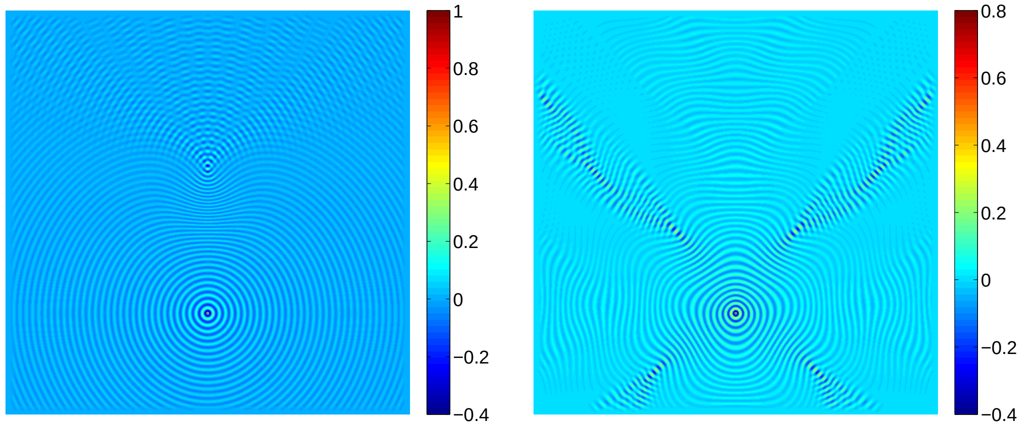


Figure 5.2: Material ε_r and μ_r for the 2D converging lens (left) and periodic medium (right).



K	N	T_{setup}	T_{solve}	N_{iter}
16	5.549e+04	4	2	5
32	2.216e+05	15	8	5
64	8.855e+05	61	32	5
128	3.540e+06	253	133	5
256	1.416e+07	1028	742	7

K	N	T_{setup}	T_{solve}	N_{iter}
16	5.549e+04	4	2	5
32	2.216e+05	15	8	5
64	8.855e+05	61	32	5
128	3.540e+06	251	132	5
256	1.416e+07	1040	630	6

Figure 5.3: 2D cartesian PML results for Maxwell's equations. Real part of the magnetic field H_z at $K = 64$ with computational results for the converging lens (left) and periodic medium (right).

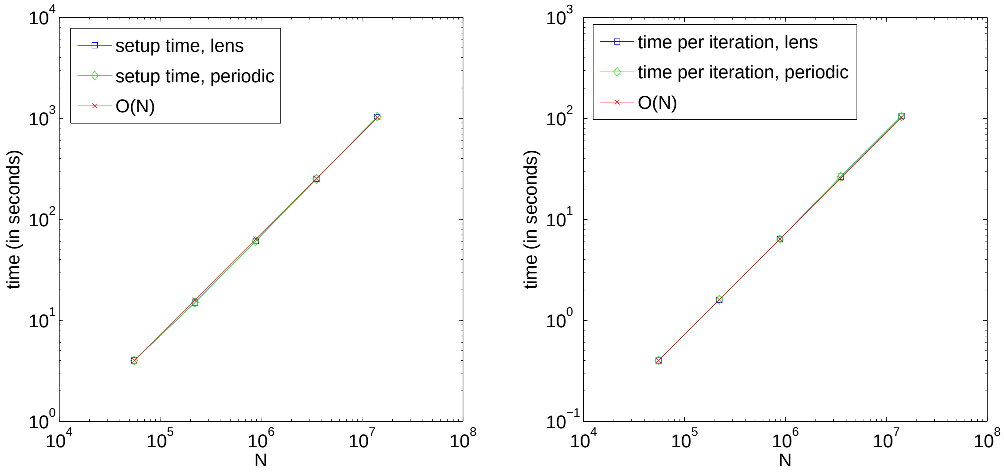


Figure 5.4: The complexity graphs for setup time of the preconditioner (left) and time per iteration of the iterative method (right) in the 2D case for Maxwell's equations.

is oriented along the radial direction, towards the center. The following are standard examples in the high-frequency scattering and metamaterial communities:

1. A cylindrical PEC scatterer. Here, a PEC cylinder is embedded in free space with a radius of 0.25.

2. A transformation optics cloaking device. Consider the cylindrical cloak derived from coordinate transformations [74] with singular parameters near the inner radius of the cloak; inside the cloak lies a PEC. The mate-

rial is characterized by the relative parameters in cylindrical coordinates:

$$\varepsilon_{\rho\rho} = \mu_{\rho\rho} = \frac{\rho - a}{\rho} \tag{5.33}$$

$$\varepsilon_{\theta\theta} = \mu_{\theta\theta} = \frac{\rho}{\rho - a} \tag{5.34}$$

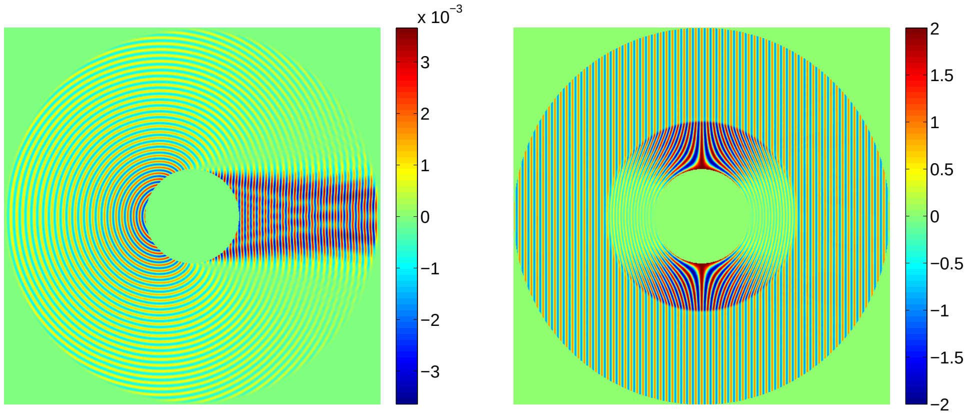
$$\varepsilon_{zz} = \mu_{zz} = \frac{\rho - a}{\rho}\left(\frac{b}{b-a}\right)^2 \tag{5.35}$$

where a and b are the inner and outer radii of the cloak, respectively. For these experiments, $a = 0.25$ and $b = 0.5$. This example is particularly interesting for a few reasons: the medium is discontinuous over the boundary of the cloaking shell, and its material tensor in Cartesian coordinates is anisotropic with off-diagonal entries.

Instead of a current source, a plane wave is chosen for the incident field. This requires the use the scattered field formulation; although the right hand side is altered slightly, this does not change the construction of the stiffness matrices or preconditioner. Here, the plane wave is

$$\mathbf{E}_{\text{inc}} = \hat{\mathbf{y}} e^{-\imath \kappa x} \tag{5.36}$$

The results for the cylindrical PML examples also show the linear complexity of the 2D algorithm. In this instance, every time the frequency is doubled, the radius of the domain in terms of wavelength is multiplied by a factor of 2; this yields an increase in the degrees of freedom by a factor of 4. For the PEC scatterer, the setup time grows roughly by a factor of 4, and the application time also increases by a factor of 4, implying $O(N)$ complexity.



K	N	T_{setup}	T_{solve}	N_{iter}
16	3.652e+04	2	1	4
32	1.454e+05	9	3	4
64	5.803e+05	40	14	4
128	2.319e+06	157	72	5
256	9.270e+06	633	291	5

K	N	T_{setup}	T_{solve}	N_{iter}
16	2.094e+05	14	3	4
32	8.354e+05	54	20	5
64	3.335e+06	258	104	5
128	1.333e+07	1150	442	5
256	-	-	-	-

Figure 5.5: 2D cylindrical PML results for Maxwell's equations. Real part of the scattered field H_z at $K = 64$ for the PEC cylinder (left) and real part of the total field E_x at $K = 64$ for the transformation optics cloak (right), with computational results.

For the cloaking device, however, it is observed that the setup and solve times increase by a factor of 4.2 instead, implying an $O(N \log N)$ complexity. The reason this example has a less optimal complexity result is because of the dense mesh used to discretize the cloaking layer. Because the cloaking device relies on transformation optics, the oscillations are condensed inside the outer shell; thus, the mesh must be refined inside to keep the same dispersion relationship. Each subproblem on the cloaking shell results in a thicker 2D problem instead of a quasi-1D strip, resulting in the increase in computational time. The added DOFs also increase the memory requirements for each problem, limiting the largest domain to $K = 128$.

5.5 Numerical Results in 3D

In the 3D case, a few examples are given to show the $O(N^{4/3})$ complexity of the sweeping preconditioner. The 3D code is implemented in sequential C++ on a server equipped with 2.2 GHz AMD Opteron 6174 processors. Once again, a sequential version of MUMPS is used for the multifrontal method and GMRES iteration is set to a residual tolerance of 10^{-3}. The preconditioner is tested on the following 3D media:

1. A converging lens profile. Here, consider the isotropic, heterogeneous material

$$\varepsilon_r = \left(1 + e^{-8(x^2+y^2+z^2)}\right)\mathbf{I} \qquad \mu_r = \left(1 + e^{-8(x^2+y^2+z^2)}\right)\mathbf{I} \tag{5.37}$$

where $\mathbf{I}$ is the 3×3 identity matrix. At the origin, the wavespeed is $\frac{1}{2}c$.

2. A periodic medium. Using the functions

$$f_{\pm}(x,y) = \cos\left(2\pi\left(\frac{x}{\sqrt{2}} \pm \frac{y}{\sqrt{2}}\right)\right), \tag{5.38}$$

$$g(x,y,z) = 1 + \frac{f_{+}(x,y)f_{-}(x,y)\cos(2\pi z)}{2}, \tag{5.39}$$

the oscillatory medium is

$$\varepsilon_r = \left(0.3 + \sqrt{g(x,y,z)}\right)\mathbf{I}, \qquad \mu_r = \left(0.3 + \sqrt{g(x,y,z)}\right)\mathbf{I}. \tag{5.40}$$

The current source in the first example is a Gaussian point source oriented in the z-direction, i.e.

$$\mathbf{J}(x,y,z) = \hat{\mathbf{z}} e^{-\frac{\kappa^2}{\pi^2}(x^2+(y-0.75)^2+z^2)}. \tag{5.41}$$

Note that the source is located farther away from the center to allow the caustics to develop behind the inhomogeneity at the center of the domain. In most cases this tends to increase the number of iterations necessary for convergence, as the source is closer to the PML, but this effect was not observed for this case. For example 2, the source is placed closer to the origin since the inhomogeneities are all over the domain. The preconditioner is constructed with $\alpha = 1$ and PML width $\ell \approx \lambda$.

The complexity of the 3D algorithm is illustrated in Figure 5.7. Keeping the element-to-wavelength ratio constant while doubling the frequency forces the total degrees of freedom to increase by a factor of 8. At the same time, the $O(N^{4/3})$ complexity estimate implies that the setup time should increase by a factor of $8^{4/3} = 16$. However, an increase in setup time by a factor of 11 or 13

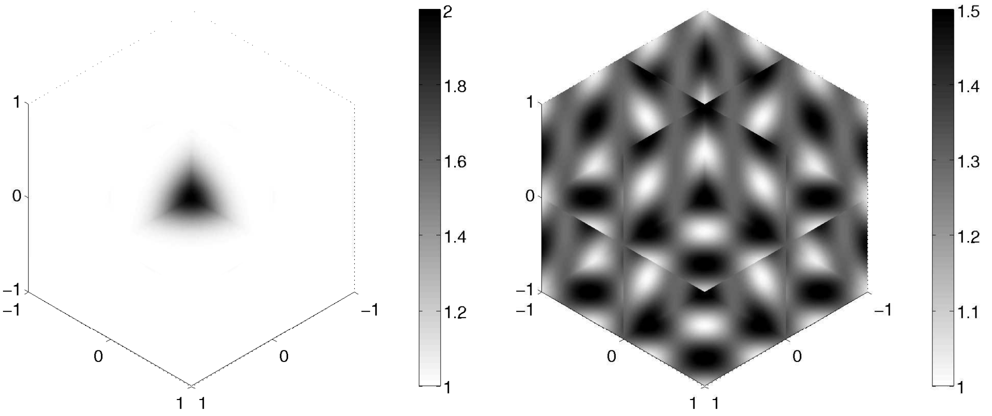
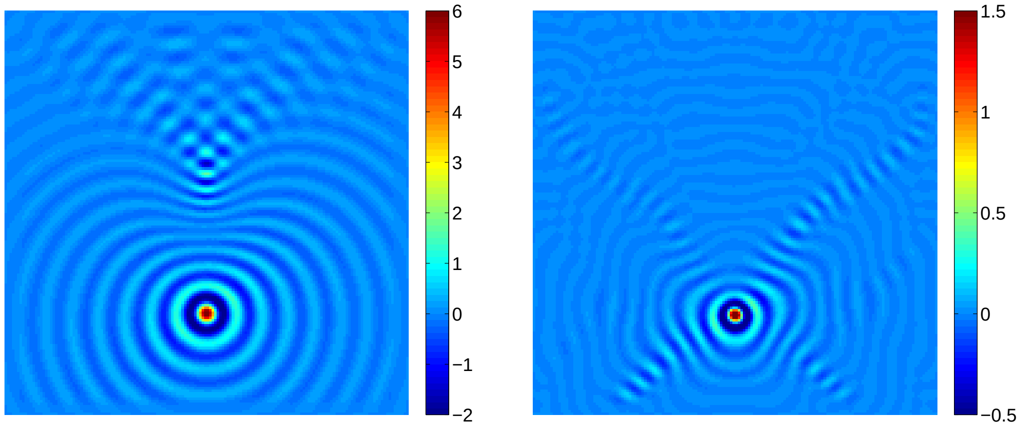


Figure 5.6: Slice plots of ε_r and μ_r for the 3D converging lens (left) and periodic medium (right).

is observed. The cause of this improvement is clear; in practice, the number of subdomains is closer to $O(N^{1/5})$ rather than $O(N^{1/3})$. Thus, the complexity grows as $O(N^{6/5})$ instead. For the application of the preconditioner, the $O(N \log N)$ estimate implies that the solve time should increase roughly by a factor of 10 given the same number of iterations. Figure 5.8 supports these claims.



K	N	T_{setup}	T_{solve}	N_{iter}
5	1.608e+05	107	17	5
10	1.258e+06	1366	171	5
20	9.948e+06	15819	1808	6

K	N	T_{setup}	T_{solve}	N_{iter}
5	1.608e+05	111	18	5
10	1.258e+06	1371	172	5
20	9.948e+06	15860	1820	6

Figure 5.7: Real part of the field E_z in the x-y plane at $K = 20$ with computational results for the converging lens (left) and the periodic medium (right).

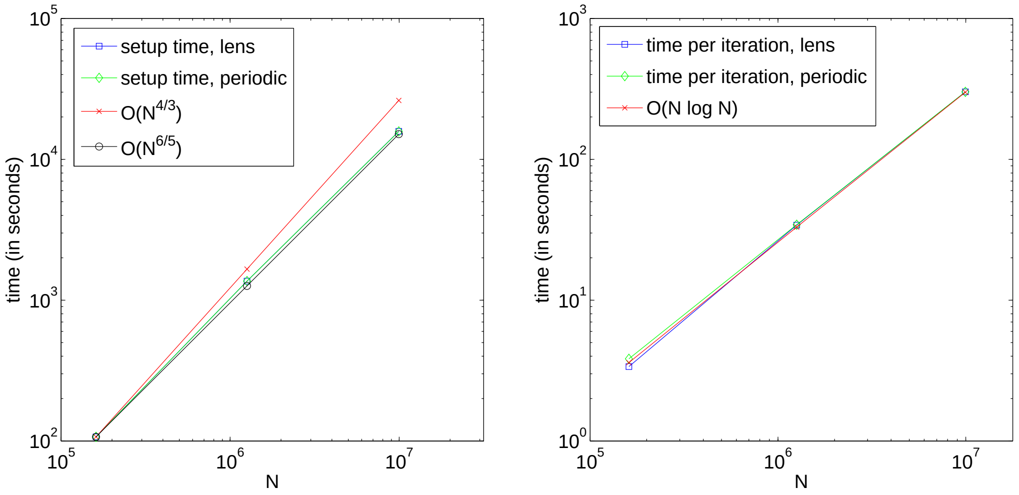


Figure 5.8: The complexity graphs for setup time of the preconditioner and time per iteration of the iterative method in the 3D case for Maxwell's equations.

Chapter 6

Preconditioners for Acoustics and Elasticity: Spectral Element Methods

6.1 PMLs for Helmholtz and Elastic Wave Equations

Consider the domain $\Omega = [0, 1]^3$ with PML thickness ℓ, and recall the ramp functions σ and stretching functions s_ξ for $\xi = x, y, z$ from (4.1) and (4.2). Given these functions, the differential operator $\tilde{\nabla}$ in complex-stretched coordinates is

$$\tilde{\nabla} = \mathbf{S} \cdot \nabla = \hat{x} \frac{1}{s_x} \frac{\partial}{\partial x} + \hat{y} \frac{1}{s_y} \frac{\partial}{\partial y} + \hat{z} \frac{1}{s_z} \frac{\partial}{\partial z}, \tag{6.1}$$

where $\mathbf{S}$ is the matrix defined in (4.3). The source-free Helmholtz equation in complex coordinates is then

$$\tilde{\nabla} \cdot (a \tilde{\nabla} u) + \frac{\omega^2}{c^2} u = 0. \tag{6.2}$$

If the stretching functions from $\tilde{\nabla}$ are transferred onto the material parameters, the equation can be rewritten in regular space as

$$\nabla \cdot (\tilde{a} \nabla u) + \frac{\omega^2}{\tilde{c}^2} u = 0, \tag{6.3}$$

where the PML material parameters $\tilde{a}$ and $\tilde{c}$ satisfy

$$\tilde{a} = \begin{pmatrix} a_{xx} \frac{s_y s_z}{s_x} & a_{xy} s_z & a_{xz} s_y \\ a_{yx} s_z & a_{yy} \frac{s_x s_z}{s_y} & a_{yz} s_x \\ a_{zx} s_y & a_{zy} s_x & a_{zz} \frac{s_x s_y}{s_z} \end{pmatrix}, \qquad \tilde{c}^2 = \frac{c^2}{s_x s_y s_z}. \tag{6.4}$$

For linear elasticity, the formulation is similar [51]. Starting with the elastic wave equation (1.12) in index notation, introduce the stretching functions s_i for the i-th component of $u = (u_1, u_2, u_3)$. This yields

$$-s_j^{-1}\big(C_{ijkl}u_{k,l}s_l^{-1}\big)_{,j} - \omega^2 \rho u_i = 0.$$

By multiplying the equation with $s_1 s_2 s_3$, the PDE can be written as

$$-\big(\tilde{C}_{ijkl}u_{k,l}\big)_{,j} - \omega^2 \tilde{\rho} u_i = 0,$$

where the fourth-order elasticity tensor and density for the PML problem are

$$\tilde{C}_{ijkl} = C_{ijkl}\frac{s_1 s_2 s_3}{s_j s_l}, \qquad \tilde{\rho} = \rho s_1 s_2 s_3.$$

6.2 Variational Formulations

The full Helmholtz problem with the radiation condition is reduced to

$$\begin{aligned} \nabla \cdot (\tilde{a}\nabla u) + \frac{\omega^2}{\tilde{c}^2}u &= f \qquad \text{in } \Omega && (6.5)\\ u &= 0 \qquad \text{on } \partial\Omega. && (6.6) \end{aligned}$$

Multiplying the PDE by a test function $v \in H_0^1(\Omega)$ and integrating by parts, the variational formulation derived is

$$\int_\Omega \tilde{a}\nabla u \cdot \nabla v dV - \omega^2 \int_\Omega \frac{1}{\tilde{c}^2} uv dV = \int_\Omega f v dV \tag{6.7}$$

Note that the $\frac{1}{\tilde{c}^2}$ term is inside the integral because the material is heterogeneous (even if $\tilde{c}$ were constant, the PML is heterogeneous). The sesquilinear form and linear functional in the Helmholtz case are then

$$B(u,v) = \int_{\Omega} \tilde{a}\nabla u \cdot \nabla v dV - \omega^2 \int_{\Omega} \frac{1}{\tilde{c}^2} uv dV \tag{6.8}$$

$$F(v) = \int_{\Omega} fv dV. \tag{6.9}$$

Now, denote $X = H_0^1(\Omega)$. The weak form of the problem is to find $u \in X$ such that

$$B(u,v) = F(v), \qquad \forall v \in X. \tag{6.10}$$

Like the indefinite Maxwell problem in the previous chapter, proving existence and uniqueness for indefinite Helmholtz problem is more difficult than for the Poisson problem. These proofs are detailed in [56].

Since the elastic wave equation is vector-valued, vector testing functions are necessary. Consider $\mathbf{v} = (v_1, v_2, v_3) \in (H_0^1(\Omega))^3$. If the dot product of the PDE and $\mathbf{v}$ is taken, and integration-by-parts is performed on the j-th derivative, the result in index notation is

$$\int_{\Omega} v_{i,j}\tilde{C}_{ijkl}u_{k,l} d\Omega - \omega^2 \int_{\Omega} \tilde{\rho} u_i v_i d\Omega = \int_{\Omega} f_i v_i d\Omega$$

Defining the sesquilinear form and linear functional as

$$B(\mathbf{u},\mathbf{v}) = \int_{\Omega} v_{i,j}\tilde{C}_{ijkl}u_{k,l} d\Omega - \omega^2 \int_{\Omega} \tilde{\rho} u_i v_i d\Omega \tag{6.11}$$

$$F(\mathbf{v}) = \int_{\Omega} f_i v_i d\Omega, \tag{6.12}$$

the weak form of the problem is to find $\mathbf{u} \in (H_0^1(\Omega))^3$ such that

$$B(\mathbf{u},\mathbf{v}) = F(\mathbf{v}), \qquad \forall \mathbf{v} \in (H_0^1(\Omega))^3. \tag{6.13}$$

6.3 Spectral Element Methods

In time-harmonic wave propagation, there are a variety of higher-order finite element methods which have been developed to improve the dispersion relationship and reduce pollution error. A review of these methods can be found in [87, 50]. Recently, the spectral element method [73] has gained popularity in the seismic wave propagation community, most notably for time-domain simulations [59, 60]; the main reason is that the diagonal mass matrix allows for computationally efficient time-stepping algorithms. In this section, these ideas are applied to the frequency domain.

The spectral element method (SEM) is simply a higher-order finite element method with the basis functions defined at the Gauss-Lobatto quadrature nodes; in contrast, the standard high-order FEM defines its basis at equispaced points of the element. The irregular spacing of the nodes produces interpolation functions which have a global maximum of 1 at the node which each function is defined on, minimizing the spurious oscillation error from Runge phenomena. Consider the Gauss-Lobatto nodes $\{x_j\}_{j=1}^{p+1}$ on the interval $[-1, 1]$, where p will be the resulting polynomial order. The 1D Lagrangian interpolation functions for this grid are

$$L_i(x) = \prod_{\substack{j=1 \\ i\neq j}}^{p+1} \frac{(x - x_j)}{(x_i - x_j)} \tag{6.14}$$

For the master hexahedral element, a tensor-product structure is used with the 1D interpolation functions to generate the basis function at node (x_i, y_j, z_k):

$$\phi^e_{ijk}(x, y, z) = L_i(x)L_j(y)L_k(z). \tag{6.15}$$

Thus, in 3D, there are $(p+1)^3$ nodal basis functions for each hexahedral element.

The discretization of the weak form (6.10) is as follows; by abusing the notation slightly and denoting $\{\phi_m\}_{m=1}^N$ as the tensor-product basis functions of a hexahedral mesh, the finite element approximation u^h takes the form

$$u^h = \sum_{m=1}^{N} c_m \phi_m. \tag{6.16}$$

Following the Galerkin formulation, this expansion is inserted into the sesquilinear form for u, while the testing functions v are represented using the same basis. The linear system is then

$$Ax = b, \tag{6.17}$$

where $A_{mn} = B(\phi_m, \phi_n)$, $x_n = c_n$, and $b_m = F(\phi_m)$.

For elasticity, each component of the vector field $\mathbf{u}$ is discretized using the scalar basis functions defined above; thus, when comparing to the Helmholtz problem on the same mesh, a discretization of the elastic wave equation should have three times the number of DOFs in 3D. The finite element approximation of the i-th component of $\mathbf{u}$ is then

$$u_i^h = \sum_{m=1}^{N} c_{i,m} \phi_m. \tag{6.18}$$

If the degrees of freedom are ordered in the coefficient vector as

$$x = (c_{1,1}, c_{2,1}, c_{3,1}, c_{1,2}, c_{2,2}, c_{3,2}, \ldots, c_{1,N}, c_{2,N}, c_{3,N})^t.$$

and the auxiliary indices $I(i,m) = 3(m-1)+i$ and $J(j,n) = 3(n-1)+j$ are used, then the Galerkin formulation yields the system $Ax = b$, where the entries are

$$A_{IJ} = B(\phi_m \mathbf{e}_i, \phi_n \mathbf{e}_j), \quad b_I = F(\phi_m \mathbf{e}_i), \quad x_I = c_{i,m}.$$

Here, $\mathbf{e}_i$ is the unit vector in direction i.

To evaluate the integrals in the sesquilinear form and linear functional, a few quadrature methods can be considered. The most common approach is to reuse the Gauss-Lobatto quadrature nodes and weights. Because the Lagrangian polynomial functions are defined at the Gauss-Lobatto nodes and $L_i(x_j) = \delta_{ij}$, many of the terms in the quadrature summation are zero. This allows for very efficient computation; the complexity of the computation over each element is $O((p+1)^6)$ in 3D, as opposed to $O((p+1)^9)$ for exact Gauss-Legendre quadrature. Furthermore, the $-\omega^2$ mass term produces a diagonal matrix; this is mainly a benefit for time-domain simulations, in which inverting the mass matrix is usually necessary. Very recently, a non-standard "blended" quadrature scheme [1] has been introduced to reduce the dispersion error of the discretization. The main idea rests on the fact that the exact mass matrix (computed by Gauss-Legendre) produces a wavefield with accelerated phase velocity, while the diagonal mass matrix (computed by Gauss-Lobatto) produces a wavefield with lagging phase velocity. If the two methods are linearly combined, then their dispersion errors can be cancelled out, resulting in an improvement of two orders of accuracy in the dispersion relationship. For tests

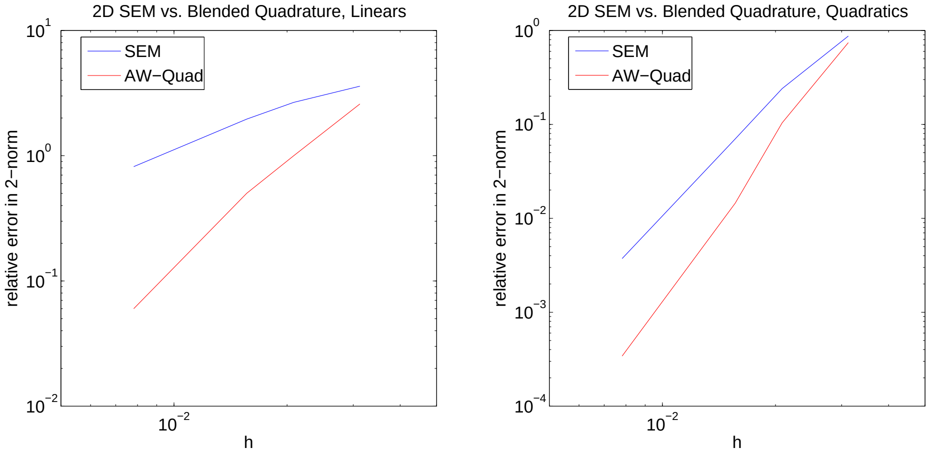


Figure 6.1: Relative error for the 2D Helmholtz Green's function computed by spectral elements as a function of h. The blue lines are for Gauss-Lobatto quadrature, while the red lines are for Ainsworth-Wajid quadrature.

with the free-space Green's function for $p = 1$ and $p = 2$ on an 8λ-wide domain, significant improvement can be seen in the L_2 relative error in the wavefield as seen in figures 6.1 and 6.2. For higher-order polynomials, the improvement was observed to be less drastic; thus, when $p > 2$, the code developed uses regular Gauss-Lobatto quadrature for efficiency.

6.4 Sweeping Preconditioners for Helmholtz and Elasticity

For uniform hexahedral elements, the same mesh partition algorithm can be utilized, and the block tridiagonal structure and block LDL^t factorization maintain the same structure as in the unstructured mesh case. A few points must be emphasized for the spectral element implementation, however. Since the hexahedral mesh is uniform on the cube, the cartesian PML is aligned

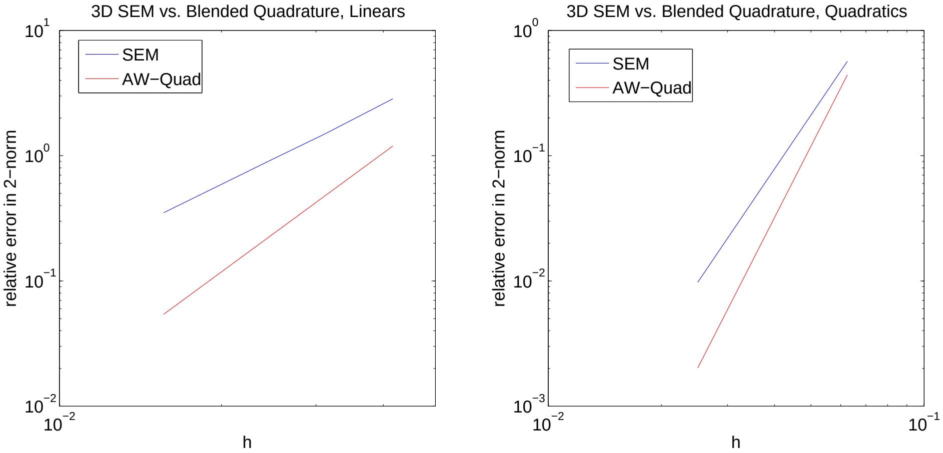


Figure 6.2: Relative error for the 3D Helmholtz Green's function computed by spectral elements as a function of h. The blue lines are for Gauss-Lobatto quadrature, while the red lines are for Ainsworth-Wajid quadrature.

with the elements; in other words, the PML thickness is an integer multiple of the element length h. Second, for high p values, there are significantly more nonzero entries in both the L and D matrices. To minimize the nonzero entries in the off-diagonal blocks of L, the domain is partitioned into subdomains Ω_i along the element boundaries; this ensures that the interior degrees of freedom in an element are not split between two preconditioning layers.

Nevertheless, the inversion of each Schur complement still represents the half-space Green's function, and the moving PML method can be applied with the same complexity estimates. In this case, mesh $\mathcal{T}$ is now made up of elements t_j which are hexahedrals. For the Helmholtz problem, the half space problem for the set of degrees of freedom $\mathcal{E}_i$ corresponding to Ω_i, $i = 2, ..., K$, is then

$$\begin{aligned} \nabla \cdot (\tilde{a}_i \nabla u) + \frac{\omega^2}{\tilde{c}_i^2} u &= f \qquad \text{in } \mathcal{T}_i \cup \mathcal{T}_{i-1} && (6.19) \\ u &= 0 \qquad \text{on } \partial(\mathcal{T}_i \cup \mathcal{T}_{i-1}), && (6.20) \end{aligned}$$

where the shifted material parameters are

$$\tilde{a}_i = \begin{pmatrix} a_{xx}\frac{s_y s_{z,i}}{s_x} & a_{xy} s_{z,i} & a_{xz} s_y \\ a_{yx} s_{z,i} & a_{yy}\frac{s_x s_{z,i}}{s_y} & a_{yz} s_x \\ a_{zx} s_y & a_{zy} s_x & a_{zz}\frac{s_x s_y}{s_{z,i}} \end{pmatrix}, \qquad \tilde{c}_i^2 = \frac{c^2}{s_x s_y s_{z,i}}, \qquad (6.21)$$

in 3D. The shifted stretching function $s_{\xi,i}$ for $\xi = x, y, z$ is defined in (5.20).

For the elastic wave problem, the truncated PML problem to be solved for the degrees of freedom in $\mathcal{E}_m$ corresponding to Ω_m, $m = 2, ..., K$ is

$$\begin{aligned} -\big(\tilde{C}^m_{ijkl} u_{k,l}\big)_{,j} - \omega^2 \tilde{\rho}^m u_i &= f_i \qquad \text{in } \mathcal{T}_m \cup \mathcal{T}_{m-1} \\ u_i &= 0 \qquad \text{on } \partial(\mathcal{T}_m \cup \mathcal{T}_{m-1}), \end{aligned}$$

where the shifted fourth-order elasticity tensor and density are

$$\tilde{C}^m_{ijkl} = C_{ijkl}\frac{s_1 s_2 s_{3,m}}{s_{j,m} s_{l,m}}, \qquad \tilde{\rho}^m = \rho s_1 s_2 s_{3,m}.$$

Here, $s_{j,m}$ and $s_{l,m}$ are the shifted stretching functions only if $j = 3$ or $l = 3$, respectively.

6.5 Parallelization

A parallel version of the moving PML preconditioner using 2nd order finite differences and the Helmholtz equation is outlined in [77]. Here, the algorithms are extended to higher order spectral elements and elasticity.

The parallelization of the sweeping preconditioner is challenging from a few perspectives. First, it is clear from algorithms 5.3.3 and 5.3.4 that the setup stage of the preconditioner can be parallelized, but the application stage of the preconditioner cannot. The multifrontal factorization of each truncated subproblem is completely independent of the other subproblems; however, when applying the approximate inverse, each sweep of the domain must be done sequentially, because the off-diagonal blocks of the lower triangular matrices $(L_i)^{-1}$ and upper triangular matrices $(L_i^t)^{-1}$ apply the Schur complement inverse of one subdomain to the adjacent subdomain. This is similar to the problem faced in domain decomposition, where multiplicative Schwarz methods are not parallelizable because of the interaction between adjacent subdomains and their effect on the updated residual. Thus, even if the factorizations can be done in parallel, the information for each factorization would have to be redistributed across all processors to be the most efficient in the solve stage.

Secondly, there are very few options for scalable sparse direct solvers in the open source community. Many of the most popular codes, such as MUMPS or SuperLU, are not fully scalable when increasing the number of cores into the thousands [49]. It should be noted that the Watson Sparse Matrix Package (WSMP) is a highly scalable direct solver which is freely available. For this work, however, the open source multifrontal code Clique [25] developed by Jack Poulson is employed. Clique uses ParMETIS to generate a nested dissection ordering for the supernodal elimination tree, and subtree-to-subteam process mappings [48] for scalable factorizations. In addition, during the factorization stage, Clique does selective inversion [79] to obtain more scalable triangular solves.

Some general issues must also be discussed when implementing the parallel spectral element method. In many domain decomposition algorithms such as Schur complement approaches or non-overlapping Schwarz methods, local subproblems are defined by doing a 3D decomposition of the global mesh and assigning a group of elements to each process. In the sweeping preconditioner, however, the mesh partitioning algorithm does a 1D decomposition of the full domain; as illustrated earlier, these subproblems are quasi-2D slabs. To utilize all processors efficiently in both construction of the global stiffness matrix and local subproblem stiffness matrices, a 2D decomposition orthogonal to the direction of the sweeping is performed. Each process only contains mesh information local to a pillar of elements. This way, mesh partitioning can occur individually on each process in parallel. For sparse matrices, both the global sparse matrix and local subproblems are stored using a distributed row format; this format is common for sparse direct solvers. Since degrees of freedom on the boundary of an element may be shared by multiple processes, some communication is necessary when computing the contribution of a process to the stiffness matrix.

In consideration of all of these issues, the most efficient parallel algorithm is as follows. For the setup stage, factorize the stiffness matrix for each subproblem one after another, using all processors for each problem; since the construction of the stiffness matrix and multifrontal factorization are scalable, the setup stage is observed to be highly scalable as well. In the solution stage, all processors are used for each triangular solve in the application of the preconditioner. Since the triangular solve is not highly scalable when going to a large number of cores, the iterative solution by GMRES usually takes a majority of the time.

6.6 Numerical Results

The parallel spectral element code with the sweeping preconditioner is implemented in C++ on the Lonestar Linux Cluster at the Texas Advanced Computing Center (TACC). On each node of the cluster, there are 2 six-core Intel Xeon 5680 processors and 24 GB of DDR3-1333MHz memory. In each of the examples, the number of processors is always a power of 2; thus, only eight of the twelve cores on each node is used. The number of total nodes can be determined by dividing the total number of processors by eight.

Many of the inputs and parameters have been kept the same as in the electromagnetics code. Specifically, the iterative solver used is GMRES with a residual tolerance of 1e-4. The damping parameter of the preconditioner, α, is set to 1. In each problem, the polynomial order for the Helmholtz equation is set to 5, while the polynomial order for the elasticity problem is set to 3. Both the PML and subdomain thickness are set to be approximately half a wavelength wide, at the fastest wave speed of the model. The models are discretized so that at the shortest wavelength (or slowest wave speed), there are approximately 4 or 5 grid points per wavelength. The forcing function for each problem is a point source placed in the middle of the domain; in the elasticity case, the orientation of the source is in the x-direction.

Because an anisotropic elasticity version of the models are not available, most geophysicists would interpret the data as P-wave velocity; here, we have modeled the data as the S-wave velocity. If μ is the shear modulus, then the speed of the S-wave is $\sqrt{\frac{\mu}{\rho}}$; the parameters are modified so that this quantity is the speed given by the model. In the following examples, isotropic elasticity is considered; that is, the fourth order elasticity tensor C_{ijkl} takes the form

$$C_{ijkl} = \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) + \lambda\delta_{ij}\delta_{kl}$$

where λ and μ are the first and second Lamé parameters, respectively.

6.6.1 Overthrust Model

A standard example of a wave speed model in seismic imaging is the SEG/EAGE Overthrust model, which takes data from thrust belts in the Canadian rockies. The dimensions are 20 km × 20 km × 4.65 km , and the original data for the wave speed is given on a $801 \times 801 \times 187$ grid. The model is characterized by discontinuous layers of material with varying wave speeds, as shown in the slice plot of figure 6.3. The minimum wave speed in the model is 2.179 km/s, while the maximum wave speed is 6 km/s. Because the data is given on a uniform grid, interpolation is necessary to get the velocities at the Gauss-Lobatto points of the spectral element mesh. For simplicity, linear interpolation is done.

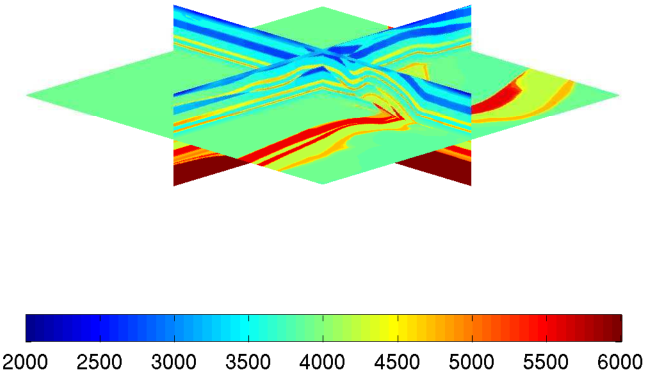
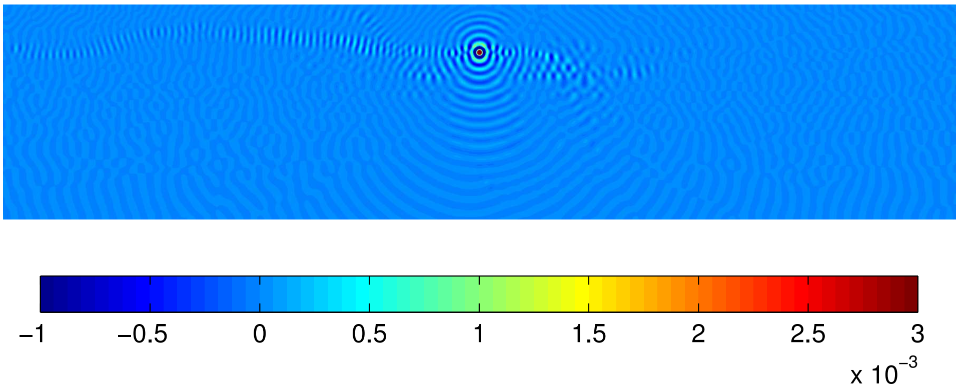


Figure 6.3: SEG/EAGE Overthrust model. Velocity data is given in meters/second.

For this problem, we have enforced the sweeping direction to be in the x-direction; realistically, the top plane which represents the surface should be a zero Dirichlet boundary condition, so sweeping from this direction is not an option, because there is no PML. In addition, since the majority of the reflections are oriented in the direction of the discontinuities, sweeping in this direction would kill some of these fields when approximating the half-space problem. It is more prudent to sweep orthogonal to the reflected rays, as pushing the PML to the domain of interest would not remove the contribution of fields that are returning to the domain.

Figures 6.4 and 6.5 show the results for the Overthrust model. For each example, the relevant quantities listed are the frequency f, number of degrees of freedom N, setup time T_{setup}, GMRES time T_{solve}, number of iterations N_{iter}, and the number of cores N_{proc}. With the Helmholtz equation, the largest problem solved is the 20 Hz example, with about 116 million degrees of freedom on 2048 processors. Because the elastic wave equation contains three times



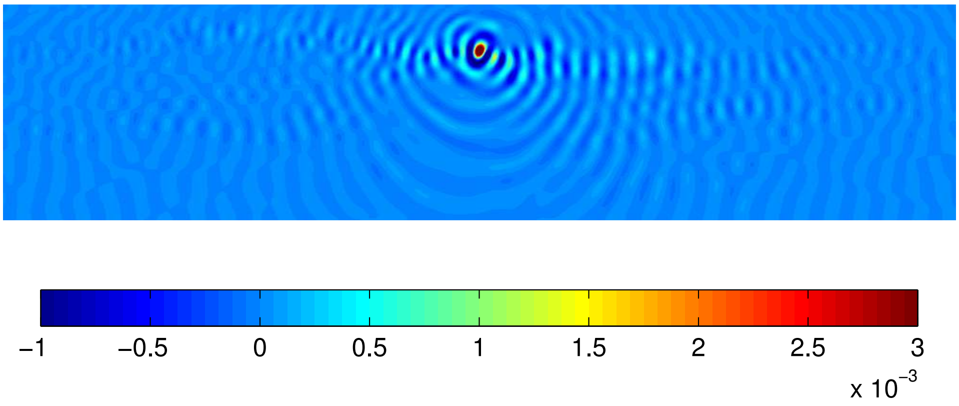
f	N	T_{setup}	T_{solve}	N_{iter}	N_{proc}
5	1.858e+06	46	46	10	32
10	1.463e+07	194	149	10	256
20	1.161e+08	563	910	11	2048

Figure 6.4: Results for the Helmholtz equation with the Overthrust model. The fields in the xz-plane at 20 Hz are shown.

the number of degrees of freedom and has a much denser sparsity pattern, and normal jobs on Lonestar only allow for 4104 processors, the largest problem solved for elasticity is the 10 Hz case.

6.6.2 Salt Dome Model

Another example of a commonly used velocity model is the SEG/EAGE Salt Dome. The model is identified by the large salt body in the middle of sedimentary layers; inside the salt body, the wave speed is very fast relative to the surrounding material. The dimensions of the model are 13.5 km × 13.5 km × 4 km , and the velocity data is given on a 676 × 676 × 210 grid. The minimum velocity is 1500 m/s, while the maximum velocity is 4482 m/s. Figure 6.6 shows a slice plot of the model.



f	N	T_{setup}	T_{solve}	N_{iter}	N_{proc}
5	4.511e+06	164	98	11	128
10	3.548e+07	670	468	12	1024

Figure 6.5: Results for the elastic wave equation with the Overthrust model. The fields in the yz-plane at 10 Hz are shown.

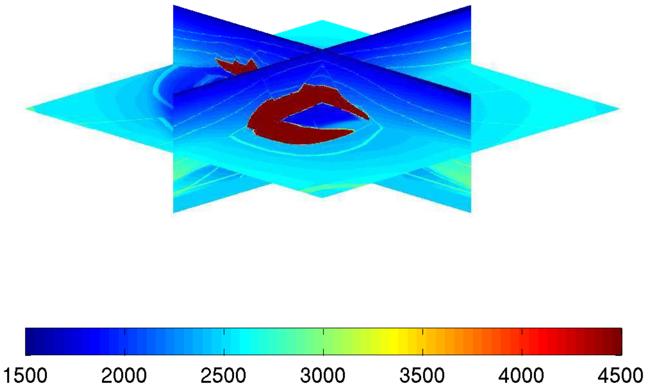
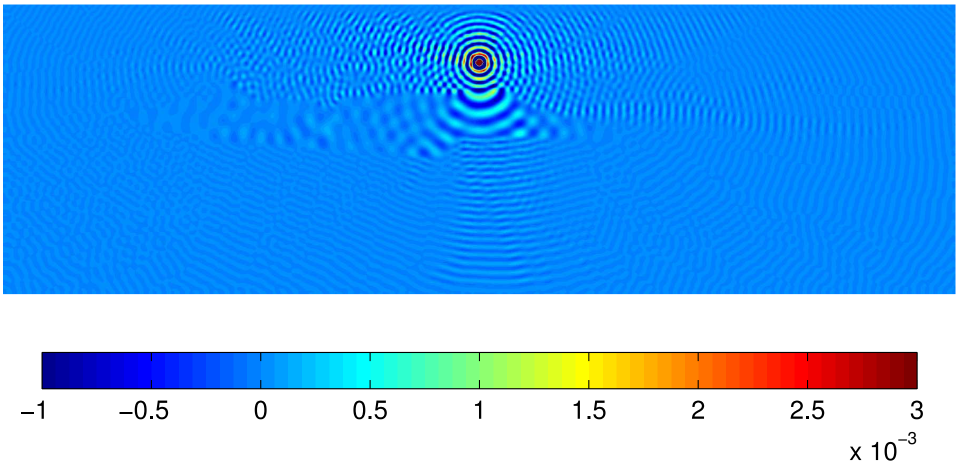


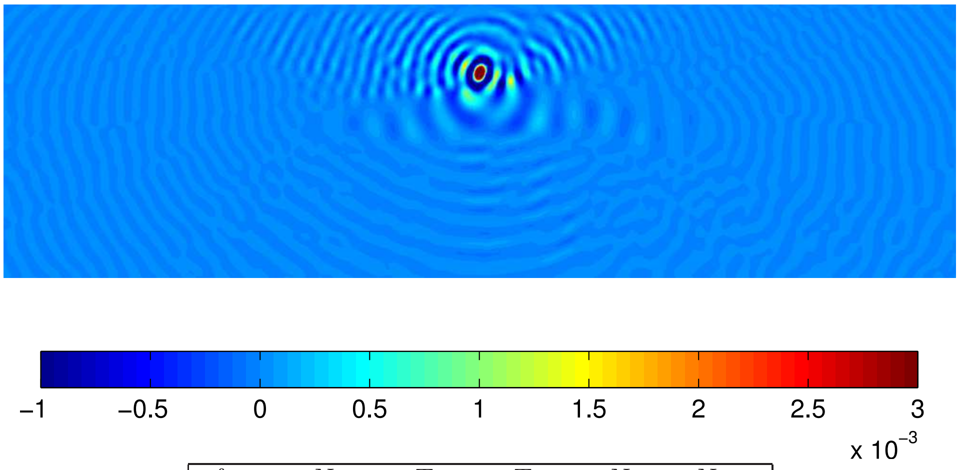
Figure 6.6: SEG/EAGE Salt Dome model. Velocity data is given in meters/second.

Figures 6.7 and 6.8 show the results for the Salt Dome model. For acoustics, the largest problem solved is the 20 Hz example, with about 88 million degrees of freedom on 2048 processors. Once again, the memory requirements are very demanding for the elasticity case, so only the 10 Hz example is given here, with about 33 million degrees of freedom on 2048 processors. In this example, the sweeping direction is taken to be the z-direction; regardless of this choice, however, it was observed that the number of iterations increased with frequency. One reason this happens is because of the large difference in the velocity between the salt body and surrounding area; it is difficult to resolve the smallest wavelength with enough grid points while keeping the PML wide enough to dampen the largest wavelength. Secondly, because the salt body is significantly large compared to the wavelength in all directions, the secondary reflections inside the structure are not restricted to a particular dimension as in a layered medium like the Overthrust case. Because there are reflected rays returning to subdomains in both the horizontal and vertical directions, the moving PML does not approximate the Green's function as well for this problem.



f	N	T_{setup}	T_{solve}	N_{iter}	N_{proc}
5	1.405e+06	73	23	8	32
10	1.106e+07	203	71	10	256
20	8.782e+07	600	558	18	2048

Figure 6.7: Results for the Helmholtz equation with the Salt Dome model. The fields in the xz-plane at 20 Hz are shown.



f	N	T_{setup}	T_{solve}	N_{iter}	N_{proc}
5	4.198e+06	150	37	7	256
10	3.305e+07	431	221	11	2048

Figure 6.8: Results for the elastic wave equation with the Salt Dome model. The fields in the yz-plane at 10 Hz are shown.

Chapter 7

Approximability of the Green's Function with PML Boundary Condition

In approximating the Schur complements in the block LDL^t factorization, the perfectly matched layer is pushed to the edge of the domain where the solution of the local subproblem acts as a Green's function. Thus, it is important to understand the effect that shifting the PML has on the solution in this area. In this chapter, a theoretical result is provided on the Helmholtz Green's function with PMLs.

7.1 Spectral Analysis of the Green's Function

In domain decomposition, particularly for alternating Schwarz methods [40], analysis is usually done on the eigenfunctions after reducing the problem to one dimension with Fourier decomposition. Here, a similar approach outlined in [72] is followed. To start, consider the 2D Helmholtz equation on a semi-infinite domain $(-\infty, \infty) \times [-D, D]$ with Dirichlet boundary conditions at $y = \pm D$ and radiation conditions as $x \to \pm\infty$,

$$\begin{aligned}
\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \kappa^2 u &= 0 && \text{in } (-\infty, \infty) \times [-D, D] \\
u &= 0 && \text{at } y = \pm D \\
\lim_{|x| \to \infty} \left[\frac{\partial u}{\partial |x|} - \imath \kappa u \right] &= 0.
\end{aligned}$$

Taking the Fourier transform of the PDE in the x-variable defined by

$$U(\lambda, y) = \int_{-\infty}^{+\infty} u(x, y) e^{\imath \lambda x} dx,$$

the 2D problem is reduced to the 1D problem for every eigenfrequency λ,

$$\begin{aligned}
\frac{\partial^2 U}{\partial y^2} + (\kappa^2 - \lambda^2) U &= 0 && \text{in } [-D, D] \\
U &= 0 && \text{at } y = \pm D.
\end{aligned}$$

The spectral Green's function $G_D(\lambda, y)$ for a source on the line $y = 0$ with zero Dirichlet boundary conditions at $y = \pm D$ is then the solution to

$$\begin{aligned}
\frac{\partial^2 G_D}{\partial y^2} + (\kappa^2 - \lambda^2) G_D &= \delta(y) && \text{in } [-D, D] \\
G_D &= 0 && \text{at } y = \pm D.
\end{aligned}$$

and is found to be

$$G_D(\lambda, y) = \frac{\sin\left(\sqrt{\kappa^2 - \lambda^2}(|y| - D)\right)}{2\sqrt{\kappa^2 - \lambda^2} \cos\left(\sqrt{\kappa^2 - \lambda^2} D\right)}.$$

It is clear that the zero Dirichlet boundary condition is not the problem of interest; to extend this to the PML, the function can be analytically continued into the complex plane by shifting the real coordinate y to be complex [72].

Thus, instead of D, the zero boundary condition is set at $D + \imath\alpha$, where α is the maximum damping value which the PML reaches. The spectral Green's function $G_{D+\imath\alpha}(\lambda, y)$ with PMLs is then

$$G_{D+\imath\alpha}(\lambda, y) = \frac{\sin\left(\sqrt{\kappa^2 - \lambda^2}(|y| - D - \imath\alpha)\right)}{2\sqrt{\kappa^2 - \lambda^2}\cos\left(\sqrt{\kappa^2 - \lambda^2}(D + \imath\alpha)\right)},$$

and the spatial Green's function $g_{D+\imath\alpha}$ with PMLs can be written as the inverse Fourier transform,

$$g_{D+\imath\alpha}(x, y) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{\sin\left(\sqrt{\kappa^2 - \lambda^2}(|y| - D - \imath\alpha)\right)}{2\sqrt{\kappa^2 - \lambda^2}\cos\left(\sqrt{\kappa^2 - \lambda^2}(D + \imath\alpha)\right)} e^{-\imath\lambda x} d\lambda.$$

7.2 Main result for moving PMLs

The theorem presented in this section will show that the Green's function for the semi-infinite domain $(-\infty, \infty) \times [-D, D]$ can be approximated well with the Green's function for the truncated semi-infinite domain $(-\infty, \infty) \times [-d, d]$, on the line $y = 0$.

Theorem 7.2.1. *Suppose $D > 0$ and $\alpha > 0$. For any $\varepsilon > 0$, there exists $\delta(\varepsilon) > 0$ such that $\delta(\varepsilon) < D$ and*

$$d > \delta(\varepsilon) \quad \Longrightarrow \quad \left|g_{D+\imath\alpha}(x, 0) - g_{d+\imath\alpha}(x, 0)\right| < \varepsilon. \tag{7.1}$$

Proof. Using trigonometric identities, the difference between the two Green's functions is

$$g_{D+\imath\alpha}(x, 0) - g_{d+\imath\alpha}(x, 0) =$$

$$\frac{1}{2\pi}\int_{0}^{\infty} \frac{\sin\left(\sqrt{\kappa^2 - \lambda^2}(D - d)\right)e^{-\imath\lambda x}}{\sqrt{\kappa^2 - \lambda^2}\cos\left(\sqrt{\kappa^2 - \lambda^2}(D + \imath\alpha)\right)\cos\left(\sqrt{\kappa^2 - \lambda^2}(d + \imath\alpha)\right)} d\lambda$$

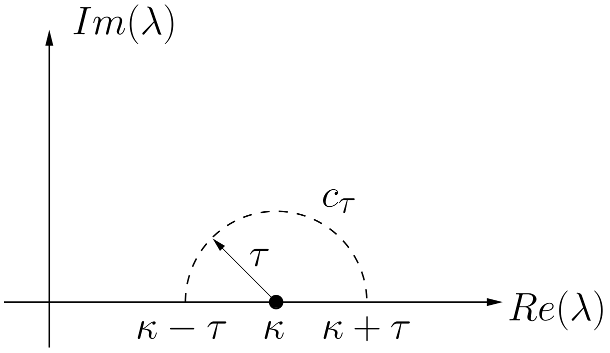


Figure 7.1: The contour path of integral (7.2).

This integral can be evaluated over the contour illustrated in figure 7.1, and is split into three parts:

$$\int_0^\infty \dots d\lambda = \int_0^{\kappa-\tau} \dots d\lambda + \int_{c_\tau} \dots d\lambda + \int_{\kappa+\tau}^\infty \dots d\lambda. \tag{7.2}$$

Since $\sin(x) \leq x$ for $x \leq 0$, the first integral can be bounded by

$$\begin{aligned}\Big|\int_0^{\kappa-\tau} &\frac{\sin\left(\sqrt{\kappa^2-\lambda^2}(D-d)\right)}{\sqrt{\kappa^2-\lambda^2}\cosh\left(\sqrt{\kappa^2-\lambda^2}(\imath D-\alpha)\right)\cosh\left(\sqrt{\kappa^2-\lambda^2}(\imath d-\alpha)\right)}d\lambda\Big| \\ &\leq \int_0^{\kappa-\tau} \frac{4(D-d)e^{-2\sqrt{\kappa^2-\lambda^2}\alpha}}{\left|(e^{2\sqrt{\kappa^2-\lambda^2}(\imath D-\alpha)}+1)(e^{2\sqrt{\kappa^2-\lambda^2}(\imath d-\alpha)}+1)\right|}d\lambda.\end{aligned}$$

But $\left|1+e^{-2\sqrt{\kappa^2-\lambda^2}\alpha}e^{2\imath\sqrt{\kappa^2-\lambda^2}D}\right| \geq 1-e^{-2\sqrt{\kappa^2-\lambda^2}\alpha}$ (equal when $\sqrt{\kappa^2-\lambda^2}=\frac{\pi}{2D}$), so

$$\begin{aligned}\left|\int_0^{\kappa-\tau}\dots d\lambda\right| &\leq \int_0^{\kappa-\tau} \frac{4(D-d)e^{-2\sqrt{\kappa^2-\lambda^2}\alpha}}{(1-e^{-\frac{\pi\alpha}{D}})(1-e^{-\frac{\pi\alpha}{d}})}d\lambda \\ &\leq \int_0^{\kappa-\tau} \frac{4(D-d)e^{-2\sqrt{2\kappa\tau-\tau^2}\alpha}}{(1-e^{-\frac{\pi\alpha}{D}})(1-e^{-\frac{\pi\alpha}{d}})}d\lambda \\ &= \frac{4(D-d)(\kappa-\tau)e^{-2\sqrt{2\kappa\tau-\tau^2}\alpha}}{(1-e^{-\frac{\pi\alpha}{D}})(1-e^{-\frac{\pi\alpha}{d}})}. \end{aligned} \tag{7.3}$$

For the integral over the arc c_τ, the polar coordinate transformation $\lambda = \kappa - \tau e^{-\imath\theta}$ for $\theta \in [0,\pi]$ is used. Denote $g(\theta) = 2\kappa\tau e^{-\imath\theta} - \tau^2 e^{-2\imath\theta}$. Because $|\sin(z)| \leq \sinh(|z|)$

for $z \in \mathbb{C}$,

$$\begin{aligned}
\left|\int_{c_\tau} \ldots d\lambda\right| &= \left|\int_0^\pi \frac{\imath\tau \sin\left(\sqrt{g(\theta)}(D-d)\right)e^{-\imath\theta}}{\sqrt{g(\theta)}\cos\left(\sqrt{g(\theta)}(D+\imath\alpha)\right)\cos\left(\sqrt{g(\theta)}(d+\imath\alpha)\right)}d\theta\right| \\
&\le \int_0^\pi \frac{\tau\sinh\left(\left|\sqrt{g(\theta)}\right|(D-d)\right)}{\left|\sqrt{g(\theta)}\cos\left(\sqrt{g(\theta)}(D+\imath\alpha)\right)\cos\left(\sqrt{g(\theta)}(d+\imath\alpha)\right)\right|} \\
&\le \int_0^\pi \frac{\tau\sinh\left(\sqrt{2\kappa\tau+\tau^2}(D-d)\right)}{\sqrt{2\kappa\tau-\tau^2}\left|\sinh\left(\mathrm{Im}(\sqrt{g(\theta)}(d+\imath\alpha))\right)\right|\left|\sinh\left(\mathrm{Im}(\sqrt{g(\theta)}(D+\imath\alpha))\right)\right|}.
\end{aligned}$$

Denote $\sigma(\alpha,d) = \min(\sqrt{2\kappa\tau-\tau^2}\alpha, \sqrt{2\kappa\tau+\tau^2}d)$. The argument of the hyperbolic sine in the denominator, $\mathrm{Im}(\sqrt{g(\theta)}(d+\imath\alpha))$, can be bounded below by $\sigma(\alpha,d)$; thus,

$$\begin{aligned}
\left|\int_{c_\tau} \ldots d\lambda\right| &\le \int_0^\pi \frac{4\tau\sinh\left(\sqrt{2\kappa\tau+\tau^2}(D-d)\right)e^{-\sigma(\alpha,d)}e^{-\sigma(\alpha,D)}}{\sqrt{2\kappa\tau-\tau^2}(1-e^{-2\sigma(\alpha,d)})(1-e^{-2\sigma(\alpha,D)})}d\theta \\
&= \frac{4\pi\tau\sinh\left(\sqrt{2\kappa\tau+\tau^2}(D-d)\right)e^{-\sigma(\alpha,d)}e^{-\sigma(\alpha,D)}}{\sqrt{2\kappa\tau-\tau^2}(1-e^{-2\sigma(\alpha,d)})(1-e^{-2\sigma(\alpha,D)})} \qquad (7.4)
\end{aligned}$$

From $|\cosh(x+\imath y)| = |\cosh(x)\cos(y)+\imath\sinh(x)\sin(y)| \ge \sinh(x)$, the third integral can be bounded by

$$\begin{aligned}
&\left|\int_{\kappa+\tau}^\infty \frac{\sinh\left(\sqrt{\lambda^2-\kappa^2}(D-d)\right)}{\sqrt{\lambda^2-\kappa^2}\cosh\left(\sqrt{\lambda^2-\kappa^2}(D+\imath\alpha)\right)\cosh\left(\sqrt{\lambda^2-\kappa^2}(d+\imath\alpha)\right)}d\lambda\right| \\
&\qquad \le \left|\int_{\kappa+\tau}^\infty \frac{\sinh\left(\sqrt{\lambda^2-\kappa^2}(D-d)\right)}{\sqrt{\lambda^2-\kappa^2}\sinh(\sqrt{\lambda^2-\kappa^2}D)\sinh(\sqrt{\lambda^2-\kappa^2}d)}d\lambda\right| \\
&\qquad \le \left|\int_{\kappa+\tau}^\infty \frac{2(e^{-2\sqrt{\lambda^2-\kappa^2}d}-e^{-2\sqrt{\lambda^2-\kappa^2}D})}{\sqrt{\lambda^2-\kappa^2}(1-e^{-2\sqrt{\lambda^2-\kappa^2}d})(1-e^{-2\sqrt{\lambda^2-\kappa^2}D})}d\lambda\right| \qquad (7.5)
\end{aligned}$$

Using the change of coordinates $z = \sqrt{\lambda^2-\kappa^2}$, this becomes

$$\begin{aligned}
\left|\int_{\kappa+\tau}^\infty \ldots d\lambda\right| &= \left|\int_{\sqrt{2\kappa\tau+\tau^2}}^\infty \frac{2(e^{-2zd}-e^{-2zD})}{\sqrt{z^2+\kappa^2}(1-e^{-2zd})(1-e^{-2zD})}dz\right| \\
&\le \frac{\left(\frac{e^{-2zd}}{-d}\Big|_{\sqrt{2\kappa\tau+\tau^2}}^\infty - \frac{e^{-2zD}}{-D}\Big|_{\sqrt{2\kappa\tau+\tau^2}}^\infty\right)}{(\kappa+\tau)(1-e^{-2\sqrt{2\kappa\tau+\tau^2}d})(1-e^{-2\sqrt{2\kappa\tau+\tau^2}D})} \\
&= \frac{\left(\frac{e^{-2\sqrt{2\kappa\tau+\tau^2}d}}{d} - \frac{e^{-2\sqrt{2\kappa\tau+\tau^2}D}}{D}\right)}{(\kappa+\tau)(1-e^{-2\sqrt{2\kappa\tau+\tau^2}d})(1-e^{-2\sqrt{2\kappa\tau+\tau^2}D})} \qquad (7.6)
\end{aligned}$$

Combining (7.3), (7.4), and (7.6), define the function $\mathrm{f}(\alpha, d)$ as

$$\begin{aligned} \mathrm{f}(\alpha, d) &= \frac{4(D-d)(\kappa-\tau)e^{-2\sqrt{2\kappa\tau-\tau^2}\alpha}}{(1-e^{-\frac{\pi\alpha}{D}})(1-e^{-\frac{\pi\alpha}{d}})} \\ &+ \frac{4\pi\tau\sinh\left(\sqrt{2\kappa\tau+\tau^2}(D-d)\right)e^{-\sigma(\alpha,d)}e^{-\sigma(\alpha,D)}}{\sqrt{2\kappa\tau-\tau^2}(1-e^{-2\sigma(\alpha,d)})(1-e^{-2\sigma(\alpha,D)})} \\ &+ \frac{\left(\frac{e^{-2\sqrt{2\kappa\tau+\tau^2}d}}{d} - \frac{e^{-2\sqrt{2\kappa\tau+\tau^2}D}}{D}\right)}{(\kappa+\tau)(1-e^{-2\sqrt{2\kappa\tau+\tau^2}d})(1-e^{-2\sqrt{2\kappa\tau+\tau^2}D})} \end{aligned} \tag{7.7}$$

Note that $\mathrm{f}(\alpha, d)$ is monotonically decreasing as d increases to D. It has been shown that

$$\left|\int_0^\infty \ldots d\lambda\right| < \mathrm{f}(\alpha, d) \qquad \text{for } \forall \quad \alpha > 0, \quad D > d > 0.$$

Thus, by the implicit function theorem, there exists a function $\mathrm{h}(\alpha, \varepsilon)$ such that $\mathrm{f}(\alpha, \mathrm{h}(\alpha, \varepsilon)) = \varepsilon$, which proves that

$$\left|\int_0^\infty \ldots d\lambda\right| < \varepsilon \qquad \text{if } d > \mathrm{h}(\alpha, \varepsilon).$$

□

To give an idea about how $\mathrm{f}(\alpha, d)$ behaves for a sample problem, a domain with $D = 100$ wavelengths is chosen, and the PML is pushed closer and closer to the line $y = 0$. Figure 7.2 shows the error bound on the Green's function at $y = 0$ as the parameter d is varied, with the maximum value of the complex shift in the PML set to $\alpha = 2\pi$ and contour radius set to $\tau = 0.1$. The plot shows that for $d \approx 5$ wavelengths, the upper bound on the error is $\mathrm{f}(\alpha, d) \approx 0.04$.

The result above shows that the Green's function for the semi-infinite problem with PMLs in the finite direction can be approximated by the truncated problem. To apply this result to the half-space approximation posed

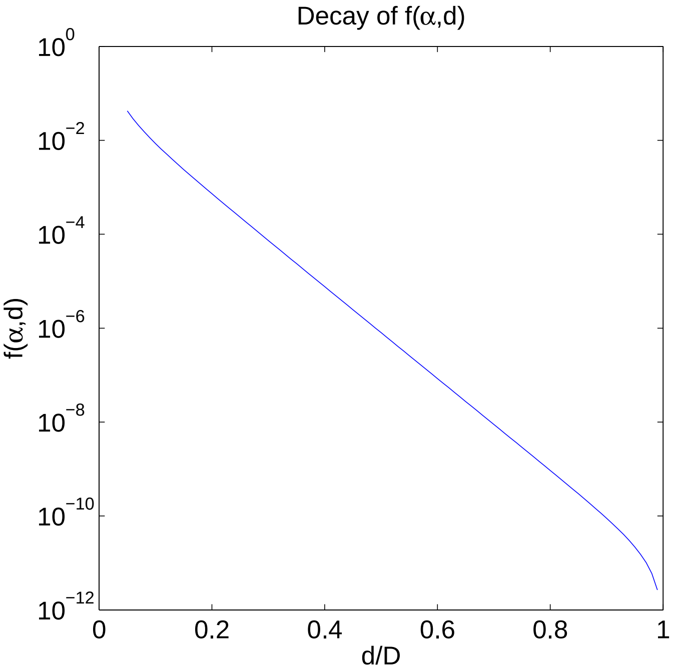


Figure 7.2: Decay of the error function $\mathrm{f}(\alpha, d)$ as $d \to D$. The parameters are $\alpha = 2\pi$, $\kappa = 200\pi$, $D = 100\lambda$, and $\tau = 0.1$.

for the sweeping preconditioner, image theory can be used. Specifically, it is well known that the Green's function for the half-space problem on the domain $(-\infty, \infty) \times [0, \infty)$ with zero Dirichlet boundary condition at $y = 0$ can be computed by $g(x, y) + g(x, -y)$, where $g(x, y)$ is the free space Green's function. Since the PML is an approximation to the radiation condition, the same image theory idea can be applied to approximating each PML half-space problem.

Chapter 8

Conclusions and Future Work

In this thesis, fast algorithms for solving linear systems in time-harmonic wave propagation were presented. For exterior scattering problems, the directional FMM was adapted to solve boundary integral equations in electromagnetics. An additional area of concern was uncertainty quantification and rough surface problems, where quasi-Monte Carlo methods were used to obtain better convergence to statistical quantities for high-frequency acoustic scattering. For variable coefficient media, sweeping preconditioners were generalized to unstructured meshes and higher-order finite element methods, with application problems such as electromagnetic cloaking and seismic wave propagation. Furthermore, theoretical justification of the sweeping preconditioner was provided by analyzing the Helmholtz Green's function with PML boundary conditions.

In light of these advancements, there are several topics which can be explored moving forward. One problem which the sweeping preconditioner suffers from is lack of parallelism in the apply stage. Unlike some domain decomposition methods, the approximation of the Schur complement relies on the structure of the block LDL^t factorization, so each Schur complement block can not be solved against independently from the others; that is, the right hand side of each local subproblem relies on the solution vector from the previous subproblem, which couples the subproblems across the whole domain. Although there exist more parallelizable methods for solving tridiagonal systems, like cyclic reduction, these destroy the structural properties of the Schur complement which are needed to justify the moving PML approximation.

Secondly, some analysis of the infinite-dimensional problem was done, but a discrete analysis of the Schur complement operator would be more useful in characterizing the eigenspectrum of the preconditioned matrix; presumably it will be dependent on factors such as the PML thickness, the maximum PML damping value, the complex perturbation parameter α, the number of layers being preconditioned, and the discretization. One way to approach this is through the idea of continued fractions. Because the Schur complement for the m-th subdomain is recursively defined through

$$S_m = A_{m,m} - A_{m,m-1} S_{m-1}^{-1} A_{m-1,m}$$

with $S_1 = A_{1,1}$, the second term in the formula can be expanded with matrix-valued continued fractions, i.e.

$$A_{m,m-1} \frac{1}{A_{m-1,m-1} - A_{m-1,m-2} \frac{1}{A_{m-2,m-2} - A_{m-2,m-3} \frac{1}{\ddots} A_{m-3,m-2}} A_{m-2,m-1}} A_{m-1,m}$$

To get an estimate on the condition number, a truncation of the continued fraction above must be inverted against the true Schur complement. If an upper bound can be computed on the norm of the product of these two fractions, then a useful estimate might be obtained. Unfortunately, many of the estimates for truncated scalar-valued continued fractions involve cutting off levels $1, ...n$ of the recursion, for some n, instead of removing the intermediary levels between the most dominant entry ($A_{m,m}$) and the first level ($A_{1,1}$); the latter is precisely what happens when the PML is pushed to the domain of interest. Therefore, obtaining a useful estimate using this approach has proven to be difficult.

Finally, one idea that has yet to be realized in code is the idea of a recursive or multilevel sweeping preconditioner. That is, if there is a PML in more than one direction, then each local subproblem itself can be solved using a sweeping preconditioner, with a sweeping direction orthogonal to the original sweep. Instead of computing the multifrontal factorization for quasi-2D slabs, the setup stage now requires the factorization of subproblems on long quasi-1D pillars. In the application stage, applying the Schur complement inverse for a particular layer would require an iterative solve of the subdomain problem. The main benefit of this algorithm is significant savings in memory; because the multifrontal method needs $O(N)$ memory in 1D and $O(N \log N)$ memory in 2D, there is a factor of $\log N$ in savings. The asymptotic complexity of the algorithm would also be slightly faster, but it is not entirely obvious if such a method would be more efficient in the frequency regime considered.

Index

Bibliography

[1] M. Ainsworth and H. A. Wajid. Optimally blended spectral-finite element scheme for wave propagation and nonstandard reduced integration. *SIAM J. Numer. Anal.*, 48(1):346–371, 2010.

[2] A. Alonso and A. Valli. An optimal domain decomposition preconditioner for low-frequency time-harmonic Maxwell equations. *Math. Computat.*, 68(226):607–631, 1999.

[3] P. R. Amestoy, I. S. Duff, J. Y. L'Excellent, and J. Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.*, 23(1):15–41, 2001.

[4] K.E. Atkinson. *Numerical solution of integral equations of the second kind*. Cambridge University Press, 1997.

[5] I. M. Babuska and S. A. Sauter. Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? *SIAM Review*, 42(3):451–484, 2000.

[6] H. Bagci, F.P. Andriulli, K. Cools, F. Olyslager, and E. Michielssen. A Calderon multiplicative preconditioner for the combined field integral equation. *IEEE Trans. Antennas Propag.*, 57(10):3387–3392, 2009.

[7] J. Barnes and P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324(4):446–449, 1986.

[8] U. Basu and A.K. Chopra. Perfectly matched layers for time-harmonic elastodynamics of unbounded domains: theory and finite element implementation. *Comput. Methods Appl. Mech. Eng.*, 192:1337–1375, 2003.

[9] A. Bayliss, C. I. Goldstein, and E. Turkel. An iterative method for the Helmholtz equation. *J. Comput. Phys.*, 49(3):443–457, 1983.

[10] M. Bebendorf. *Hierarchical matrices: a means to efficiently solve elliptic boundary value problems.* Springer, 2008.

[11] J. P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 114(2):185–200, 1994.

[12] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz. AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems. *Radio Science*, 31(5):1225–1251, 1996.

[13] A. Bossavit. *Computational electromagnetism: variational formulations, complementarity, edge elements.* Academic Press, 1997.

[14] J. Bramble and J. Pasciak. A note on the existence and uniqueness of solutions of frequency domain elastic wave problems: a priori estimates in H^1. *J. Math Anal. Appl.*, 345(1):396–404, 2008.

[15] J. Bramble, J. Pasciak, and D. Trenev. Analysis of a finite PML approximation to the three dimensional elastic wave scattering problem. *Mathematics of Computation*, 79(272):2079–2101, 2010.

[16] A. Brandt and I. Livshits. Wave-ray multigrid methods for standing wave equations. *Electronic Transactions on Numerical Analysis*, 6:162–181, 1997.

[17] Oscar P. Bruno and Leonid A. Kunyansky. A fast, high-order algorithm for the solution of surface scattering problems: basic implementation, tests, and applications. *J. Comput. Phys.*, 169(1):80–110, 2001.

[18] Russel E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. In *Acta numerica, 1998*, volume 7 of *Acta Numer.*, pages 1–49. Cambridge Univ. Press, Cambridge, 1998.

[19] E.J. Candes, L. Demanet, and L. Ying. A fast butterfly algorithm for the computation of Fourier integral operators. *Multiscale Model. Simul*, 7(4):1678–1694, 2009.

[20] C. Cecka and E. Darve. The black-box fast multipole method. *J. Comput. Phys.*, 228(23):8712–8725, 2009.

[21] K.M. Chen. A mathematical formulation of the equivalence principle. *IEEE Transactions on Microwave Theory and Techniques*, 37(10):1576–1581, 1989.

[22] Hongwei Cheng, William Y. Crutchfield, Zydrunas Gimbutas, Leslie F. Greengard, J. Frank Ethridge, Jingfang Huang, Vladimir Rokhlin, Norman Yarvin, and Junsheng Zhao. A wideband fast multipole method for the Helmholtz equation in three dimensions. *J. Comput. Phys.*, 216(1):300–325, 2006.

[23] W. C. Chew and W. H. Weedon. A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates. *Microwave Opt. Tech. Lett.*, 7(13):599–604, 1994.

[24] W.C. Chew, J. Jin, E. Michielssen, and J. Song. *Fast and efficient algorithms in computational electromagnetics*. Artech House, 2001.

[25] Clique. *version 0.1*. Jack Poulson, Austin, Texas, 2012.

[26] D.L. Colton and R. Kress. *Inverse acoustic and electromagnetic scattering theory*. Springer Verlag, 1998.

[27] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszynski, W. Rachowicz, and A. Zdunek. *Computing with hp-adaptive finite elements: three dimensional elliptic and Maxwell problems with applications*. Chapman and Hall/CRC, 2007.

[28] I.S. Duff and J.K. Reid. The multifrontal solution of indefinite sparse symmetric linear systems. *ACM Trans. Math. Software*, 9(3):302–325, 1983.

[29] M.G. Duffy. Quadrature over a pyramid or cube of integrands with a singularity at a vertex. *SIAM J. Numer. Analysis*, 19:1260–1262, 1982.

[30] H. C. Elman, O. G. Ernst, and D. P. O'Leary. A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations. *SIAM J. Sci. Comput.*, 23(4):1291–1315, 2001.

[31] B. Engquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. *Math. Computat.*, 31:629–651, 1977.

[32] B. Engquist and L. Ying. Fast directional multilevel algorithms for oscillatory kernels. *SIAM Journal on Scientific Computing*, 29(4):1710–1737, 2008.

[33] B. Engquist and L. Ying. A fast directional algorithm for high frequency acoustic scattering in two dimensions. *Commun. Math. Sci*, 7(2):327–345, 2009.

[34] B. Engquist and L. Ying. Sweeping preconditioner for the Helmholtz equation: hierarchical matrix representation. *Comm. Pure Appl. Math.*, 64:697–735, 2011.

[35] B. Engquist and L. Ying. Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers. *Multiscale Model. Simul.*, 9:686–710, 2011.

[36] Y. A. Erlangga. Advances in iterative methods and preconditioners for the Helmholtz equation. *Archives of Computational Methods in Engineering*, 15(1):37–66, 2008.

[37] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee. On a class of preconditioners for solving the Helmholtz equation. *Applied Numerical Mathematics*, 50(3-4):409–425, 2004.

[38] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee. A comparison of multigrid and incomplete LU shifted Laplace preconditioners for the inhomogeneous Helmholtz equation. *Applied Numerical Mathematics*, 56(5):648–666, 2006.

[39] O. G. Ernst and M. J. Gander. *Why it is difficult to solve Helmholtz problems with classical iterative methods*, volume 83, pages 325–361. 2011.

[40] M. J. Gander. Optimized Schwarz methods. *SIAM J. Numer. Anal.*, 44(2):699–731, 2006.

[41] M. J. Gander and F. Nataf. An incomplete LU preconditioner for problems in acoustics. *J. Comput. Acoust.*, 13:455–476, 2005.

[42] W.C. Gibson. *The method of moments in electromagnetics.* Chapman and Hall, 2007.

[43] J. Gopalakrishnan and J. E. Pasciak. Overlapping Schwarz preconditioners for indefinite time harmonic Maxwell equations. *Math. Computat.*, 72(241):1–15, 2001.

[44] J. Gopalakrishnan, J. E. Pasciak, and L. F. Demkowicz. Analysis of a multigrid algorithm for time harmonic Maxwell equations. *SIAM J. Numer. Anal.*, 42(1):90–108, 2004.

[45] L. Greengard, D. Gueyffier, P.G. Martinsson, and V. Rokhlin. A fast direct solver for integral equations in complex three-dimensional domains. *Acta Numerica*, 18:243–275, 2009.

[46] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348, 1987.

[47] Leslie Greengard. *The rapid evaluation of potential fields in particle systems.* ACM Distinguished Dissertations. MIT Press, Cambridge, MA, 1988.

[48] A. Gupta, G. Karypis, and V. Kumar. Highly scalable parallel algorithms for sparse matrix factorization. *IEEE Trans. Parallel and Dist. Systems*, 8(5):502–520, 1997.

[49] A. Gupta, S. Koric, and T. George. Sparse matrix factorization on massively parallel computers. In *Proc. of the Conference on High Performance Computing, Networking, Storage and Analysis*, Portland, OR, November 2009.

[50] I. Harari. A survey of finite element methods for time-harmonic acoustics. *Comput. Methods Appl. Mech. Eng.*, 195:1594–1607, 2006.

[51] I. Harari and U. Albocher. Studies of FE/PML for exterior problems of time-harmonic elastic waves. *Comput. Methods Appl. Mech. Eng.*, 195:3854–3879, 2006.

[52] R.F. Harrington. *Time-harmonic electromagnetic fields.* McGraw-Hill, 1961.

[53] R.F. Harrington. *Field computation by moment methods.* Wiley-IEEE Press, 1968.

[54] R. Hiptmair. Multigrid method for Maxwell's equations. *SIAM J. Numer. Anal.*, 36(1):204–225, 1998.

[55] R. Hiptmair. Finite elements in computational electromagnetism. *Acta Numerica*, 11:237–339, 2002.

[56] F. Ihlenburg. *Finite element analysis of acoustic scattering.* Springer-Verlag, 1998.

[57] S. Jiang, B. Ren, P. Tsuji, and L. Ying. Second kind integral equations for the first kind Dirichlet problem of the biharmonic equation in three dimensions. *J. Comput. Phys.*, 230(19):7488–7501, 2011.

[58] J. Jin. *The finite element method in electromagnetics.* Wiley-IEEE Press, 2002.

[59] D. Komatitsch and J. Tromp. Spectral-element simulations of global seismic wave propagation - I. Validation. *Geophysical Journal International*, 149(2):390–412, 2002.

[60] D. Komatitsch and J. Tromp. Spectral-element simulations of global seismic wave propagation - II. Three-dimensional models, oceans, rotation and self-gravitation. *Geophysical Journal International*, 150(1):303–318, 2002.

[61] R. Kress. *Linear integral equations.* Springer Verlag, 1990.

[62] L. Lin, J. Lu, L. Ying, R. Car, and W. E. Fast algorithm for extracting the diagonal of the inverse matrix with application to the electronic structure analysis of metallic systems. *Communications in Mathematical Sciences*, 7(3):755–777, 2009.

[63] J. Liu. The multifrontal method for sparse matrix solution: theory and practice. *SIAM Review*, 34(1):82–109, 1992.

[64] P. G. Martinsson and V. Rokhlin. A fast direct solver for scattering problems involving elongated structures. *J. Comput. Phys.*, 221(1):288–302, 2007.

[65] P.G. Martinsson and V. Rokhlin. A fast direct solver for boundary integral equations in two dimensions. *J. Comput. Phys.*, 205(1):1–23, 2005.

[66] E. Michielssen, A. Boag, and W. C. Chew. Scattering from elongated objects: Direct solution in O(N $\log^2$ N) operations. *IEEE Proc. Microw. Antennas Propag.*, 143(4):277–283, 1996.

[67] P. Monk. *Finite element methods for Maxwell's equations.* Oxford University Press, 2003.

[68] G. Mur. Absorbing boundary conditions for the finite-difference approximation of time-domain electromagnetic field equations. *IEEE Trans. Electromag. Compat.*, 23:377–382, 1981.

[69] J. C. Nédélec. Mixed finite elements in $\mathbb{R}^3$. *Numerische Mathematik*, 35(3):315–341, 1980.

[70] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*, volume 63 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

[71] E.J. Nyström. Über die Praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Mathematica*, 54(1):185–204, 1930.

[72] F. Olyslager. Discretization of continuous spectra based on perfectly matched layers. *SIAM J. Appl. Math.*, 64(4):1408–1433, 2004.

[73] A.T. Patera. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *J. Comput. Phys.*, 54(3):468–488, 1984.

[74] J. B. Pendry, D. Schurig, and D. R. Smith. Controlling electromagnetic fields. *Science*, 312(5781):1780–1782, 2006.

[75] A.F. Peterson, S.L. Ray, and R. Mittra. *Computational methods for electromagnetics*. Wiley-IEEE Press, 2001.

[76] J.R. Phillips and J.K. White. A precorrected-FFT method for electrostatic analysis of complicated 3-D structures. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 16(10):1059–1072, 1997.

[77] J. Poulson, B. Engquist, S. Fomel, S. Li, and L. Ying. A parallel sweeping preconditioner for high frequency heterogeneous 3D Helmholtz equations. *Submitted.*

[78] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*. Oxford University Press, 1999.

[79] P. Raghavan. Efficient parallel sparse triangular solution with selective inversion. *Parallel Processing Letters*, 8(1):29–40, 1998.

[80] S.M. Rao, D.R. Wilton, and A.W. Glisson. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Trans. Antennas Propag.*, 30(3):409–418, 1982.

[81] V. Rokhlin. Rapid solution of integral equations of scattering theory in two dimensions. *J. Comput. Phys.*, 86(2):414–439, 1990.

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

[83] Y. Saad. *Iterative methods for sparse linear systems.* SIAM, 2003.

[84] J.M. Song and W.C. Chew. Multilevel fast multipole algorithm for solving combined field integral equations of electromagnetic scattering. *Microwave Opt. Tech. Lett.*, 10(1):14–19, 1995.

[85] J.M. Song, C.C. Lu, and W.C. Chew. Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects. *IEEE Trans. Antennas Propag.*, 45(10):1488–1493, 1997.

[86] A. Taflove and S.C. Hagness. *Computational electrodynamics: the finite difference time domain method.* Artech House, 2005.

[87] L. L. Thompson. A review of finite element methods for time-harmonic acoustics. *J. Acoust. Soc. Am.*, 119(3):1315–1330, 2006.

[88] P. Tsuji, B. Engquist, and L. Ying. A sweeping preconditioner for time-harmonic Maxwell's equations with finite elements. *Submitted.*

[89] P. Tsuji and L. Ying. A fast directional algorithm for high-frequency electromagnetic scattering. *J. Comput. Phys.*, 230(14):5471–5487, 2011.

[90] P. Tsuji, L. Ying, and D. Xiu. A fast method for high-frequency acoustic scattering from random scatterers. *International Journal for Uncertainty Quantification*, 1(2):99–117, 2011.

[91] S. Wandzura and H. Xiao. Symmetric quadrature rules on a triangle. *Comput. Math. Appl.*, 45(12):1829–1840, 2003.

[92] G.R. Werner and J.R. Cary. A stable FDTD algorithm for non-diagonal, anisotropic dielectrics. *J. Comput. Phys.*, 226(1):1085–1101, 2007.

[93] H. Whitney. *Geometric integration theory*. Princeton University Press, 1957.

[94] A.C. Woo, H.T. Wang, M.J. Schuh, and M.L. Sanders. Benchmark radar targets for the validation of computational electromagnetics programs. *IEEE Antennas and Propagation Magazine*, 35(1):84–89, 1993.

[95] D. Xiu. Fast numerical methods for stochastic computations: a review. *Comm. Comput. Phys*, 5(2–4):242–272, 2009.

[96] D. Xiu and J.S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing*, 27(3):1118–1139, 2005.

[97] D. Xiu and J. Shen. An efficient spectral method for acoustic scattering from rough surfaces. *Communications in computational physics*, 2(1):54–72, 2006.

[98] Dongbin Xiu and George Em Karniadakis. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.*, 24(2):619–644 (electronic), 2002.

[99] L. Ying. *Fast algorithms for boundary integral equations*, volume 66, pages 139–194. 2009.

[100] L. Ying. Sparse Fourier transform via butterfly algorithm. *SIAM J. Sci. Comput*, 31(3):1678–1694, 2009.

[101] L. Ying, G. Biros, and D. Zorin. A kernel-independent adaptive fast multipole algorithm in two and three dimensions. *Journal of Computational Physics*, 196(2):591–626, 2004.

[102] Y. Zhu and A. C. Cangellaris. *Multigrid finite element methods for electromagnetic field modeling*. IEEE Press, 2006.