Copyright

by

Khoa Dang Tran

2012

The Dissertation Committee for Khoa Dang Tran

certifies that this is the approved version of the following dissertation:

Fast Numerical Methods for High Frequency Wave Scattering

Committee:

Bjorn Engquist, Supervisor

Hao Ling

Omar Ghattas

Richard Tsai

Lexing Ying

Fast Numerical Methods for High Frequency Wave Scattering

by

Khoa Dang Tran, B.S.C.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
May 2012

Dedicated to my parents.

Acknowledgments

A dissertation is considered to be the work of an individual. Even though this dissertation will be known as my own, it could not have been made possible without the contribution of several people who in their own ways have contributed to it significantly.

My deepest gratitude goes to Bjorn Engquist. His endless guidance, patience and vision inspired me to continue with my work on this paper. He was there to guide me during the entire time. His patience and willingness to share his vision helped me perceive the issues discussed in this paper in a new light. He also was willing to give me enough freedom to explore the various research ideas on my own.

I consider Lexing Ying as my second adviser. In addition to his deep involvement in my research by way of contribution of new ideas, he also taught me a few invaluable technical skills that I am sure will be of immense use to me.

It is also my duty to thank the committee members who have been instrumental by providing their valuable feedback on this dissertation.

I would also like to express my gratitude to all my colleagues whom I have had the pleasure of interacting with, both at the University of Texas at

Austin and also all over the world. My discussions with them have helped me greatly while working on this paper.

Finally, I would like to thank my family; my parents who have instilled in me a strong sense of passion and commitment, my wife Huong Thai for her care, encouragement and the rest of my family who have supported my ambitions and allowed me to continue with my academic pursuits. I thank them all.

Fast Numerical Methods for High Frequency Wave Scattering

Publication No. _____

Khoa Dang Tran, Ph.D.
The University of Texas at Austin, 2012

Supervisor: Bjorn Engquist

Computer simulation of wave propagation is an active research area as wave phenomena are prevalent in many applications. Examples include wireless communication, radar cross section, underwater acoustics, and seismology. For high frequency waves, this is a challenging multiscale problem, where the small scale is given by the wavelength while the large scale corresponds to the overall size of the computational domain. Research into wave equation modeling can be divided into two regimes: time domain and frequency domain. In each regime, there are two further popular research directions for the numerical simulation of the scattered wave. One relies on direct discretization of the wave equation as a hyperbolic partial differential equation in the full physical domain. The other direction aims at solving an equivalent integral equation on the surface of the scatterer. In this dissertation, we present three new techniques for the frequency domain, boundary integral equations.

The first development is the parallelization of the directional fast multipole method (DFMM) and its related high frequency wave solver. DFMM is an efficient matrix-vector multiplication algorithm, proposed by Engquist and Ying, in the context of wave equations. The need for this operation arises naturally when one solves the discretized wave equations using an iterative method. However, due to both CPU memory and wait-time constraints, it remains impractical to solve a very large scale wave scattering problem using a single machine. Hence, a distributed algorithm that can scale well to thousands or more processors is needed. This distributed algorithm is particularly important for the simulation of wave propagation at very high frequencies and in complex geometries.

The second development concerns high frequency wave scattering by convex objects, for which we developed an algorithm with $O(1)$ computational complexity with respect to the frequency. The convexity of each scatterer gives rise to a compact representation of the solution based on asymptotic analysis. Then, a localized integration technique is used for fast integral computations. These ideas, when combined, yield an $O(1)$ method. In the case of scattering by a convex object, our main contribution is a rigorous analysis and a more robust localized integration method compared to the original one proposed by Bruno *et al.* We also extended this $O(1)$ technique to the 2-body scattering problem using an iterative approach.

The final development is a hybrid method that uses an $O(1)$ representation of the solution on the convex part of the scatterer and fine-scale

representation on the rest of the geometry. This hybrid method combines the two recent and highly efficient algorithms mentioned above. While DFMM accelerates the computations where fine-scale resolution is needed, the localized integration technique provides control of the accuracy of the solution where high frequency asymptotics are used. Our hybridization technique provides smooth transition between the two regions through partition of unity and also allows a full two-way coupling in the overlapping regions. A typical future application could be a fine modeling of an antenna coupled to an asymptotic model for the building on which the antenna is attached.

Table of Contents

Acknowledgments v

Abstract vii

List of Tables xiii

List of Figures xv

Chapter 1. Introduction 1

1.1 Motivation 1

1.2 Numerical Analysis of Wave Equations: Background 2

1.2.1 Overview 2

1.2.2 Boundary integral equation for the Helmholtz equation . 9

1.2.3 High frequency asymptotics 21

1.3 Summary of Research Accomplishments and Outline of the Dissertation 25

Chapter 2. Fast Multipole Methods 28

2.1 FMM for Non-oscillatory Kernels 28

2.1.1 Introduction 28

2.1.2 The problem. 29

2.1.3 Approximation results for well-separated sets. 30

2.1.4 Multilevel approximation for mixed distribution. 32

2.1.5 Algorithm description. 36

2.2 Directional FMM for oscillatory kernels 38

2.2.1 Motivation 38

2.2.2 Directional low-rank property 40

2.2.3 Compact representation 41

2.2.4 Translation operators 43
2.2.5 Algorithm description 44

Chapter 3. Parallel DFMM and BIE Solver **47**
3.1 Introduction 47
3.2 Parallel Algorithm 49
3.2.1 DFMM 50
3.2.2 BIE solver 54
3.3 Numerical Tests 55
3.3.1 On the parallel DFMM 57
3.3.2 On the BIE solver 63
3.4 Concluding Remarks and Future Improvements 67

Chapter 4. GO-BIE Method **69**
4.1 Introduction 69
4.2 Asymptotic Representation 71
4.3 Localized Integration 74
4.4 Algorithm Description 88
4.5 Numerical Results 91
4.6 Extension to multibody scattering 95
4.6.1 Analysis 95
4.6.2 Preliminary results 99
4.7 Concluding Remarks 102

Chapter 5. Hybrid Method **106**
5.1 Motivation 106
5.2 Analysis 107
5.3 Algorithm 112
5.4 Numerical Results 115
5.5 Concluding Remarks 119

Chapter 6. Conclusions and Future Work **124**
6.1 Results. 124
6.2 Future Work 126

Appendix A. Implementation considerations for DFMM 131

Appendix B. Neumann series in multibody scattering 133

Nomenclature 135

Bibliography 136

List of Tables

3.1 **Strong scalability** results for a fixed-size problem: 64×16 cylinders, each of height $H = 4$. Each cylinder is discretized with 2252 points and the whole domain has approximately 2.3 million points. The largest side of the board is 256-wavelengths long and so the its covering box (root of the octree) has size $256 \times 256 \times 256$. Here #P is the number of processors; Total is the total time of a single interaction computation. L/H Up/Down are the times spent in the low/high frequency regimes while traveling up/down the tree. Comm is the time spent for MPI communications. We also report timing for the tree construction (in the Setup column). 58

3.2 **Strong scalability** results: the scaling factor of parallel DFMM (not counting the tree construction). We define scaling factor as the speedup in runtime for each quadrupling the number of processors. The test description is the same as in Table 3.1. . 58

3.3 **Weak scalability** results. The number of processors increases with respect to the size of the board so that each cylinder of height $H = 4$ is owned by a processor. There are 2252 points per cylinder. Total, L-Up, H-Up, H-Down, L-Down, Comm, and Setup have the same meanings as in Table 3.1. Board Dim is the board dimension, expressed in terms of cylinders per dimension. 62

3.4 **Strong scalability** results for a fixed-size problem: 64×16 cylinders, each of height $H = 2$. Each cylinder is discretized with approximately 7250 points and the whole domain has slightly more than 2.3 million points. For this problem and the prescribed error bound of 10^{-3}, the solver converges in 52 GMRES iterations. Here, we report the CPU time and scaling factor (defined in Table 3.2) of the program with respect to different numbers of processors used. 63

3.5 Number of GMRES iterations for different board configurations. In these tables, $M \times N$ represents the number of cylinders per dimension, while H denotes the height of each cylinder. 65

3.6 Weak scalability results for the parallel BIE solver. The number of processors used for each test equals the number of cylinders on the board. 66

4.1 The condition number of the induced matrix A is stably small with respect to k. 92
4.2 Accuracy of the GO-BIE method. 94
4.3 Error and CPU time (in seconds) for different numbers of iterations. The solvers were implemented in MATLAB and ran on a dual-core 2.8 GHz Intel CPU. The problem setting is described in Figure 4.15. 101
4.4 Error of the iterative method, using 10 iterations, for different frequencies. 103

5.1 Performance comparison between the hybrid and DFMM solvers with respect to the averaged time spent per iteration (in seconds) and also the number of iterations needed, when $L = O(k^{-1/2})$. The 10^{-3} error bound control was specified for both methods. While the DFMM solver did not finish the computations within a reasonable time frame, the hybrid solver produced the solution within a relatively short time. 117
5.2 Numerical error of the hybrid method and its CPU time, in comparison with the performance of the DFMM solver, for the case $L = O(k^{-1/2})$. The numerical error of a simulated solution is the discrete L_∞ error when compared against the corresponding "exact" solution, which was computed using the DFMM-based direct solver with 10^{-5} error tolerance. See Figure 5.8 for the plot of errors on the whole boundary. As in Table 5.1, we were not able to produce the results using the DFMM solver for $k > 6400$. 119
5.3 Accuracy of the hybrid method when $L = O(k^{-1})$ and when a fixed number of unknowns were used. The errors are computed by comparing the approximate solutions against the reference solutions, via the discrete L_2 error formula. The reference solutions were generated with the DFMM solver with 10^{-6} accuracy level. To obtain L_2 errors of less than 10^{-3}, we used 120 sampling points. The method converged in 14-15 iterations for all k and the whole program took on average 4.2 seconds to finish. For 10^{-5} error bound, we used 240 sampling points. The method converged in 19-21 iterations and the program finished within 38 seconds on average. 120

List of Figures

1.1 A simple illustration of a wave scattered by an obstacle. Here, u^{inc} denotes the incoming wave, and u^{scat} denotes the scattered wave. 2

1.2 An electromagnetic application of high frequency scattering. In this example, we are interested in simulating the wave scattered by an airplane and its radar cross section [Engquist]. 3

1.3 Exterior domain of the Helmholtz equation,$\Delta u + k^2 u = 0$, with Dirichlet condition, $u = -u^{\text{inc}}$. 10

1.4 Proof of Lemma 1.1 12

1.5 BIE method for exterior problems 14

1.6 Discretization in the Nyström method 18

1.7 Instead of directly approximating the highly oscillatory wave, geometric optics solves the phase and amplitude separately. . 22

1.8 Reflections and diffractions in the scattering of an incoming wave u^I by a smooth and convex obstacle. 25

2.1 Distribution of quadrature points $\{x_i\}$ on the boundary $\partial\Omega$. . 30

2.2 Interaction between two well-separated sets A and B [109]. Above: Direct computation of $\{u_i\}$ for all $x_i \in A$, via the interaction (2.2), takes $O(n^2)$ FLOPs. Below: if A are B are far apart, this matrix-vector multiplication can be approximated efficiently in three steps. The total number of FLOPs is $O(n)$. 31

2.3 The quadtree generated from the domain in Figure 2.1. Different levels of the quadtree are shown from left to right [109]. 33

2.4 Illustration of the interaction lists, colored gray, of boxes at different levels [109]. 34

2.5 (a) The quadtree constructed for point distribution on the boundary of a kite-shaped scatterer. (b) A box B in the high frequency regime, *i.e.* B has width larger than λ, and one of its directional wedges. The interaction between B and each of its wedges can be shown to be approximately low-rank. The points $\{e_q^{B,\ell}\}_{1\le q\le r_\varepsilon}$ and $\{c_p^{B,\ell}\}_{1\le q\le r_\varepsilon}$ are used in the separated approximation (2.4). 40

3.1 A potential application of the parallel directional fast multipole method and the related BIE solver: simulation of radio wave propagation in a forest. (a) Photograph of a loblolly pine forest in Bastrop, TX. (b) Tree distribution at site. Wave propagation through this forest was studied in [76]. 48

3.2 An illustration of partitioning a non-uniform set of points in the unit square using Lloyd's algorithm. 55

3.3 Model geometry for the numerical tests of the parallel DFMM and BIE solver. This board of multiple cylinders is a simplified model of a forest. 57

3.4 **Strong scalability** results. Top: the stacked bar chart of runtimes for different phases in the interaction computation. Bottom: the total time of the interaction computation displayed in log scale. The test description is the same as in Table 3.1. . . 59

3.5 **Weak scalability** results. The test configuration is described in Table 3.3. 62

3.6 The log-scale plot of CPU time against the number of processors, for the test configuration described in Table 3.4. 64

3.7 Weak scalability results: runtime of the parallel BIE solver against different board dimensions, for different heights H of cylinders. The number of processors used for each test equals the number of cylinders on the board. 66

4.1 Left: real part of the solution $\mu(x)$ to (4.1) corresponding to scattering off a unit circle. Right: real and imaginary parts of $\mu^s(x)$. The incoming wave incident has frequency $k = 100$. . . 72

4.2 Multiple wave reflections by a non-convex scatterer. 72

4.3 Reflections and diffractions in the scattering of an incoming wave u^I by a circular scatterer. 74

4.4 Top left: the shadow boundaries in the scattering by a circle. Top right: The boundary parameterization function has derivative of $O(k^{-1/3})$ at the shadow points. Bottom left: with uniform parameterization, the slow density μ^{s} has a sharp turn at $\frac{\pi}{2}$ when k is large. Bottom right: with the boundary parameterization in the above picture, μ^{s} has a bounded derivative with respect to k. For the circle geometry, we used $\xi(\theta) = \theta - \frac{k^{1/3}-1}{4} sin(2\theta) + \frac{k^{1/3}-1}{8} sin(4\theta)$ as the change of variable function, where θ corresponds to the parameterization angle of the boundary. 75

4.5 Cutoff function w_A for different values of A. 77

4.6 Localized integration: the integral of $e^{ik\phi(t)}w_1(t)$ (dense) on $[-1, 1]$ can be approximated by that of $e^{ik\phi(t)}w_\epsilon(t)$ on $[-\epsilon, \epsilon]$. 77

4.7 Convergence test of the localized integral approximation on $I(k)$ defined in (4.13). Left: with fixed k, the localized approximation of the integral has an error that decreases faster than any polynomial, with respect to $k\epsilon^2$, when ϵ increases. Right: with fixed $k\epsilon^2$, *i.e.* ϵ chosen to be proportional to $k^{-1/2}$, the error is approximately proportional to ϵ but the relative error stays constant. 82

4.8 (a) The smooth convex scatterer is decomposed into four parts: illuminated region, shadow region, and two boundary regions. (b) For a target point T, the localized integrator is applied at the three stationary points S_1, S_2, and S_3. 83

4.9 Using partition of unity, the integral of oscillatory function over the whole boundary can be split and localized to the neighborhoods of critical points. 84

4.10 Top: The plot of the phase $\phi(x, y)$ as a function of y when x is near the shadow point $\pi/2$. The phase behaves like a linear function on the left side of x, while it behaves like a cubic function on the right side. Bottom: The plot of the kernel $K(x, y)$ under the same setting. 86

4.11 Illustration of the localized integration in implementation. The cutoff windows of different widths are generated using the results from Theorem 4.5. Overlapping windows are made thinner and aligned to avoid over evaluation of the integral. 90

4.12 The solutions computed using the GO-BIE method for two different geometries. In both settings, the frequency $k = 100$. Left: scattering by a unit circle. Right: scattering by a more general convex object. Top row: illustration of the geometries. Middle row: the plot of the oscillatory density μ. Bottom row: the plot of the slow density μ^s. 93

4.13 Real and imaginary parts of μ^s from the scattering by a circular cylinder at $k = 100000$. Note the sharp transition of the function from the shadow to illuminated region. 95

4.14 A plane wave incident scattered by two circular obstacles . . . 96

4.15 Solution to the problem setting described in Figure 4.14. For this test, $k = 50$, Ω_1 has radius 1, Ω_2 has radius 2, and the distance between the two circles is 2 (*i.e.* the distance between the two centers is 5). Top: the plot of μ_1^s against the parametric angle of Γ_1; Bottom: the plot of μ_2^s against the parametric angle of Γ_2. The boundary density $\mu^s = \{\mu_1^s, \mu_2^s\}$ shown here was obtained using full-scale direct simulation. 100

4.16 The L_2 energy of the wave field, restricted to Γ, decays exponentially with respect to the number of geometrical optics reflections it has gone through. 101

4.17 Plot η_i^m $(i = 1, 2)$ for different values of m and i. 102

4.18 Convergence test of the iterative scattering method. 103

5.1 The scattering of an incoming plane wave u^I from a smooth scatterer Ω. u is the scattered field. The boundary Γ of the scatterer can be represented as a large convex part and a small structure (inside the small circle). 107

5.2 Real and imaginary parts of μ^s for the scattering from a circular cylinder with a small non-convex structure: μ^s is highly oscillatory in the small non-convex region due to multiple reflections. 108

5.3 (a) The boundary of the scatterer is represented as the union of Γ_1 and Γ_2. Γ_1 contains the small structure while Γ_2 covers the large convex part. (b) The sample points P_1 in the dense discretization of Γ_1. (c) The sample points P_2 in the sparse discretization of Γ_2. 109

5.4 The function $\mu_2(x)$ changes rapidly in the transition regions, due to the sharp cutoff $\alpha_2(x)$. 111

5.5 The smooth and periodic function τ transforms $M + N$ points sampled uniformly on Γ onto the same domain in a non-linear fashion. The first M points are squeezed into the small region Γ_1 while the rest are mapped into a much larger region. . . . 113

5.6 The real and imaginary parts of $\mu^s(x)$. The scatterer is given by the formula in (5.5) and $k = 100$. $\mu^s(x)$ is smooth in the convex part and is highly oscillatory in the small non-convex structure. 116

5.7 Growth of the CPU time per iteration and the number of iterations with respect to k, for both the DFMM and hybrid solvers, when $L = k^{-1/2}$. Top left: the CPU time per iteration using the DFMM solver has $O(k \log k)$ growth rate; Top right: the number of iterations using the DFMM solver has $O(k^{1/3})$ growth rate; Bottom left: the CPU time per iteration using the hybrid solver has $O(\sqrt{k} \log k)$ growth rate; Bottom right: the number of iterations using the hybrid solver has $O(k^{1/6})$ growth rate. The data used are from Table 5.1.
In each case, the growth is plotted in comparison with its theoretical growth $O(\sigma(k))$. Two parameters a and b are used for fitting the theoretical growth function $a\sigma(k) + b$. 118

5.8 The absolute value of the difference between the true solution and the solution computed using the hybrid method, at $k = 400$ and at $k = 3200$. 121

5.9 Growth of the total CPU time for both the DFMM and hybrid solvers. Left: the total CPU time using the DFMM solver has $O(k^{4/3} \log k)$ growth rate; Right: the total CPU time using the hybrid solver has $O(k^{2/3} \log k)$ growth rate; The data used are from Table 5.2. In each case, the growth is plotted in comparison with its theoretical growth $O(\sigma(k))$. Two parameters a and b are used for fitting the theoretical growth function $a\sigma(k) + b$. 122

Chapter 1

Introduction

1.1 Motivation

Numerical simulation of wave scattering is prevalent in many applications. Examples include the simulation of electromagnetic scattering, acoustics, seismology, medical imaging, quantum physics, and photonics. When the relative wave frequencies are high, and thus the wavelengths are short compared to the overall computational domain, this presents a computationally difficult multiscale problem. The small scale is given by the wavelength while the large scale corresponds to the overall size of the computational domain. Typically, 5 to 20 unknowns per wavelength are needed in each dimension for reasonable engineering accuracy.

This multiscale problem is an important study of great interest. For instance, in computational electromagnetic radiation and scattering problems, electromagnetic waves emitted by antennas, including radar devices, often have a very short wavelength compared to the size of the scatterer, which can be a building or an entire aircraft. Another typical application is in the design of antennas and their placement. For acoustic problems, high frequency

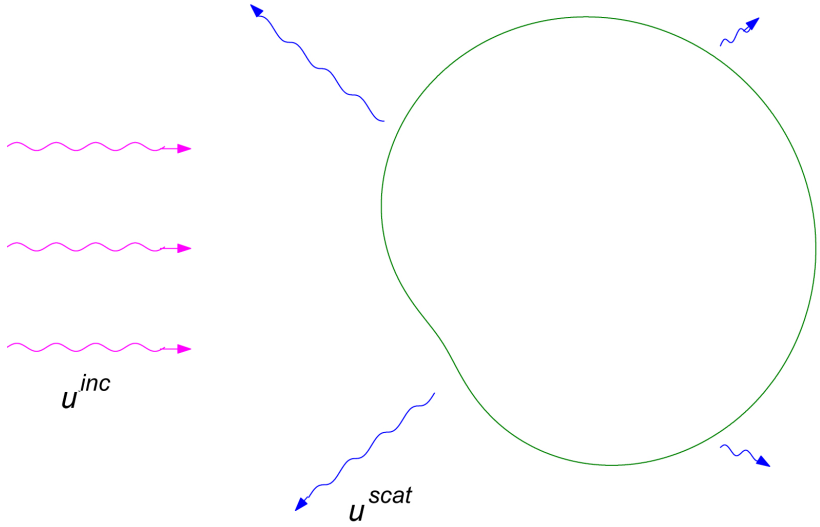

Figure 1.1: A simple illustration of a wave scattered by an obstacle. Here, u^{inc} denotes the incoming wave, and u^{scat} denotes the scattered wave.

techniques become interesting, for example, in underwater acoustics where waves of moderate frequency travel over very large distances.

Figure 1.1 gives a simplistic illustration of a wave scattering problem and Figure 1.2 displays a popular high frequency wave application application in electromagnetics.

1.2 Numerical Analysis of Wave Equations: Background

1.2.1 Overview

Here, we provide a brief overview of the study of wave equations and their approximations that are relevant to our research.

Wave equations are hyperbolic Partial Differential Equations (PDEs)

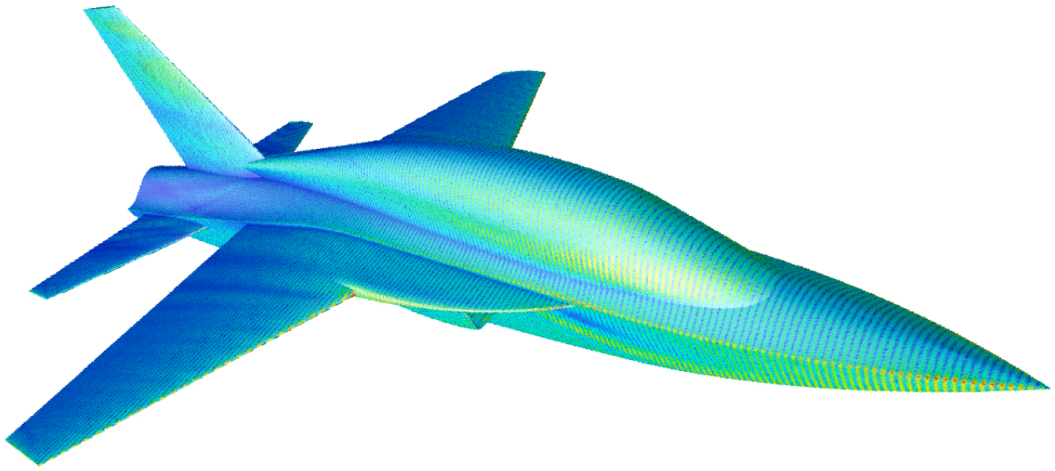

Figure 1.2: An electromagnetic application of high frequency scattering. In this example, we are interested in simulating the wave scattered by an airplane and its radar cross section [Engquist].

with finite propagation speed. The actual form that the waves take is not just defined by the equation but strongly dependent on the initial and boundary conditions.

While in some applications, the nonlinearity of the wave equation must be kept, in many other applications, linear models give quite accurate approximation to real life phenomena. In fact, a large body of research, including this doctoral work, is devoted to linear wave equations. For these equations, all the linear analysis tools can be applied, with Fourier analysis (expressing general solutions in terms of sums or integrals of well known basic solutions) being one of the most useful .

There are many linear wave models that are used in different applications [52, 55]. Some typical models are listed below.

- The canonical linear scalar wave equation is often used to describe a wide range of wave phenomena such as acoustic waves or vibration of a membrane:

$$u_{tt} - c(x)^2 \Delta u = 0, \quad (t, x) \in \mathbb{R}^+ \times \Omega, \tag{1.1}$$

where u may represent the acoustic pressure (for acoustic applications), c is the local speed of wave propagation of the medium, and $\Omega \subset \mathbb{R}^3$ is the physical domain where u is defined. Elastic and Maxwell equations can also be reduced to the linear scalar equation in the case of homogeneous medium.

- In electromagnetism, Maxwell equations are a system of equations that describe the coupling between the electric and magnetic fields, which oscillate in phase perpendicular to each other, and relate these fields to their sources, charge density, and current density. The Maxwell equations are presented mathematically as

$$\begin{aligned} \nabla \times E &= -\frac{\partial B}{\partial t}, \\ \nabla \times B &= \mu_0 \epsilon_0 \frac{\partial E}{\partial t} + \mu_0 J \end{aligned}$$

in $\mathbb{R}^+ \times \mathbb{R}^3 \setminus \Omega$, where

  - E and B represent the electric and magnetic fields, respectively,
  - ϵ_0 and μ_0 are permittivity (electric constant) and permeability (magnetic constant) of free space,

- J is the total current density (including both free and bound current).

- Elastic wave equation, used to model waves that propagate in elastic or viscoelastic materials (such as seismic wave propagation):

$$\rho u_{tt} - \nabla \cdot \sigma(x, \nabla u) = F,$$

  where

  - ρ is the mass density,
  - u is the unknown displacement vector,
  - σ is the (typically linear) Cauchy stress tensor,
  - F is the body force per unit volume.

Note that these wave models themselves are not high frequency. They are complemented with initial and boundary data, which generate high-frequency solutions.

Most wave equations cannot be solved analytically and therefore numerical methods must be resorted to. Numerical analysis of wave equations is itself a rich and broad research area. Hence, approximations of these equations are extensively studied to help design fast algorithms that can solve problems in practical time. These approximations also help scientists understand the

physics underlying those equations. In the following discussion of approximation methods, I will use (1.1) as the model equation. Similar approximations can be applied to other systems, such as Maxwell and elastic equations.

Approximations of linear waves can be classified into two main approaches [38].

1. One approach is to numerically solve the full wave model, usually after some dimension reductions [11, 22, 68, 37, 108]. The dimension reduction methods discussed below can be done either independently or combined, when applicable, in order to further reduce the complexity of the problem.

   (a) The most commonly used dimension reduction scheme is the reduction of time domain. In this case, the wave is approximated by a finite sum of harmonic components, each corresponding to a known frequency of the wave. Each of these components is governed by a so-called frequency domain wave equation, such as Helmholtz equation ($\Delta u(x) + k^2 u(x) = 0$ in $\mathbb{R}^d \setminus \Omega$ [1]). Direct discretization of the Helmholtz equation, via methods such as finite difference or finite element, leads to solving a sparse matrix system. An absorbing boundary condition or a perfectly matched layer needs to be applied to limit the size of those matrices [37, 11, 84, 77].

[1] See (1.2) for more details.

(b) Another common technique is the boundary integral equation method (also known as the method of moments), which uses the Green's function to transform the wave equation as a hyperbolic partial differential equation to a boundary integral equation (BIE) on the surface of the scatterer [22, 68]. The BIE technique is usually applied on top of frequency-domain equations. Time-domain BIEs also exist but will not be discussed in this dissertation. Since all of the research in this dissertation is based on the BIE framework, further background of this technique is provided in Section 1.2.2. Using the BIE framework, we end up solving a system of linear equations in which the induced matrix represents the Green's interaction among points on the boundary. Iterative methods, such as GMRES, are typically used to solve this matrix equation. Hence, an active line of research focuses on fast matrix-vector multiplication methods, notably ones based on the fast multipole method, originally developed by Greengard and Rokhlin [51, 50].

(c) There are also other settings that allow different ways of reducing the problem dimension. One such example is electromagnetic scattering by an *infinite* cylindrical structure which is illuminated by a field that does not vary along the axis of the cylinder. If the cylinder axis lies along the $\hat{z}$ axis of the Cartesian coordinate system, it is usually convenient to separate the fields into transverse magnetic (TM) and transverse electric (TE) parts with respect to

the variable z [92].

2. The other approach is to use asymptotic approximations of the solution when the frequency tends to infinity. Asymptotic methods, such as geometrical optics (GO) [69, 85, 39, 61, 78, 90, 89], geometrical theory of diffraction (GTD) [67, 82], or Gaussian beams [101, 107], are fast because they do not depend on the multiscale nature of the problem. For example, one can represent the solution to the Helmholtz equation as

$$u(x) = \sum_{i=1}^{M} A_i(x) e^{ik\phi_i(x)}$$

and approximate the amplitudes and the phases independently.

However, these methods do not allow accuracy control, *i.e.* the solution is not guaranteed to be be within a prescribed error tolerance. As a consequence, they are mostly used at very high frequencies in order to study physical properties of the solution. A major component of this research work involves high frequency asymptotics. Hence, further background will also be provided, in Section 1.2.3.

It is important to note that these two approaches are independent and not necessarily exclusively used. Asymptotic analysis can be applied at any level of dimension reduction. A popular example is the so-called GO-based boundary integral equation method (or GO-BIE method for short). This method, while using the boundary integral equation framework, utilizes the asymptotic expansion of the scattered wave to reduce the number of resolutions required to

represent this wave. This GO-BIE method, when applicable, is highly efficient while retaining the ability to control the accuracy of the computed solution. (See Chapter 4 for details.)

There are also so-called hybrid methods, which have gained much interest recently. Hybrid methods couple both direct and asymptotic techniques through domain decomposition [60, 81, 103]. In these methods, an asymptotic method is used on part of the domain while another full-scale method is needed on the other part. A transition scheme must be provided to connect the two methods.

1.2.2 Boundary integral equation for the Helmholtz equation

We will use the frequency domain Helmholtz equation with constant coefficients as the model equation to illustrate the BIE method, although the method can be applied to the time domain equation in a similar fashion.

The exterior Helmholtz problem can be stated mathematically as following. Suppose that $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ denotes a smooth impenetrable object with boundary $\Gamma = \partial\Omega$, $\Omega^E = \mathbb{R}^d \setminus \bar{\Omega}$ denotes the exterior domain, and $k = \frac{2\pi}{\lambda}$ denotes the wave frequency. We further assume that the incoming wave is given by $u^I(x) = e^{ik\alpha\cdot x}$ with $|\alpha| = 1$. The scattered wave field $u(x)$ then satisfies the following Helmholtz equation with Dirichlet (or sound soft) boundary

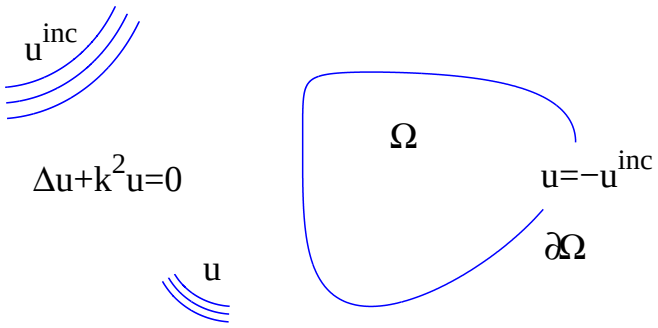

Figure 1.3: Exterior domain of the Helmholtz equation,$\Delta u + k^2 u = 0$, with Dirichlet condition, $u = -u^{\text{inc}}$.

condition

$$\begin{aligned} &\Delta u(x) + k^2 u(x) = 0 \quad \text{for } x \in \Omega^E \\ &u(x) = -u^I(x) \quad \text{for } x \in \Gamma \\ &\lim_{|x|\to\infty} |x|^{(d-1)/2} \left(\left(\frac{x}{|x|}, \nabla u(x) \right) - iku(x) \right) = 0, \end{aligned} \tag{1.2}$$

where the last condition is the Sommerfeld radiation condition, which guarantees that the scattered field $u(x)$ vanishes naturally towards infinity. The geometry of the problem is described in Figure 1.3.

The Boundary Integral Equation (BIE) method, also known as the method of moments, is a fundamental numerical technique used to compute solutions to Partial Differential Equations (PDEs) of a variety of physical problems with well-defined boundary conditions. The differential equation, which is defined over the entire problem domain, is transformed into an integral

equation over the boundary, or surface. The essential re-formulation of the PDE that underlies the BIE method consists of an integral equation that is defined on the boundary of the domain and an integral that relates the the boundary solution, called density, to the solution at points in the domain.

'The problem (1.2) can be formulated as an integral equation in a variety of ways, which are typically classified into direct and indirect approaches. In the direct approach, one applies the Green's function directly to the scattered wave and hence the unknown on the boundary represents a physical quantity. In the indirect approach, an artificial unknown on the boundary is used and its formulation can be chosen more freely. The direct approach has gained popularity recently due to the asymptotic property of the physical unknown quantity (more details are given Chapter 4). Here, we will present the direct formulation of the boundary integral equation.

Let G denote the standard free-space fundamental solution, also called the Green's function, of the Helmholtz equation (1.2). This Green's function is given, in 2D and 3D cases, by:

$$G(x,y) = \begin{cases} \frac{i}{4} H_0^{(1)}(k|x-y|) & (\text{for } d=2) \\ \frac{1}{4\pi} \frac{e^{ik|x-y|}}{|x-y|} & (\text{for } d=3) \end{cases}, \tag{1.3}$$

where $H_0^{(1)}$, in the two-dimensional case, is the Hankel's function of the first kind. The Green's function possesses the following important properties

- $G(x,y)$ is symmetric,

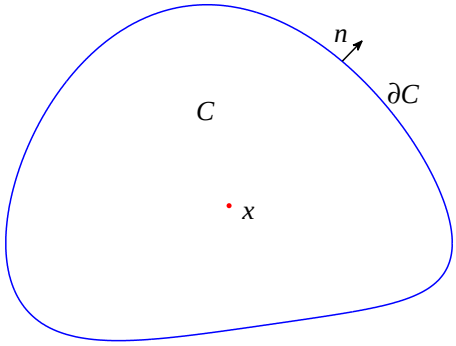

Figure 1.4: Proof of Lemma 1.1

- $G(x, y)$ is oscillatory,
- $(\Delta_x + k^2)G(x, y) = -\delta_y(x)$ and $(\Delta_y + k^2)G(x, y) = -\delta_x(y)$.

Lemma 1.1. *Let C be a bounded domain with smooth boundary (see Figure 1.4). Suppose that u is sufficiently smooth in $\bar{C}$ and satisfies $(\Delta + k^2)u = 0$ in C. Then:*

$$\int_{\partial C} \left(G(x,y)\frac{\partial u(y)}{\partial n(y)} - u(y)\frac{\partial G(x,y)}{\partial n(y)} \right) ds(y) = \begin{cases} u(x) & \text{for } x \in C, \\ 0 & \text{for } x \in \mathbb{R}^d \setminus \bar{C}. \end{cases} \tag{1.4}$$

Proof. Application of Green's theorem to both terms of the left hand side of

(1.4) gives:

$$\begin{aligned}
&\int_{\partial C}\left(G(x,y)\frac{\partial u(y)}{\partial n(y)}-u(y)\frac{\partial G(x,y)}{\partial n(y)}\right)\,ds(y)\\
&=\int_C\left(G(x,y)\Delta u(y)-u(y)\Delta_y G(x,y)\right)\,ds(y)\\
&=\int_C\left(G(x,y)(\Delta u+k^2u)(y)-u(y)(\Delta_y G+k^2G)(x,y)\right)\,ds(y)\\
&\quad\downarrow\ (\text{using } \Delta u+k^2u=0 \text{and } \Delta_y G+k^2G=-\delta_x)\\
&=\int_C u(y)\delta_x(y)\,ds(y).
\end{aligned}$$

The last expression vanishes if $x\in\mathbb{R}^d\setminus\bar{C}$ but equals $u(x)$ if $x\in C$ due to the property of the delta function. □

Lemma 1.2. *The scattered wave u in (1.2) satisfies the equation,*

$$u(x)=-\int_\Gamma\left(G(x,y)\frac{\partial u(y)}{\partial n(y)}-u(y)\frac{\partial G(x,y)}{n(y)}\right)\,ds(y) \tag{1.5}$$

for $x\in\Omega^E$.

Proof. Pick a large ball O that contains Ω (see Figure 1.5). Consider the domain $O\setminus\bar{\Omega}$. Let $\tilde{n}$ be the exterior normal of $O\setminus\bar{\Omega}$. From the previous lemma, we have

$$\begin{aligned}
u(x)=&\int_{\partial O}\left(G(x,y)\frac{\partial u(y)}{\partial\tilde{n}(y)}-u(y)\frac{\partial G(x,y)}{\partial\tilde{n}(y)}\right)\,ds(y)\\
&+\int_\Gamma\left(G(x,y)\frac{\partial u(y)}{\partial\tilde{n}(y)}-u(y)\frac{\partial G(x,y)}{\partial\tilde{n}(y)}\right)\,ds(y)
\end{aligned}$$

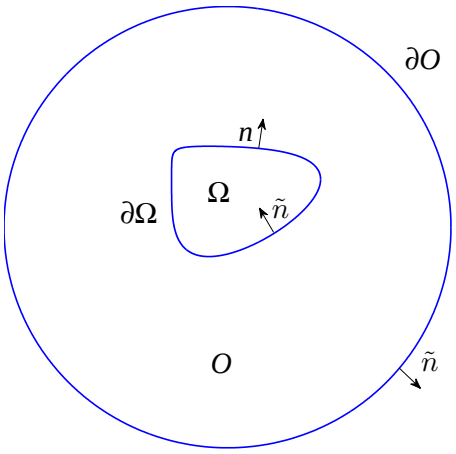

Figure 1.5: BIE method for exterior problems

for $x \in O \setminus \bar{\Omega}$. Pushing the radius of O to infinity and using the Sommerfeld radiation condition, we obtain

$$\begin{aligned} u(x) &= \int_\Gamma \left(G(x,y) \frac{\partial u(y)}{\partial \tilde{n}(y)} - u(y) \frac{\partial G(x,y)}{\partial \tilde{n}(y)} \right) ds(y) \\ &= - \int_\Gamma \left(G(x,y) \frac{\partial u(y)}{\partial n(y)} - u(y) \frac{\partial G(x,y)}{\partial n(y)} \right) ds(y). \end{aligned}$$

□

Consider a point in the exterior domain Ω^E. Application of Lemma 1.1 to u^I gives

$$0 = - \int_\Gamma \left(G(x,y) \frac{\partial u^I(y)}{\partial n(y)} - u^I(y) \frac{\partial G(x,y)}{\partial n(y)} \right) ds(y), \quad x \in \Omega^E. \tag{1.6}$$

Adding (1.5) and (1.6) side by side and using the boundary condition $u = -u^I$ on Γ, we get

$$u(x) = -\int_\Gamma G(x,y)\frac{\partial(u+u^I)}{\partial n}(y)\,ds(y), \quad x \in \Omega^E.$$

This can be rewritten as

$$u(x) = -\int_\Gamma G(x,y)\mu(y)\,ds(y), \quad x \in \Omega^E, \tag{1.7}$$

where

$$\mu(y) = \frac{\partial(u+u^I)}{\partial n}(y) \tag{1.8}$$

and μ solves a integral equation on Γ

$$\int_\Gamma G(x,y)\mu(y)\,ds(y) = u^I(x), \quad x \in \Gamma. \tag{1.9}$$

The equation (1.9) is called the single layer form for μ. Taking the normal derivative of (1.7) with respect to x on the boundary Γ and using the fact that $u = -u^I$ on Γ, we obtain double-layer form for μ

$$\theta(x)\mu(x) + \int_\Gamma \frac{\partial G(x,y)}{\partial n(y)}\mu(x)\,ds(y) = \frac{\partial u^I(x)}{\partial n(x)}, \quad x \in \Gamma, \tag{1.10}$$

where the coefficient $\theta(x)$ denotes the interior angle of the geometry at x. The extra term $\theta(x)\mu(x)$ appears in (1.10) due to the improper integral that involves the Green's function at singularity [31, 71]. As we assumed that Γ is

smooth, $\theta(x) = \frac{1}{2}$ for all $x \in \Gamma$.

In practice, one often uses the *combined layer form*

$$\frac{1}{2}\mu(x) + \int_\Gamma \left(\frac{\partial G(x,y)}{\partial n(x)} - i\gamma G(x,y) \right) \mu(y)\, ds(y) = \frac{\partial u^I(x)}{\partial n(x)} - i\gamma u^I(x), \quad (1.11)$$

where γ is a coupling parameter. The reason of using the combined layer form is that the induced matrix of (1.11) is well-conditioned for appropriate γ. Typically, we choose γ proportional to k. With this choice, the combined integral operator has a condition number that grows as $O(k^{1/3})$ with respect to k, for circular or convex geometries [70, 3, 12].

Based on the above formulation, the steps to solve the Helmholtz equation via the BIE approach are:

1. Solve μ, which satisfied the abstracted BIE

$$\frac{1}{2}\mu(x) + \int_\Gamma K(x,y)\mu(y)\, ds(y) = f(x), \quad x \in \Gamma, \quad (1.12)$$

where

$$K(x,y) = \frac{\partial G(x,y)}{\partial n(y)} - i\gamma G(x,y).$$

2. For point $x \in \Omega^E$, compute $u(x)$ with

$$u(x) = -\int_\Gamma G(x,y)\mu(y)\, ds(y).$$

Discretization. In order to solve (1.12) numerically, one often uses one of the following discretization methods: the Nyström method, the collocation method, and the Galerkin method. Our presentation of the three methods mostly follows the discussions in [109].

1. Nyström Method. The idea of the Nyström method is to approximate the infinite dimensional integral operators with finite dimensional quadrature operators. The steps are:

   - Approximate the integral operator $(K\mu)(x) := \int K(x,y)\mu(y)\,ds(y)$ with the quadrature operator

   $$(K_N\mu)(x) = \sum_{j=1}^{N} w_j K(x,x_j)\mu(x_j),$$

   where $\{x_j\}$ are the quadrature points and $\{w_j\}$ are the quadrature weights (see Figure 1.6). Here, we make the assumption that $\{w_j\}$ are independent of $\{x_j\}$. In practice, $\{w_j\}$ often depend on $\{x_j\}$, especially when x_j is in the neighborhood of x as to resolve the singularity of the kernel $K(x,y)$ at $y = x$.

   - Find μ such that $\frac{1}{2}\mu + K_N\mu = f$. We write this equation in the discrete form

   $$\frac{1}{2}\mu_i + \sum_{j=1}^{N} w_j K(x_i,x_j)\mu(x_j) = f(x_i), \qquad i = 1,2,...,N \tag{1.13}$$

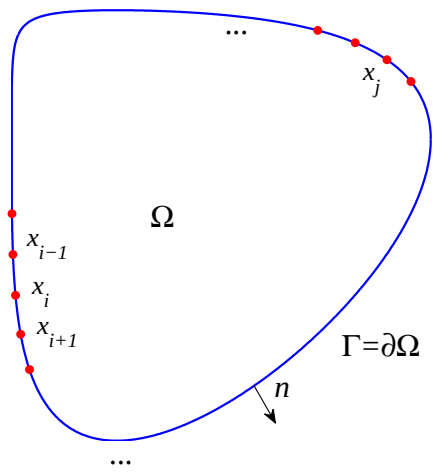

Figure 1.6: Discretization in the Nyström method

and solve for $\{\mu_i\}$.

- The value of $\mu(x)$ for $x \in \Gamma$ is computed using either interpolation of $\{\mu_i\}$ or using the quadrature operator

$$\mu(x) = 2\left(f(x) - \sum_{j=1}^{N} w_j K(x, x_j)\mu(x_j)\right).$$

2. Collocation Method. The idea of Collocation Method is to use subspace approximation. The steps are:

   - Approximate $\mu(x)$ by $\sum_{i=1}^{N} \mu_i \varphi_i(x)$ where $\{\varphi_i(x)\}_{i=1}^{N}$ are basis functions on Γ. Basis functions are typically chosen so that each φ_j is centered equals 1 at x_i. For such choice of basis functions, $\mu(x_i) = \mu_i$.

- Find $\{\mu_j\}_{j=1}^N$ such that $\frac{1}{2}\mu + K_N\mu = f$ is satisfied at $\{x_i\}_{i=1}^N$, *i.e.*

$$\frac{1}{2}\sum_{j=1}^{N}\mu_j\varphi_j(x_i) + \left(K\left(\sum_{j=1}^{N}\mu_j\varphi_j\right)\right)(x_i) = f(x_i), \qquad i = 1, 2, ..., N,$$

3. Galerkin Method. The idea of the Galerkin method is to use space approximation with orthogonalization. The steps are:

   - The approximation of $\mu(x)$ is similar to that in the collocation method.
   - Find $\{\mu_j\}_{j=1}^N$ such that $\frac{1}{2}\mu + K_N\mu - f$ is orthogonal to all the subspace generated by $\{\varphi_j\}_{j=1}^N$, *i.e.*

$$\left\langle \frac{1}{2}\sum_{j=1}^{N}\mu_j\varphi_j + \left(K\left(\sum_{j=1}^{N}\mu_j\varphi_j\right)\right) - f, \varphi_i \right\rangle = 0, \qquad i = 1, 2, ..., N.$$

The Nyström Method is ideal for smooth objects, due to its ease of implementation. However, the other two methods are more suitable when the geometries have edges or corners.

After the discretization step, the problem becomes solving a system of linear equations, for which the corresponding matrix is usually dense. For small problems, this can be done by Gaussian elimination. Larger problems are typically solved by iterative methods. In each iteration, a matrix-vector multiplication is performed. During the last decades, much attention has been devoted to the development of fast numerical algorithms for approximating

the matrix-vector multiplication of this type [13, 2, 16, 23, 19, 93, 17, 88]. One of the most popular class of such algorithms are the fast multipole methods (FMM) [51, 25, 29, 28]. Relevant fast multipole algorithms are described in Section 2.

Remark 1.3. The advantage of the BIE method arises from the fact that only a mesh over the modeled surface is required, while in the finite difference method (FDM) or finite element method (FEM), the whole domain of the PDE requires discretization. Thus, the dimension of the problem is effectively reduced by one. In cases where the domain is exterior to the boundary, as it is in scattering models, the extent of the domain is infinite and hence the advantages of the BIE method are even more striking: under the Sommerfeld's far field boundary condition, the equation governing the infinite domain is reduced to an equation over the finite boundary.

It is important to point out that the one-less-dimension advantage described above would not lead to a faster method unless it is powered by a fast matrix-vector algorithm. The reason is that direct discretization of the exterior domain, via DFM or FEM, yields a sparse matrix. This matrix is mathematically larger but contains significantly fewer interactions than the dense matrix given by the BIE method. We refer to [53] for a cost comparison of boundary integral equation and finite element methods. With the recent advances of fast multipole methods, this dense matrix disadvantage of the BIE approach has become less relevant.

Remark 1.4. The main weakness of the boundary integral equation method

is its dependence on the Green's function. For nonhomogenous problems, *i.e.* when k is a function of space, the Green's function is generally not known and hence the BIE method is not applicable.

1.2.3 High frequency asymptotics

At high frequency, the solution to the wave equation exhibits certain asymptotic properties, from which useful approximations can be made. There are different kinds of asymptotic approximations such as geometrical optics (GO) [69, 85, 39, 61, 78, 90, 89], physical optics (PO) [86, 85], geometrical theory of diffraction (GTD) [67, 82], and Gaussian beams [101, 107]. Here, we describe some aspects of GO and GTD that are related to our research.

In the short-wavelength limit, geometrical optics can be formulated in several different ways. One classical formulation uses the PDEs for the amplitude and phase of the solution. At sufficiently high frequency, the solution to the linear scalar wave equation can be approximated by a simple wave form,

$$u(t,x) \approx A(t,x)e^{i\omega\phi(t,x)/\epsilon} = \sum_{j=0}^{\infty} i^j \epsilon^j A_j(t,x) e^{i\phi(t,x)/\epsilon}. \tag{1.14}$$

As $\epsilon \to 0$, the phase function ϕ satisfies the Hamilton-Jacobi-type eikonal equation

$$\phi_t + c(x)|\nabla\phi| = 0, \tag{1.15}$$

where $|\cdot|$ denotes the Euclidean norm in $\mathbb{R}^d$ [43]. For the leading order

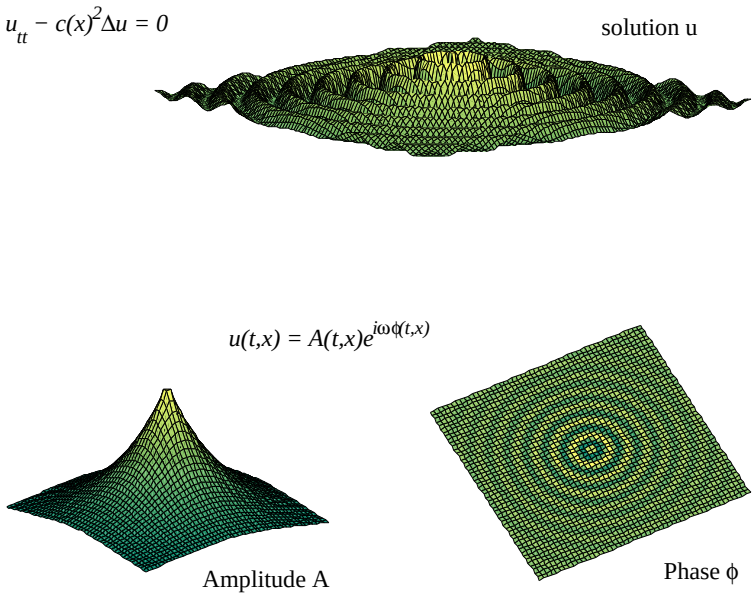

Figure 1.7: Instead of directly approximating the highly oscillatory wave, geometric optics solves the phase and amplitude separately.

amplitude term A, one obtains the transport equation

$$A_t + c(x)\frac{\nabla\phi \cdot \nabla A}{|\nabla\phi|} + \frac{c(x)^2\Delta\phi - \phi_{tt}}{2c(x)|\nabla\phi|}A = 0. \tag{1.16}$$

Solving for the phase ϕ and the amplitude A is easier than solving the full wave equation as these quantities do not depend on the frequency k (see Figure 1.7).

Geometrical optics can also be formulated in terms of ray tracing [49, 9, 73, 62, 26]. This is an ordinary differential equation (ODE) model. As long as the phase function is smooth, it corresponds to solving the eikonal equation

(1.15) through the method of characteristics, *i.e.* solving the system of ODEs

$$\frac{dx}{dt} = \nabla_p H(x,p), \quad \frac{dp}{dt} = -\nabla_x H(x,p), \quad H(x,p) = c(x)|p|, \tag{1.17}$$

where the momentum variable $p \in \mathbb{R}^d$ is usually called the slowness vector. There are also ODEs to compute the amplitude A.

One drawback of the geometrical optics is that diffraction effects at boundaries are lost. Geometrical theory of diffraction (GTD) is an improvement of GO that includes diffraction phenomena by explicitly taking into account the geometry of the object and boundary conditions. In the past decades, a lot of effort has been invested in this direction (see [67, 82] and references therein). According to these theories, the scattered field u in the Helmholtz equation (1.2) is approximated as a superposition of the form

$$u(x) = \sum_{i=1}^{M} A_i(x) e^{ik\phi_i(x)}, \tag{1.18}$$

This sum consists of reflected fields as well as diffracted fields. Each field $A_i(x)e^{ik\phi_i(x)}$ is a solution to the Helmholtz equation with different boundary conditions. We can use the differential equation methods described above to determine the phase and amplitude of each field.

For smooth and convex obstacles, another important result by Melrose and Taylor [82] shows that the normal derivative of the total field $\frac{\partial(u+u^I)}{\partial n}$ is asymptotically a slowly varying function. For such objects, there are three im-

portant regions with different properties, illustrated in Figure 1.8: the illuminated region, the shadow region, and the transitional shadow boundary region [58]. In the illuminated region, the scattered wave is described asymptotically by geometrical optics: a wave is reflected such that the angle of incidence and the angle of reflection are identical. A wave tangential to the shadow boundary causes diffraction. The density function decays rapidly away from the shadow boundary into the shadow region, due to the continuous emission of diffracted waves. In the deep shadow region, the density function vanishes. The asymptotic behavior of $\frac{\partial(u+u^I)}{\partial n}$ reflects these three regions, *i.e.* as $k \to \infty$,

$$\frac{\partial(u+u^I)}{\partial n} = \begin{cases} O(k) & \text{in the illuminated region,} \\ O\left(k^{2/3}\right) & \text{near the shadow boundary,} \\ O\left(e^{-k^{1/3}}\right) & \text{in the shadow region.} \end{cases}$$

Since we can also formulate the normal derivative density $\frac{\partial(u+u^I)}{\partial n}(x)$ as a solution to a boundary integral equation, its asymptotic property gives rise to a new fast method for solving the high frequency scattering problem by a convex obstacle. We provide our analysis and algorithm for this approach in Chapter 4.

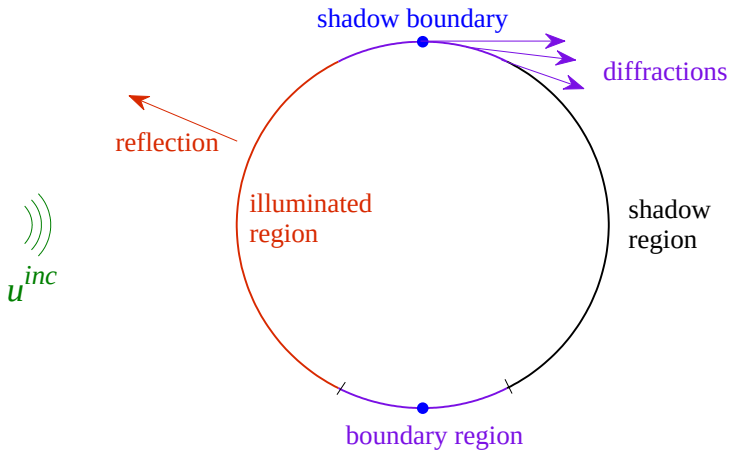

Figure 1.8: Reflections and diffractions in the scattering of an incoming wave u^I by a smooth and convex obstacle.

1.3 Summary of Research Accomplishments and Outline of the Dissertation

With the completion of this dissertation, we have accomplished the following tasks.

1. We have developed a new distributed algorithm, using message passing (via MPI), to solve the high frequency wave scattering problem based on the Directional Fast Multipole Method (DFMM) framework. The sequential DFMM algorithm, which was pioneered by the candidate's advisers, Bjorn Engquist and Lexing Ying, is efficient for solving a wave equation with fine scale discretization. However, due to both memory and run-time constraints, it remains impractical to solve a very large scale scattering problem (where the scatterer has size of more than 256

wavelengths) on a single processor. Our distributed algorithm can solve many realistic scattering applications that are in thousands or even tens of thousands of wavelengths. One such example is the simulation of wave propagation in cluttered environments.

2. We have analyzed and implemented a new robust GO-BIE method, used for high frequency waves scattered by a convex object. The method is most closely related to one proposed by Bruno *et al.* in [18]. However, their description of the localized-integration technique, used in the GO-BIE method, is ambiguous and a rigorous error analysis is not provided. We have extended this technique to make it robust and more generally applicable. We further devised a new iterative algorithm which exploits the GO-BIE technique for the 2-body scattering.

3. We have also developed a domain-decomposition hybrid method for solving wave scattering by objects which are convex except for a small complicated area. One typical example is the structure of a building with an antenna at the top. Our hybrid method couples the GO-BIE algorithm on the convex part of the scatterer's surface and DFMM on the rest of the surface. Unlike other hybrid methods, our method combines the two recent and highly efficient methods mentioned above. Our method also uses partition of unity to provide the smooth transition between the two parts of the domain. A typical future application would be cell phone wave scattering by a building with an antenna attached to it.

In this dissertation, we focus on the scalar Helmholtz equation (1.2). This equation is chosen as the model for our research because it contains difficult aspects of general high frequency scattering problems, while being simple enough to allow a substantial amount of numerical analysis.

Structure of the dissertation

The rest of this dissertation is structured as follows. In Chapter 2, we briefly review the central ideas of the fast multipole and directional fast multipole methods as background for the research on DFMM-based parallel algorithm, which is detailed in Chapter 3. In Chapter 4, we present the GO-BIE method. Our contribution starts appearing in Section 4.3, where we show the extended stationary phase lemma and provide the implementation of the new localized integration technique. In Chapter 5, we introduce the new hybrid method. Finally, in Chapter 6, we draw conclusions and discuss future work.

Chapter 2

Fast Multipole Methods

2.1 FMM for Non-oscillatory Kernels

2.1.1 Introduction

As we discussed in Section 1.2.2, a discretized system is often solved via an iterative method such as GMRES [95]. At each iteration, we face the problem of multiplying a dense matrix of size $N \times N$ by a vector of size N. Direct matrix-vector multiplication would take $O(N^2)$ floating point operations (FLOPs), which is costly in both memory usage and computational time for large N.

The Fast Multipole Method (FMM) originally developed by Greengard and Rohklin [51, 50] provides a numerical framework that dramatically reduces the complexity of matrix-vector multiplication involving a certain type of dense matrix, which arises out of many physical systems. Based closely on the discussion in [109], as a background this section gives insight to the underlying algorithm idea behind FMM for the Poisson kernel, which has the

analytic form:

$$G(x,y) = \begin{cases} \frac{1}{2\pi} \ln \frac{1}{|x-y|} & (d=2) \\ \frac{1}{4\pi} \frac{1}{|x-y|} & (d=3). \end{cases} \tag{2.1}$$

Because FMM does not apply directly to the Helmholtz kernel (1.3) due to its oscillatory characteristics, we selected the Poisson kernel to illustrate the idea behind the method. Just like the Helmholtz kernel, the Poisson kernel is the Green's function in the boundary integral formulation of the Laplace equation

$$\Delta u = 0, \quad x \in \Omega,$$

where Γ is a bounded domain in $\mathbb{R}^d$.

FMM, when applied to the Poisson kernel, reduces the complexity of matrix-vector multiplication from $O(N^2)$ to $O(N)$ for any fixed error tolerance.

2.1.2 The problem.

The matrix-vector multiplication problem is presented as follows. With reference to Figure 2.1, consider the inputs: a set of charges $\{f_i\}$ located at points $\{x_i\}$ and the Green's function $G(x,y)$ as in (2.1). We want to compute at each x_i the potential

$$u_i = \sum_{j=1}^{N} G(x_i, x_j) f_j \tag{2.2}$$

for all $i = 1, .., N$.

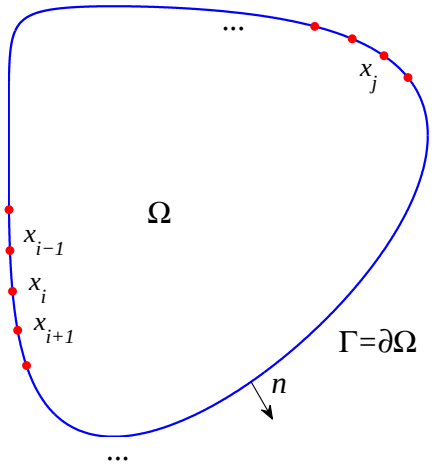

Figure 2.1: Distribution of quadrature points $\{x_i\}$ on the boundary $\partial\Omega$.

2.1.3 Approximation results for well-separated sets.

The two sets A and B are called *well-separated* if the distance between them is greater than their diameters. Figure 2.2 represents the interaction from a set of points $\{y_j\}$ in B to a set of points $\{x_i\}$ in A, where both A and B are subsets of $\{p_i\}$, which are distant apart from each other. Suppose that $\{f_j\}$ are the charges at $\{y_j \in B\}$ and $\{u_i\}$ are the potentials at $x_i \in A$ defined via the interaction (2.2). If both A and B contain $O(n)$ points then direct computation of the potentials$\{u_i\}$ takes $O(n^2)$ FLOPs.

However, we can approximate u_A using the three-step procedure:

$$u_i \approx u(C_A) = \sum_j G(C_A, y_j) f_j \approx G(C_A, C_B) \sum_j f_j, \tag{2.3}$$

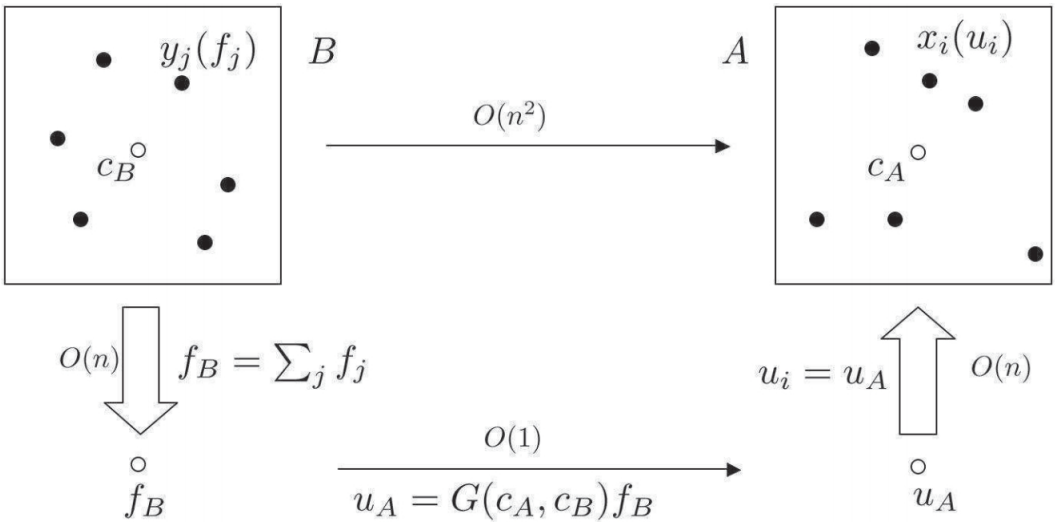

Figure 2.2: Interaction between two well-separated sets A and B [109]. Above: Direct computation of $\{u_i\}$ for all $x_i \in A$, via the interaction (2.2), takes $O(n^2)$ FLOPs. Below: if A are B are far apart, this matrix-vector multiplication can be approximated efficiently in three steps. The total number of FLOPs is $O(n)$.

where C_A is the center of A and C_B is the center of B. A geometric description of this approximation is given in Figure 2.2. We have also introduced two representations in this simple approximation:

- f_B, the *source representation* of B that allows one to approximate in the far field of B the potential generated by the source charges inside B.
- u_A, the *potential representation* of A that allows one to approximately reproduce inside A the potential generated by the source charges in the far field of A.

This three-step approximation is highly efficient and takes only $O(n)$

FLOPs. Here is a closer analysis:

- $f_B = \sum_j f_j$ takes $O(n)$ FLOPs.
- $u_A = G(C_A, C_B) f_B$ takes $O(1)$ FLOPs.
- $u_i = u_A$ for all $x_i \in A$ takes $O(n)$ FLOPs as well.

From an algebraic point of view, the three-step procedure is a rank-1 approximation of the interaction between A and B. This approximation is quite accurate when A and B are far away from each other and is in fact used quite often in computational astrophysics to compute the interaction between planets.

2.1.4 Multilevel approximation for mixed distribution.

The rank-1 approximation is only applicable for well-separated sets. If the distance between the two sets is comparable to their diameters, this approximation introduces significant error. Moreover, it cannot be applied directly to a cluster of points $\{p_i\}$ where each p_i can act both as a source and a target. The solution to such system is to use an adaptive tree structure, namely the octree in 3D or the quadtree in 2D. For consistency and simplicity, we will explain this method in a two-dimensional context.

Starting from the top level box containing all the points $\{x_i\}$, recursively divide every box in half along each dimension. This yields a natural quadtree structure, where the boxes obtained by subdivision of a box at higher

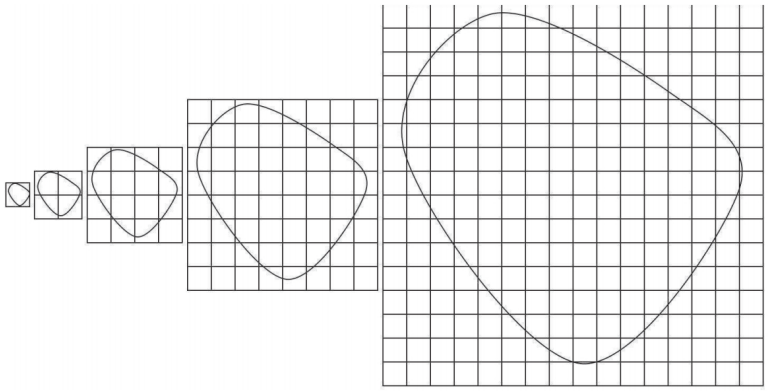

Figure 2.3: The quadtree generated from the domain in Figure 2.1. Different levels of the quadtree are shown from left to right [109].

level are considered its children. The refinement of each box is stopped when the number of points inside it is less than a prescribed small constant.[1] Figure 2.3 illustrates this domain partitioning process.

Our goal is to compute the interaction between every two leaf boxes of the quadtree. We begin the description of the fast multipole method under following assumptions and definitions:

1. The points $\{x_i\}$ are fairly homogeneously distributed or distributed almost uniformly on boundary Γ. The number of levels of the quadtree is thus $O(\log N)$.

2. For a given box B in the quadtree, all the adjacent boxes are said to be in the *near field* while the rest are in the *far field*.

[1] In practice this number can vary from 50 to 200, depending on the kernel function.

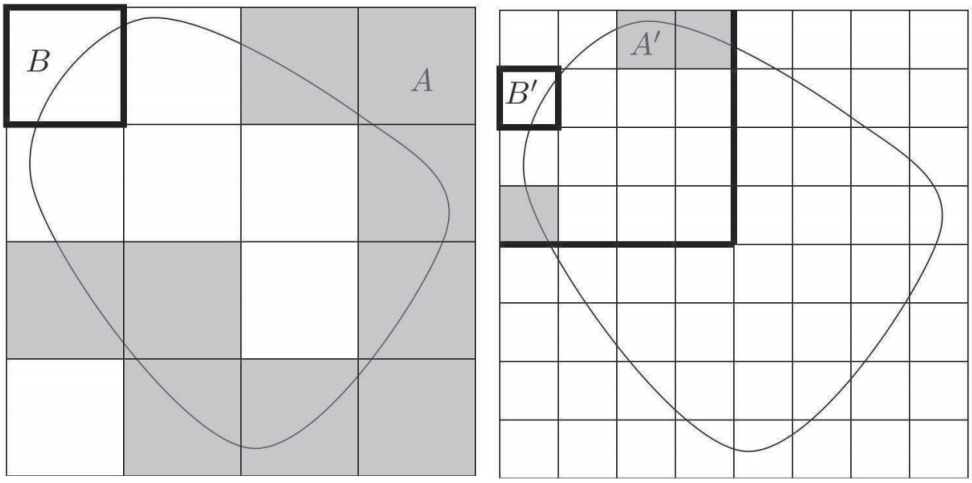

Figure 2.4: Illustration of the interaction lists, colored gray, of boxes at different levels [109].

3. The *interaction list* of B contains the boxes on the same level that are in its far field but not the far field of its parent. Note that the size of the interaction list is $O(1)$ for every B. Figure 2.4 shows the interaction list, colored gray, of a box B at the second level of the tree.

Here, we demonstrate the multilevel idea of the fast multipole method through a series of improvements, starting with direct matrix-vector multiplication.

1. **First improvement.** The potential u_i at x_i in a box A can be computed by adding the following independent contributions.

   (a) The far field contribution u_A^F generated by source points in boxes well-separated from A, collectively called F_A. The important point

here is that the far field contribution, also called the potential representation of A, is the same and only needs to be computed once for all $x_i \in A$.

(b) The near field contribution u_A^N from source points within A and in boxes adjacent to A. This contribution can be computed directly.

2. **Second improvement.** Suppose that A is a child box of A'. The local field representation u_A^F is composed of the potential generated by the interaction boxes of A and $u_{A'}^F$, which is the same for every child of A'. Therefore, it makes sense to compute the potential representations of parent boxes and then translate them to the potential representations of their children. This can be done recursively starting from the second-level boxes of the quadtree.[2]

   Then all we have left to compute are the following:

   (a) the source representation of interaction boxes for every box A at every level,[3]

   (b) and direct computation of the near field contribution u_A^N when A is the leaf box.

3. **Third improvement.** Since every box is in the interaction list of another box, the second improvement implies that we will eventually need

[2]Note that there are no well-separated boxes in the zeroth and first level.
[3]In the context of this section, by every box, we mean the boxes in level two or higher.

to compute the source representation for every box in level two or above. This can be done efficiently in a hierarchical manner starting from the leaf boxes.

(a) For a leaf box, its source representation can be aggregated from the sources at points inside it.

(b) For a non-leaf box, its source representation can be aggregated from the source representations of its children.

Previously, we used the summation of sources in B as the source representation of the B and a rank-1 approximation for the interaction between two well-separated sets. Such simplistic approximation does not give us the control of accuracy. However, we can still use a finite-rank approximation, *i.e.* a linear combination of more than one representative point sources for each box, to achieve any prescribed accuracy. For the purpose of this dissertation, we skip this analytic part of FMM. The computations of source representations in the far field with rigorous error bounds can also be done numerically using methods in [111, 110, 4].

2.1.5 Algorithm description.

Taking the above observations into consideration, the algorithm for evaluating the potentials $\{u_i\}$ can be summarized as follows:

1. Bottom-up traversal of the quadtree. At each level, for each box B,

- if leaf, compute f_B from the points in B,
- if non-leaf, compute f_B from the source representations of its children.

This step is called the *far-to-far translation*.

2. At each level, for each pair A and B in each other's interaction list, add $G(C_A, C_B)f_B$ to u_A. This step is called the *far-to-local translation*.

3. Top-down traversal of the quadtree. At each level, for each box A,

   - if leaf, add u_A to u_j for each point p_j in A,
   - if non-leaf, add u_A to the local field representations of its children.

   This step is called the *local-to-local translation*.

4. At the final level, for each leaf box A, compute the interaction with its adjacent boxes directly.

In order to estimate the complexity of this algorithm, the following points are considered.

- One far-to-far translation and one local-to-local translation is performed to each of the $O(N)$ boxes in the quadtree during the first and third steps. Since each of the far-to-far and local-to-local translations takes only $O(1)$ operations, the complexity of the first and the third steps is $O(N)$.

- In the second step, there are $O(N)$ boxes in the quadtree. Each box has $O(1)$ boxes in the interaction list. Because each far-to-local translation takes $O(1)$ operations, the complexity of the second step is $O(N)$.

- Finally, there are $O(N)$ leaf boxes in total, each having $O(1)$ neighbors. Since each leaf box contains only $O(1)$ points, the direct computation in the last step costs $O(N)$ FLOPs.

Therefore, the overall complexity of this algorithm is $O(N)$.

2.2 Directional FMM for oscillatory kernels

2.2.1 Motivation

An overview of the fast multipole algorithm in Section 2.1 shows that the key to this method lies in creating low-rank block approximations to the $N \times N$ matrix of interactions. It would not be wrong to say that the approximately low rank property of the kernel plays a fundamental role in FMM. This scenario changes with the Helmholtz equation. The interaction between two well-separated sets, through the Helmholtz kernel (1.3), is not approximately finite-rank. In fact, its rank is proportional to the square of the diameters of those sets.

This problem can be solved efficiently by the high frequency fast multipole method (HF-FMM) developed by Rokhlin *et al.* [93, 94, 28, 30, 32, 33, 87]. In short, after recursively partitioning the whole computational domain with

a quadtree, HF-FMM uses FFT-type techniques to accelerate the far field interaction in the high frequency regime resulting an optimal $O(N \log N)$ complexity. Though efficient and accurate, this method is highly technical.

On the other hand, Engquist and Ying recently have developed a new approach which aims to recover the low rank property of the oscillatory interaction through the Helmholtz kernel [42, 41]. This so-called directional fast multipole method (DFMM) is more general and also more scalable (*i.e.* easier to parallelize) than the HF-FMM. In this section, we briefly discuss DFMM as the background for the majority of our research.

Let us assume that the integral equation (4.1) is solved with the Nyström discretization scheme. With a few unknowns per wavelength, the integral in (4.1) is discretized with a set of $N = O(k)$ equally spaced quadrature points $\{x_i\}_{1 \le i \le N}$ on Γ. Let $\{\mu_i\}$ to be the approximations to $\{\mu(x_i)\}$.

The directional fast multipole method proposed in [42, 41] evaluates all $\{u_i\}_{i=1}^N$ in $O(N \log N) = O(k \log k)$ FLOPs. It starts by building an adaptive quadtree for the whole scatterer (see Figure 2.5(a)). The whole domain is partitioned dyadically and recursively until all of the leaf boxes have side length equal to $\lambda = 2\pi/k$. Since the boundary Γ is smooth and only a constant number of points is used per wavelength, each leaf box contains only a small number of quadrature points x_i.

2.2.2 Directional low-rank property

For a box B with side length $w\lambda$, its far field F^B is defined as the region that is $O(w^2\lambda)$ away from B and its near field N^B is the complement of F^B. A box A is in the interaction list of B if A is in B's far field but not B's parent's far field. The far field F^B is further partitioned into a group of $O(w)$ *directional wedges* $\{W^{B,\ell}\}$, each contained in a cone with center direction ℓ and spanning angle $O(1/w)$ (see Figure 2.5(b)).

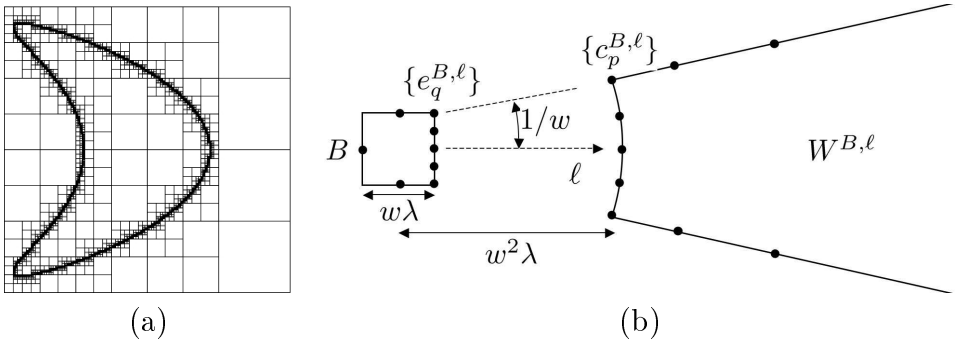

Figure 2.5: (a) The quadtree constructed for point distribution on the boundary of a kite-shaped scatterer. (b) A box B in the high frequency regime, *i.e.* B has width larger than λ, and one of its directional wedges. The interaction between B and each of its wedges can be shown to be approximately low-rank. The points $\{e_q^{B,\ell}\}_{1\le q\le r_\varepsilon}$ and $\{c_p^{B,\ell}\}_{1\le q\le r_\varepsilon}$ are used in the separated approximation (2.4).

The main idea behind DFMM is a directional low rank property of the Helmholtz kernel between B and each wedge $W^{B,\ell}$. It can be shown that, for any prescribed accuracy ε, there exists a rank-r_ε separated approximation of

$G(x,y)$ for $y \in B$ and $x \in W^{B,\ell}$:

$$\left| G(x,y) - \sum_{q=1}^{r_\varepsilon} G(x, e_q^{B,\ell}) \sum_{p=1}^{r_\varepsilon} d_{qp}^{B,\ell} G(c_p^{B,\ell}, y) \right| \leq \varepsilon, \tag{2.4}$$

where $\{y_s^{B,\ell}\}_{1\leq q\leq r_\varepsilon} \subset B$ are called the *directional equivalent source points* and $\{c_q^{B,\ell}\}_{1\leq q\leq r_\varepsilon} \subset W^{B,\ell}$ are called the *directional check target points* of B in direction l [41, 20, 44, 83]. The points $\{y_s^{B,\ell}\}_{q=1}^{r_\varepsilon}$, $\{x_p^{B,\ell}\}_{q=1}^{r_\varepsilon}$, and the matrix $D^{B,\ell} = (d_{qp}^{B,\ell})_{1\leq p,q\leq r_\varepsilon}$ can be computed efficiently by a randomized procedure [42, 63, 7, 8, 80]. Note that the rank r_ε depends only in the kernel function and the desired accuracy, and is box-independent.

2.2.3 Compact representation

The approximation (2.4) allows us to construct two compact representations for each box B, which gives rise to a fast multilevel algorithm.

First is the compact representation for the potential in $W^{B,\ell}$ generated by the points in B. Application of the approximation in (2.4) to $y = x_i \in B$ and summing them up with source densities f_i, we obtain

$$\left| \sum_{x_i\in B} G(x,x_i) f_i - \sum_{q=1}^{r_\varepsilon} G(x, e_q^{B,\ell}) \left(\sum_{p=1}^{r_\varepsilon} d_{qp}^{B,\ell} \left(\sum_{x_i\in B} G(c_p^{B,\ell}, x_i) f_i \right) \right) \right|$$
$$\leq \left(\sum_{x_i\in B} |f_i| \right) \varepsilon.$$

This implies that by placing a set of sources

$$\left\{ f_q^{B,\ell} = \sum_p d_{qp}^{B,\ell} \left(\sum_{x_i \in B} G(c_p^{B,\ell}, x_i) f_i \right) \right\}_{1 \leq q \leq r_\varepsilon}$$

at points $\{e_q^{B,\ell}\}_{1 \leq q \leq r_\varepsilon}$, we can approximate at any $x \in W^{B,\ell}$ the potential generated by the sources in B. We call $\left\{f_q^{B,\ell}\right\}_{1 \leq q \leq r_\varepsilon}$ the *directional outgoing equivalent sources* of B in direction ℓ. These equivalent sources play the role of the source representations of B in the FMM algorithm. Note that the computation of $\{f_q^{B,\ell}\}_{1 \leq q \leq r_\varepsilon}$ uses only kernel evaluations and a matrix vector product with the $r_\varepsilon \times r_\varepsilon$ matrix $D^{B,\ell}$.

Similarly, the approximation (2.4) also provides us with a compact representation for the potential in B generated by the points in $W^{B,\ell}$. Since $G(x,y) = G(y,x)$, summing over $x_i \in W^{B,\ell}$ with weights f_i leads us to

$$\left| \sum_{x_i \in W^{B,\ell}} G(x', x_i) f_i - \sum_{p=1}^{r_\varepsilon} G(x', c_p^{B,\ell}) \left(\sum_{q=1}^{r_\varepsilon} d_{qp}^{B,\ell} \left(\sum_{x_i \in W^{B,\ell}} G(e_q^{B,\ell}, x_i) f_i \right) \right) \right|$$
$$\leq \left(\sum_{x_i \in W^{B,\ell}} |f_i| \right) \varepsilon.$$

This means that, from the potentials $\{u_q^{B,\ell} = \sum_{x_i \in W^{B,\ell}} G(e_q^{B,\ell}, x_i) f_i\}_{1 \leq q \leq r_\varepsilon}$ at points $\{e_q^{B,\ell}\}_{1 \leq q \leq r_\varepsilon}$, we can compute at any $x' \in B$ the potential generated by $x_i \in W^{B,\ell}$ through a matrix vector product with $D^{B,\ell}$ and kernel evaluations. These potentials $\{u_q^{B,\ell}\}_{1 \leq q \leq r_\varepsilon}$ are called the *directional incoming check potentials* of B in direction ℓ. In the DFMM setting, these check potentials play

the role of the potential representations of the FMM algorithm.

2.2.4 Translation operators

Another important component of the algorithm is the translation operators that transform the compact representations introduced above. As per definition, the wedges of the parent box and the child box are nested. Therefore, for each box B and one of its directions ℓ, we can find a direction ℓ' such that $W^{B,\ell} \subset W^{B_c,\ell'}$ for every child B_c of B.

1. The far-to-far translation then computes the outgoing directional equivalent sources of $\{f_q^{B,\ell}\}_{1\le q\le r_\varepsilon}$ of B in direction ℓ from the outgoing equivalent sources of B's children in direction ℓ'. More precisely, it takes $\{f_{q'}^{B_c,\ell'}\}_{c,q'}$ located at $\{e_{q'}^{B_c,\ell'}\}_{c,q'}$ as the true sources and performs the following computation:

$$f_q^{B,\ell} \Leftarrow \sum_p d_{qp}^{B,\ell} \left(\sum_c \sum_{q'} G(c_p^{B,\ell}, e_{q'}^{B_c,\ell'}) f_{q'}^{B_c,\ell'} \right). \tag{2.5}$$

2. On the other hand, the local-to-local translation computes the incoming directional check potentials of B's children at direction ℓ' from the the incoming directional check potentials $\{u_q^{B,\ell}\}_{1\le q\le r_\varepsilon}$ of B at directional ℓ:

$$u_{q'}^{B_c,\ell'} \Leftarrow u_{q'}^{B_c,\ell'} + \sum_p G(e_{q'}^{B_c,\ell'}, c_p^{B,\ell}) \left(\sum_q d_{qp}^{B,\ell} u_q^{B,\ell} \right). \tag{2.6}$$

3. Finally, the far-to-local translation is applied to all pairs of boxes A and B that are in each other's interaction list. Suppose that B is in the wedge $W^{A,\ell'}$ of A and that A is in the wedge $W^{B,\ell}$ of B. The far-to-local translation uses $\{f_{q'}^{A,\ell'}\}_{1\le q'\le r_\varepsilon}$ to update $\{u_q^{B,\ell}\}_{1\le q\le r_\varepsilon}$:

$$u_q^{B,\ell} \Leftarrow u_q^{B,\ell} + \sum_{q'} G(e_q^{B,\ell}, e_{q'}^{A,\ell'}) f_{q'}^{A,\ell'}. \tag{2.7}$$

2.2.5 Algorithm description

Taking these components into consideration, following steps summarizes the directional fast multipole algorithm.

Tree construction. Construct the quadtree adaptively. The whole domain is partitioned dyadically and recursively until all of the leaf boxes contain a small number of points.

DFMM evaluation. For each iteration of matrix-vector multiplication:

1. Travel up the quadtree and visit the boxes in the low frequency region. These boxes have width less than the wave length λ. For each box B in this region, compute its equivalent sources using the non-directional far-to-far translation.

2. Continue traveling up the tree and visit the boxes in the high frequency region, where the boxes have width of at least λ. For each box B in this

region and each direction ℓ, compute its directional equivalent sources $\{f_q^{B,\ell}\}_{1\le q\le r_\varepsilon}$ using the far-to-far translation (2.5).

3. Travel down the quadtree and visit the boxes in the high frequency region. For each box B and each direction ℓ, perform the following two steps:

   (a) For each A in B's interaction list and in $W^{B,\ell}$ update $\{u_q^{B,\ell}\}_{1\le q\le r_\varepsilon}$ using the directional far-to-local translation.

   (b) If B is not a leaf, perform the directional local-to-local translation (2.6) to transform $\{u_q^{B,\ell}\}$ to the incoming check potentials for B's children. If B is a leaf, perform for $x_i \in B$

$$u_i \Leftarrow u_i + \sum_p G(x_i, c_p^{B,\ell}) \left(\sum_q d_{qp}^{B,\ell} u_q^{B,\ell} \right).$$

4. Continue traveling down the tree and visit the boxes in the low frequency regime. For each box B, perform the non-directional far-to-local and non-directional local-to-local translations.

5. Nearby interaction. For each leaf box B, for each $x_i \in B$, perform

$$u_i \Leftarrow u_i + \sum_{x_j \in N^B} G(x_i, x_j) f_j.$$

Computational complexity. It can be shown that, for the quadrature points $\{x_i\}$ distributed uniformly on Γ, this algorithm takes at most $O(k \log k)$

FLOPs (see [41] for the proof). An iterative BIE solver powered by DFMM for matrix-vector multiplication would take $O(n^{\text{iter}} k \log k)$ FLOPs, where n^{iter} is the number of iterations required for the solver to converge.

Final remarks. This chapter has reviewed the fast multipole method (FMM) for non-oscillatory kernels and the directional fast multipole method (DFMM) for oscillatory kernels. We now turn to the the parallel DFMM in the next chapter.

Chapter 3

Parallel DFMM and BIE Solver

3.1 Introduction

Here, we present a new parallel DFMM and an accelerated BIE solver based on it, using MPI (Message Passing Interface). The directional fast multipole method described in Section 2.2 is efficient for solving a wave equation at full scale. However, due to both memory and CPU runtime constraints, it remains impractical to solve a very large scale scattering problem (where the scatterer has size of more than 200 wavelengths) on a single processor. Since many realistic scattering applications are in thousands of wavelength, a distributed algorithm is needed. Such a parallel solver is particularly useful for the simulation of wave propagation in high frequencies and complex geometries.

An example is radiowave propagation in a forest, which has been studied for communication and sensing for more than four decades [100, 76, 75] (see Figure 3.1). In this setting, the total size of the scatterer is very large compared to the wavelength. At such high frequency, geometrical optics (GO) is often used. However, due to the complex nature of the many-body scatterer,

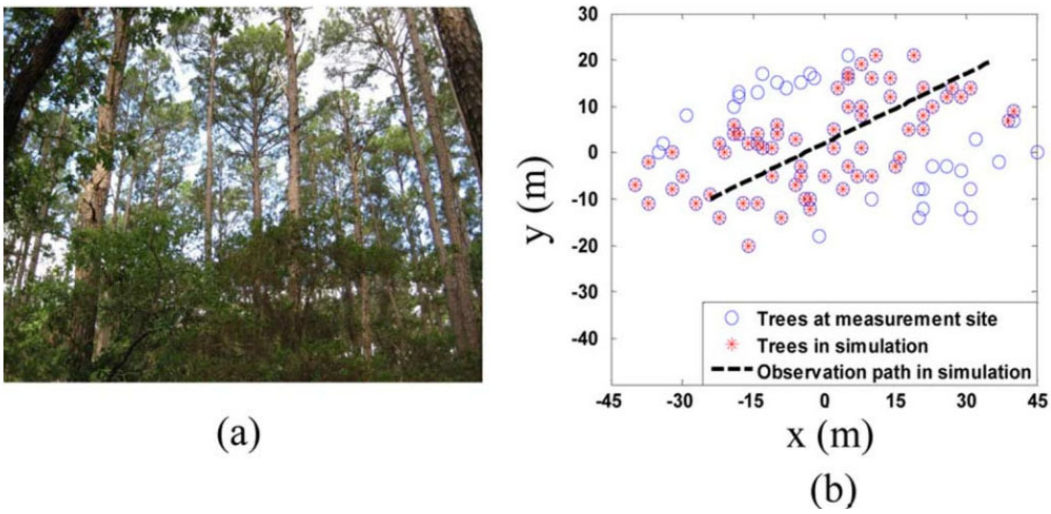

Figure 3.1: A potential application of the parallel directional fast multipole method and the related BIE solver: simulation of radio wave propagation in a forest. (a) Photograph of a loblolly pine forest in Bastrop, TX. (b) Tree distribution at site. Wave propagation through this forest was studied in [76].

the GO approach is no longer feasible. Furthermore, the individual objects in the channel may be too small for accurate approximation by GO. Hence, it is most efficient to use the boundary integral equation approach combined with a parallel version of DFMM to accelerate the computations.

Contributions. There have been many attempts to implement a parallel version of the classical FMM for multi-body interactions, on both distributed memory computing clusters [74, 112, 72, 54, 106] and shared memory processors [97, 91]. However, to our knowledge, no parallel FMM for highly oscillatory kernel or FMM-based parallel high frequency solver has been developed.

Our parallelization strategy is similar to previous work by Ying *et al.*

[112] for the parallelization of the kernel-independent FMM, but extended to the directional fast multipole algorithm. Another main contrast to the method in [112] is our scalable method for tree construction and data partitioning. Our design for tree construction is in fact inherent in the directional fast multipole method itself.

Since our objective in this project is to develop a scalable and parallel DFMM, we used the scalar Helmholtz equation as our test model for simplicity. With more effort, our method can be generalized to solve Maxwell or other vector-based wave equations. In fact, a sequential DFMM-based solver for electromagnetic scattering has recently been developed by Tsuji and Ying [104].

3.2 Parallel Algorithm

To achieve effective speedups over the sequential DFMM, we focused on the three goals in designing our parallel algorithm.

1. Load balancing: the workload should be distributed among processors in such a way that minimizes the time that any processor spends waiting for the other processors to reach the synchronization points.

2. The inter-processor communications and synchronizations are minimized.

3. The common data that processors share are minimized so that the program can scale efficiently without memory overflow.

Since the matrix-multiplication is the computationally dominant part of the whole solver, it is reasonable to address the parallel strategy for DFMM first.

3.2.1 DFMM

We first present some key observations about the properties of DFMM that guide our parallel strategy.

1. Most of the computations in the sequential algorithm is at the box levels. Therefore, in the parallel DFMM, we should partition the workload based on the boxes instead of the quadrature points.

2. As we pointed out in Section 2.2, the directional multipole algorithm does not require traversal through the boxes with width greater than $\sqrt{k}\lambda$, as the interaction lists of these boxes are empty. While this property makes DFMM theoretically slower, with $O(N \log N)$ complexity, than the FMM for non-oscillatory kernels, it also makes DFMM more suitable for parallelization. In fact, the main difficulty in the parallelization of hierarchical methods is the computational bottleneck at the top part of the tree [112, 96, 74]. In DFMM, we practically do not need to process the top part of the tree.

Data Partitioning. The input in the tree construction consists of both the source and target points, which we will treat as a single set of points. We first partition the domain occupied by the points into boxes of size $\sqrt{k}\lambda \times$

$\sqrt{k}\lambda \times \sqrt{k}\lambda$ (or smaller, if there are many processors to use). We call them *machine-level boxes* and denote the set by $\mathcal{B}$. These boxes are distributed to different processors, called $\{P_i\}_{1 \le i \le p}$, in such a way that the box distribution is spatially coherent with the point distribution.[1]

Each processor P_i checks which machine-level boxes it owns and and how many of its points reside in each box, and sends the result to P_0. On receiving the results from all other processors, P_0 then computes an load-balanced partition of $\mathcal{B} = \cup_{i=1}^{p} \mathcal{B}_i$, where $\mathcal{B}_i$ contains the boxes that will be owned by P_i. There has been extensive research on developing domain partitioning algorithms for optimal load balancing [79, 65, 66, 102]. In our solution, we use Lloyd's algorithm [79] due to its simplicity. Next, P_0 will send the partition information to all processors $\{P_i\}$. Finally, we redistribute the points corresponding the the partitioning of $\mathcal{B}_i$.

Tree construction and cache strategy. Once the redistribution is done, each processor constructs its local tree topology independently. After this step, each processor P_i needs to generate data associated with each tree node. In the directional fast multipole algorithm, the data for each box B includes but is not limited to the following:

- the index set of points in B if B is a leaf box;

[1] By spatially coherent, we mean that if a machine-level box is owned by P_i, the number of points inside that box that are owned by a different processor should be minimized.

- the *4 box lists* [51, 50], including the adjacency and interaction lists;
- directions $\{l^B\}$ associated with B;[2]
- the directional equivalent sources, including the source points $\{e_q^{B,l}\}_{1\leq q\leq r_\epsilon}$ and source charges $\{f_q^{B,l}\}_{1\leq q\leq r_\epsilon}$, in each direction l;
- the directional check points $\{c_q^{B,l}\}_{1\leq q\leq r_\epsilon}$ and check potentials $\{u_q^{B,l}\}_{1\leq q\leq r_\epsilon}$ in each direction l;

Therefore, after building its local tree topology, each processor P_i needs to do the following.

1. Obtain the topology of other trees that contain the boxes adjacent to those in $\mathcal{B}_i$ in order to compute various adjacency and directional interaction lists in both the high frequency and low frequency regions. The communication in this step is in fact quite small since the partition $\mathcal{B} = \cup_{i=1}^p \mathcal{B}_i$ is spatially coherent.
2. For each local box B and each associated direction l, compute the list $\{P_i^{B,l}\}$ of processors that will use $\{f_q^{B,l}\}_{1\leq q\leq r_\epsilon}$ in their far-to-local translations. We call P_i the *server* and $\{P_i^{B,l}\}$ the *clients* of the box-and-direction tuple $< B, l >$.
3. For each $< B, l >$, compute the directional equivalent source points $\{e_q^{B,l}\}_{1\leq q\leq r_\epsilon}$ and send these data to the clients of $< B, l >$.

[2] If B is in the low frequency regime, we can consider B having only one associated direction.

The rest of the parallel tree construction is the same as in the sequential algorithm.

DFMM evaluation. In the upward pass of the tree, no inter-processor communication is needed. For a box A, its directional equivalent sources $\{f_q^{A,\ell}\}_{1\leq q\leq r_\epsilon}$ for each direction l are only dependent on its children, which are owned by the same processor.

In the downward pass, communications are only needed during the far-to-local translations. In DFMM, the computation of the directional check potential $\left\{u_q^{B,l}\right\}$ of B depends on the directional equivalent sources $\left\{f_q^{A,l}\right\}$ of the boxes A in $W^{B,l}$. If the two boxes A and B belong to different processors, P_i and P_j respectively, P_i will transmit $\left\{f_q^{A,l}\right\}$ to P_j before the computation of $\left\{u_q^{B,l}\right\}$. Since the probability of a box in the interaction list of B belonging to a different processor is low and since$\left\{f_q^{A,l}\right\}$ is of size $O(1)$, the communication cost would be relatively small to the total computational cost.

Nevertheless, for minimal waiting time, the communication should take place in the upward pass of the tree, in parallel with the computations of equivalent sources. For each box A and direction l in the upward pass, once $\{f_q^{A,\ell}\}$ is computed, P_i immediately sends this information to the clients of $< A, l >$, using a non-blocking communication scheme.

For other performance optimizations that are not necessarily related to parallelization, see Appendix A.

3.2.2 BIE solver

The parallelization for the solver is fairly straight forward compared with that of the DFMM part. In our implementation, we used the collocation method with continuous piecewise linear basis functions for discretization (see Section 1.2.2). The standard Gaussian quadrature rule on triangles were used for approximating the integral $\int_\Gamma K(x_i, y)\mu(y)\, ds(y)$, where $x_i \in X$ is a collocation point. For singularity correction, we used the Duffy transformation in the neighborhood of x_i [35].

The source points (quadrature points) and the target points (collocation points) are treated as a single set of input points for the tree construction of the parallel DFMM. These points are partitioned in a spatially coherent way using Lloyd's algorithm [79] (see Figure 3.2). Since the parallel DFMM distributes the workload based on boxes instead of points, there are cases where a point $x \in X$ belongs to P_i in the BIE solver, but its copy in the parallel DFMM belongs to another processor P_j. In such cases, for every GMRES iteration:

1. Each processor P_i sends the sources $\{f_{X_j}\}$ to P_j, where $X_j \subset X$ are the points that belong to P_j in the parallel DFMM, before the interaction is computed.

2. P_j sends the computed potentials $\{u_{X_i}\}$ back to P_i, where $X_i \subset X$ are the points that belong to P_i in the solver.

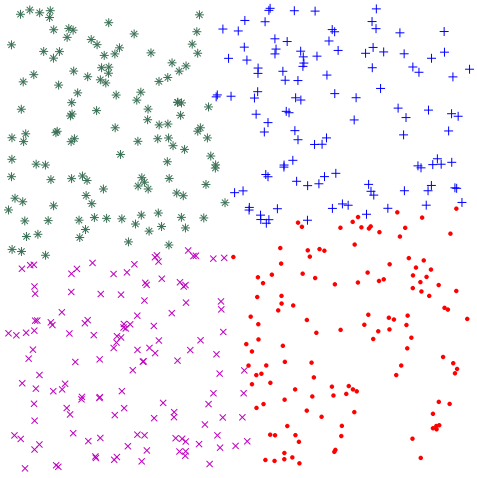

Figure 3.2: An illustration of partitioning a non-uniform set of points in the unit square using Lloyd's algorithm.

3.3 Numerical Tests

In this section, we describe different numerical tests that demonstrate the efficiency and scalability of our parallel algorithm. Here's the summary of our test configurations.

Model geometry. We used a forest-like geometry to test our parallel algorithm (see Figure 3.3). The geometry is a board of $M \times N$ perfect conducting cylinders, each of radius 1 and height H, which can be configured for different test purposes. The spacing between two nearest cylinders on the grid is 4. These parameters are measured as the number of wavelengths, *i.e.* the wavelength has unit length.

Accuracy Level. The error bound for all of the tests described below

is 10^{-3}.

Compute architecture. We ran our tests on the Lonestar cluster of the Texas Advanced Computing Center (TACC). TACC/Lonestar consists of 1888 compute nodes, with two Xeon Intel Hexa-Core 64-bit Westmere processors per node, for a total of 22,656 cores. The core frequency is 3.3 GHz and the theoretical peak compute performance of TACC/Lonestar is 302 TFLOPS. The cluster has 44 TB of total memory and 276 TB of local disk space. Nodes are interconnected with InfiniBand technology in a fat-tree topology with a 40 Gbit/sec point-to-point bandwidth. See [99] for more information.

Implementation and libraries. Our parallel algorithm was implemented in C++. We used:

- Intel's Math Kernel Library (MKL) [59] for fast exponential, square root, and reciprocal function evaluation,
- FFTW [45], an efficient Fast Fourier Transform library, to compute the far-to-local translations and for the interpolation of the solution during each GMRES iteration,
- and PETSc [6] for its distributed GMRES solver.

We used the combined layer kernel for solving the Helmholtz equation in three dimensions, *i.e.*

$$K(x,y) = \frac{\partial G(x,y)}{\partial n(x)} - i\gamma G(x,y),$$

where $G(x,y) = \frac{1}{4\pi}\frac{e^{ik|x-y|}}{|x-y|}$.

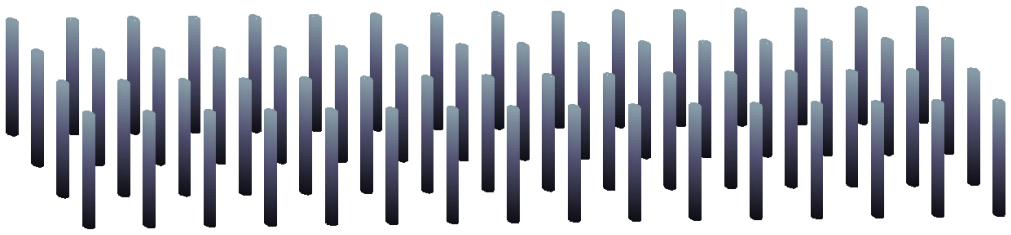

Figure 3.3: Model geometry for the numerical tests of the parallel DFMM and BIE solver. This board of multiple cylinders is a simplified model of a forest.

3.3.1 On the parallel DFMM

Here, we report the strong and weak scalability tests for the parallel DFMM. For strong scalability analysis, we keep the problem size fixed but increase the number of processors. For weak scalability analysis, the number of input points per processor is fixed but the problem size and hence the number of processors can vary. For both scalability tests, we report the wall-clock time for different phases of a single matrix-vector multiplication using the parallel DFMM. Our timing reflects the maximum time across processes as we synchronize each phase before the end time is measured. The communication time is measured using IPM (Integrated Performance Monitoring), an MPI profiling tool [46].

Strong scalability. Tables 3.1, 3.2, and Figure 3.4 show our strong scalability test results.

#P	CPU Time (seconds)						
	Matrix-vector computation						Setup
	Total	L-Up	H-Up	H-Down	L-Down	Comm	
1	1750.0	9.20	70.1	1575.4	95.3	0.0	35.9
4	475.9	2.80	19.7	425.3	27.2	0.9	9.2
16	124.3	0.72	6.1	110.1	6.9	0.5	2.7
64	34.1	0.21	2.3	29.5	1.7	0.3	1.5
256	9.9	0.05	1.2	8.0	0.4	0.2	2.4
1024	4.0	0.03	0.8	2.7	0.23	0.2	4.3

Table 3.1: **Strong scalability** results for a fixed-size problem: 64×16 cylinders, each of height $H = 4$. Each cylinder is discretized with 2252 points and the whole domain has approximately 2.3 million points. The largest side of the board is 256-wavelengths long and so the its covering box (root of the octree) has size $256 \times 256 \times 256$. Here #P is the number of processors; Total is the total time of a single interaction computation. L/H Up/Down are the times spent in the low/high frequency regimes while traveling up/down the tree. Comm is the time spent for MPI communications. We also report timing for the tree construction (in the Setup column).

#P	Seconds	Scaling factor
1	1750.0	n/a
4	475.9	3.68
16	124.3	3.83
64	34.1	3.65
256	9.9	3.44
1024	4.0	2.48

Table 3.2: **Strong scalability** results: the scaling factor of parallel DFMM (not counting the tree construction). We define scaling factor as the speedup in runtime for each quadrupling the number of processors. The test description is the same as in Table 3.1.

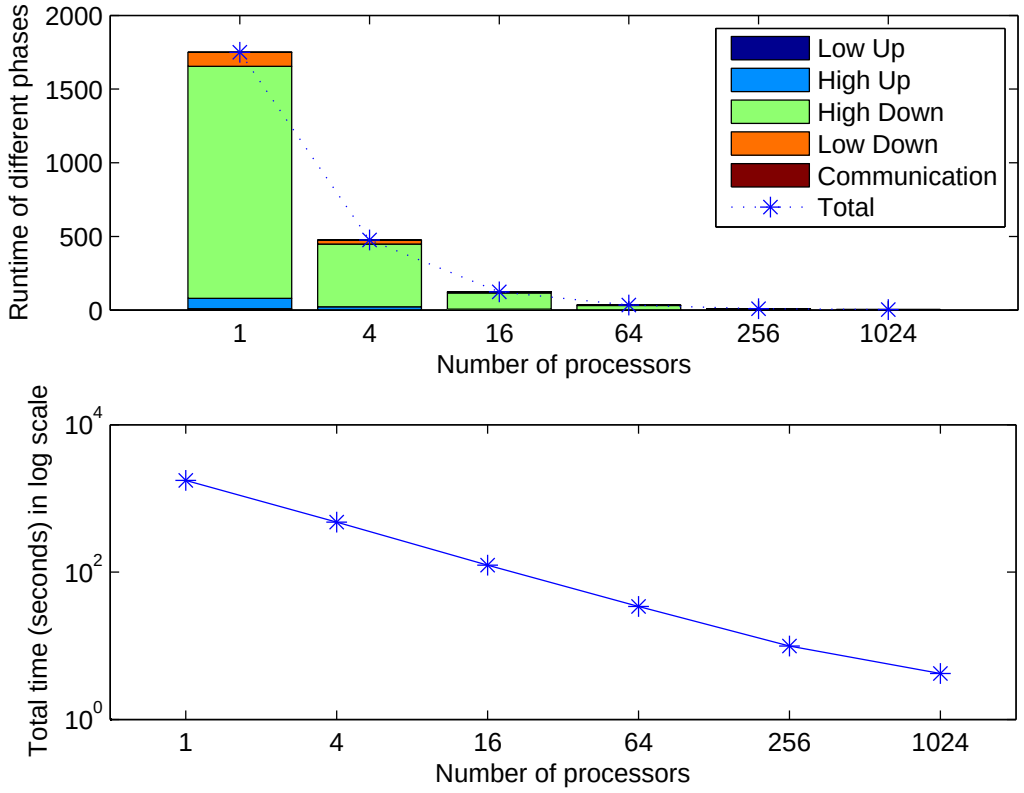

Figure 3.4: **Strong scalability** results. Top: the stacked bar chart of runtimes for different phases in the interaction computation. Bottom: the total time of the interaction computation displayed in log scale. The test description is the same as in Table 3.1.

For the problem configuration described in Table 3.1, we obtained excellent scalability for up to 256 processors. We also see that the communication time is substantially small compared to the computational phases, especially the downward tree traversal in the high frequency regime. Load balancing is the main bottleneck to achieve better scalability. Boxes in the middle of the domain have more partners in their interaction lists than the number of partners that boxes near the edges or corners of the domain have. Thus, the processors that serve the middle boxes have to do slightly more work.

There is a small dip in scalability when the number of processors increases from 256 to 1024. The reason is that for optimal performance, the machine-level boxes should have size $\sqrt{k}\lambda$ (or 8 for the tested geometry, as $k = 256$). If given plenty of computational resources, our parallel algorithm makes the machine-level boxes smaller than the ideal size so that they can be distributed fairly among the processors. The trade off is that each machine-level box has to interact and communicate with more partners, *i.e.* boxes that are in its interaction list, as we cannot travel up further the tree. A resolution to this problem is discussed in the future work part (see Section 3.4).

We also observed that the parallel tree construction has far worse scalability. In fact, there's even a dip in its performance when the number of processors gets larger than 64. There are two main reasons.

1. There are multiple iterations of redistribution of the machine-level boxes to obtain a good load balance among the processors. The complexity

of this greedy algorithm in fact increases with respect to the number of processors.

2. Part of the tree construction step is to compute and retrieve the adjacency and interaction lists for each box. The number of boxes in those lists increases with respect to the number of processors. This is more prevalent when the size of the machine-level boxes gets too small.

However, the non-optimal scalability of the parallel tree construction is our deliberate design choice. It plays a significant role in the good scalability of the interaction computations and hence of the solver. In the solver, the tree construction is done only once while the matrix-vector multiplication is called many times (for different vectors).

Weak scalability. Table 3.3 and Figure 3.5 show our weak scalability test results. We do not report the scaling factor for weak scalability results because the algorithmic complexity changes as we increase the problem size.

We can see from Figure 3.5 that in the weak scalability test, the runtime grows as $O(\log p)$ for p up to 1024, where p is the number of processors. This is expected since the parallel DFMM has theoretical complexity of $O(n \log n)/p$, which equals $O(\log p)$ when the number of points per cylinder is fixed. For higher p, the program's scalability suffers since the machine-level boxes become too small, as explained in the strong scalability test.

Board Dim	CPU Time (seconds)						
	Matrix-vector computation						Setup
	Total	L-Up	H-Up	H-Down	L-Down	Comm	
4×1	0.50	0.01	0.13	0.28	0.08	0.00	0.04
8×2	1.25	0.01	0.35	0.78	0.10	0.01	0.58
16×4	1.97	0.01	0.50	1.33	0.10	0.02	0.66
32×8	2.92	0.02	0.69	2.02	0.11	0.06	2.82
64×16	4.01	0.03	0.78	2.71	0.23	0.23	4.29
128×32	19.96	2.10	1.29	6.38	5.89	2.86	25.42

Table 3.3: **Weak scalability** results. The number of processors increases with respect to the size of the board so that each cylinder of height $H = 4$ is owned by a processor. There are 2252 points per cylinder. Total, L-Up, H-Up, H-Down, L-Down, Comm, and Setup have the same meanings as in Table 3.1. Board Dim is the board dimension, expressed in terms of cylinders per dimension.

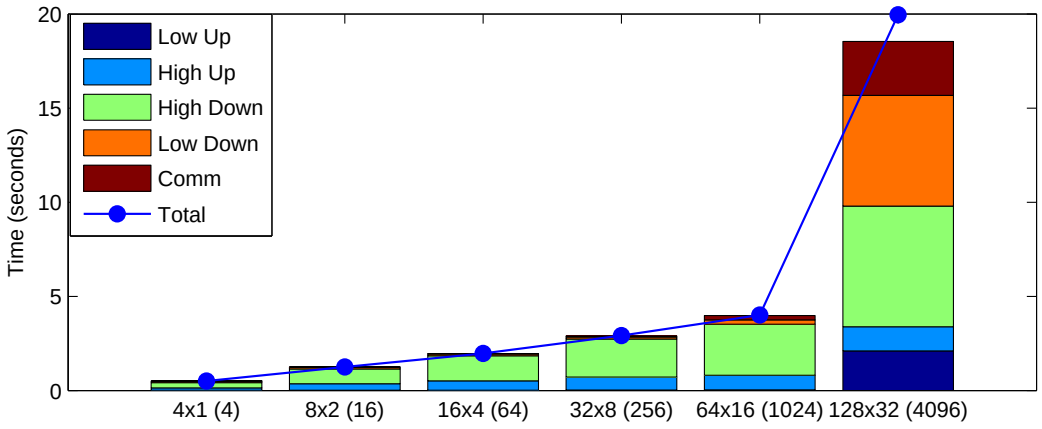

Figure 3.5: **Weak scalability** results. The test configuration is described in Table 3.3.

#P	Seconds	Scaling factor
4	15998.5	n/a
16	4181.9	3.83
64	1147.6	3.64
256	364.7	3.15
1024	209.2	1.73

Table 3.4: **Strong scalability** results for a fixed-size problem: 64×16 cylinders, each of height $H = 2$. Each cylinder is discretized with approximately 7250 points and the whole domain has slightly more than 2.3 million points. For this problem and the prescribed error bound of 10^{-3}, the solver converges in 52 GMRES iterations. Here, we report the CPU time and scaling factor (defined in Table 3.2) of the program with respect to different numbers of processors used.

3.3.2 On the BIE solver

Similarly to the the previous part, we provide the strong scalability and weak scalability tests for the BIE solver. In addition, we experimented with the number of iterations to see how it varies with respect to the board configuration. The analysis on the number of GMRES iterations is important and needed to explain the weak scalability results.

In testing the performance of the parallel BIE solver, we discretized the surface of cylinders using the collocation scheme with 8 points per wave length per dimension.[3] The Gaussian quadrature rule generated an additional 7 points per triangle.

[3]We discretized with a smaller number of points when testing the parallel DFMM since only the accuracy of the matrix-vector multiplication, not of the the solution, was being examined.

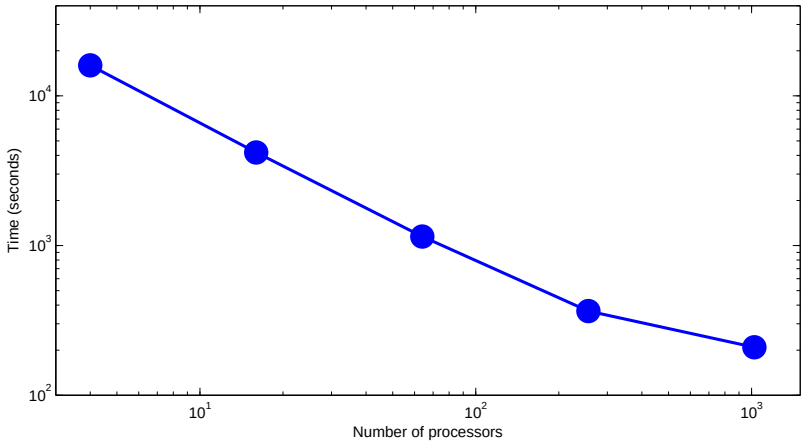

Figure 3.6: The log-scale plot of CPU time against the number of processors, for the test configuration described in Table 3.4.

Strong scalability. Table 3.4 and Figure 3.6 show the strong scalability test results for the parallel BIE solver. Since matrix-vector multiplication is the bottleneck of the solver, we see that the accelerated BIE solver has a similar scalability property to the parallel DFMM.

Number of GMRES iterations. The number of GMRES iterations required to achieve certain accuracy is an important quantity since the runtime of the BIE solver is closely proportional to this number. In fact, how to minimize this number, by way of preconditioners for instance, is a research of great interest.

We mentioned in Section (1.2.2) that with the usual choice of coupling parameter for the combined layer BIE, *i.e.* γ is proportional to k, the condition number of the system typically scales as $O(k^{1/3})$. However, this is not true

Board Dim	Height of cylinders 2	4	8	16
4×1	12	13	13	15
8×2	15	18	22	24
16×4	23	37	53	66
32×8	35	84	191	490
64×16	52	237	787	2800+

Table 3.5: Number of GMRES iterations for different board configurations. In these tables, $M \times N$ represents the number of cylinders per dimension, while H denotes the height of each cylinder.

for trapping environments [12]. Here, we present our numerical results by measuring the number of iterations needed for the BIE solver to converge, in the context of forest-like geometries, when the "forests" grow.

Table (3.5) shows that the number of iterations required for convergence grows tremendously with respect to the area of the board as well as the height of the cylinders. The reason is that for a large board and tall cylinders, the wave field is trapped by the obstacles. This phenomenon is similar to the cases where waves propagate through a porous medium.

Weak scalability. Finally, we show the weak scalability results for the accelerated BIE solver in Table 3.6 and Figure 3.7. The complexity of the parallel solver is $O(n^{\text{iter}} \log p)$, where n^{iter} denotes the required number of GMRES iterations for convergence and p denotes the number of processors used.

Board Dim	CPU Time (seconds) $H = 2$	$H = 4$	$H = 8$
4×1	7.0	12.2	26.08
8×2	16.2	30.8	75.04
16×4	35.0	91.2	270.98
32×8	81.6	297.8	1449.70
64×16	269.2	1360.4	8066.80

Table 3.6: Weak scalability results for the parallel BIE solver. The number of processors used for each test equals the number of cylinders on the board.

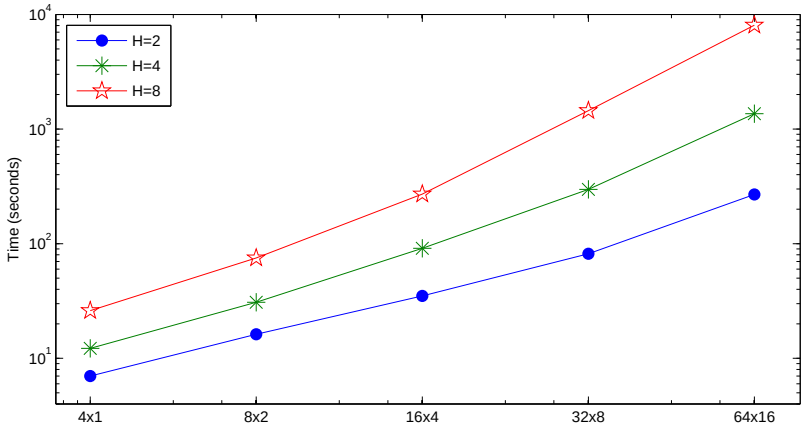

Figure 3.7: Weak scalability results: runtime of the parallel BIE solver against different board dimensions, for different heights H of cylinders. The number of processors used for each test equals the number of cylinders on the board.

3.4 Concluding Remarks and Future Improvements

We have presented a new parallel directional fast multipole method and an accelerated BIE solver based on this parallel DFMM. Despite the complexity of the directional multilevel algorithm, which has several different phases and combines many independent techniques, we were able to achieve good scalability for up to 1024 processors (with regards to the problem setting as described).

The new fast parallel solver also allowed us to examine the convergence property of iterative solvers when solving high frequency wave propagation in cluttered environments.

There are, however, also limitations to the current method that prevent us from solving very large scale problems. Here, we discuss these limitations and also propose improvements.

1. The current parallel DFMM does not scale well for very large number of processors. In an attempt to utilize all processors, the algorithm makes the machine-level boxes too small, *i.e.* smaller than $\sqrt{k}\lambda \times \sqrt{k}\lambda \times \sqrt{k}\lambda$. Since the local tree construction starts at the the machine level, it sacrifices the speedup of the multilevel algorithm at higher levels. One can remedy this problem using dynamic allocation of processors, described as follows.

   (a) The tree construction starts at the *mother boxes*, which always have size $\sqrt{k}\lambda \times \sqrt{k}\lambda \times \sqrt{k}\lambda$ and are not dependent on the number of

processors used.

(b) The mother boxes are processed by a subset of processors, called mother processors.

(c) At lower levels, the active processors allocate jobs to more processors since there are more boxes involved. This allocation process stops when all processors start their computations.

2. The number of GMRES iterations grows exponentially with respect to the number of obstacles, assuming that they are distributed arbitrarily. To avoid this problem, a good preconditioner is needed.

One can also further improve the performance by using a hybrid parallel architecture, such as the combination of distributed memory parallelization (via MPI) and shared-memory parallelization (*e.g.* via GPU acceleration [4]) in each process.

[4]GPU, which stands for Graphics Processing Unit, is a new parallelism paradigm that allows massively parallel computations on a single machine using shared memory.

Chapter 4

GO-BIE Method

4.1 Introduction

The high frequency asymptotic methods discussed earlier in this dissertation have motivated a number of varying attempts to combine such asymptotic results with the integral equation formulation [1, 18, 27, 34, 58, 48]. This combination is referred to as geometrical optics boundary integral equation (GO-BIE) method. We have also contributed to the development of this method by analyzing and developing a more robust localized integration technique, compared to the original one proposed in [18], used in the GO-BIE method. For simplicity, we will present this method for the wave scattering by a circle. The circular scatterer, while simple, contains difficult aspects of a more realistic setting. We also extended this method to more than one convex scatterer, which we propose in Section (4.6).

As we mentioned in Section 1.2.2, there are direct as well as indirect boundary integral equation formulations. While the two formulations appear mathematically similar, they differ significantly from a physical standpoint.

In the direct formulation (1.11), rewritten

$$\frac{1}{2}\mu(x) + \int_\Gamma \left(\frac{\partial G(x,y)}{\partial n(x)} - i\gamma G(x,y) \right) \mu(y)\, ds(y) = \frac{\partial u^I(x)}{\partial n(x)} - i\gamma u^I(x), \quad x \in \Gamma, \tag{4.1}$$

μ entails a physically realizable unknown (*e.g.* the total surface current in electromagnetics). No natural physical interpretation can be attached to the unknown function used in indirect formulations. Both formulations have two main characteristics that result in impractical computational times at high frequencies. Specifically, these correspond to:

1. the highly oscillatory behavior of the unknown density, which thus requires a large number of degrees of freedom to accurately represent it;

2. and the equally fast variations of the known kernel, which demands a large number of integration points even if the density varies slowly.

The new GO-BIE method addresses those two difficulties by:

- exploiting the pre-determined physical characteristics of the density μ in (4.1) when the scatterer is convex

- and using a new localized integration technique that requires fewer integration points by exploiting the asymptotic property of the kernel.

4.2 Asymptotic Representation

The function μ can be decomposed as

$$\mu(x) = k\mu^s(x)e^{ik\alpha\cdot x}. \tag{4.2}$$

For a convex and smooth scatterer, μ^s is a slowly oscillatory function of $x \in \Gamma$ (see, for example, Figure 4.1). This result follows the well-known research by Melrose and Taylor [82], and earlier by Keller [67]. Note that this is not true for a non-convex scatterer, when the plane wave incidence is such that its ray theory approximation gives rise to multiple reflections, as depicted in Figure 4.2. It is also not true if the scatterer has corners since the corners generate strong diffracted rays which illuminate the shadow side of the obstacle much more strongly than the rays that creep into the shadow zone of a smooth convex obstacle [27]. [1]

Now, introducing the representation (4.2) in (4.1) and dividing by $ke^{ik\alpha\cdot x}$, we obtain:

$$\frac{1}{2}\mu^s(x)+\int_\Gamma \left(\frac{\partial G(x,y)}{\partial n(y)} - i\gamma G(x,y)\right) e^{ik\alpha\cdot(y-x)}\mu^s(y)\,ds(y) = ix\cdot\alpha - i\frac{\gamma}{k}, \quad x \in \Gamma. \tag{4.3}$$

[1] For smooth but non-convex objects, $\mu(x)$ can be decomposed into a finite sum of k-independent terms, plus a controllable error tolerance. Scattering by convex polygons has also been explored in the research by by Chandler-Wilde and Langdon [27].

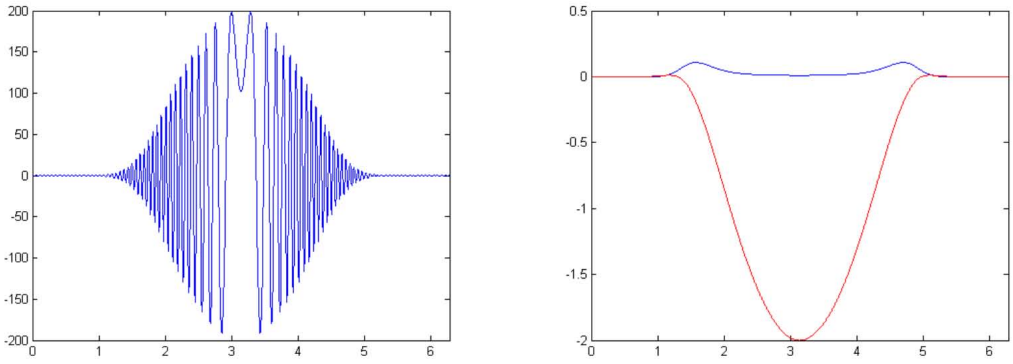

Figure 4.1: Left: real part of the solution $\mu(x)$ to (4.1) corresponding to scattering off a unit circle. Right: real and imaginary parts of $\mu^s(x)$. The incoming wave incident has frequency $k = 100$.

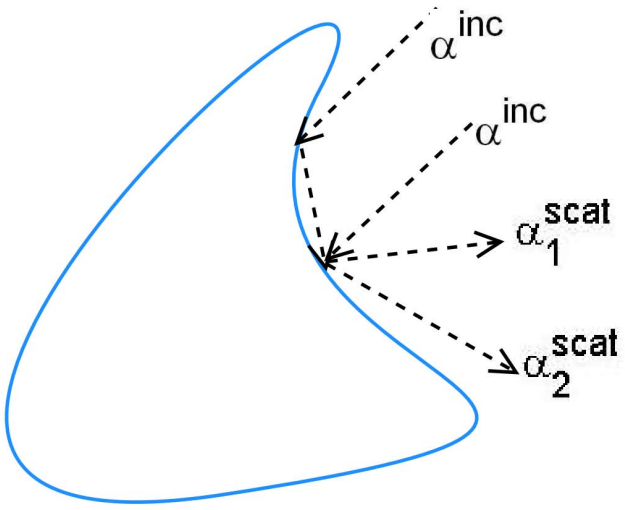

Figure 4.2: Multiple wave reflections by a non-convex scatterer.

This equation for μ^s can be abstracted as

$$\frac{1}{2}\mu^s(x) + \int_\Gamma K(x,y)\mu^s(y)\,ds(y) = f(x), \quad x \in \Gamma, \tag{4.4}$$

where

$$K(x,y) = \left(\frac{\partial G(x,y)}{\partial n(x)} - i\gamma G(x,y)\right) e^{ik\alpha\cdot(y-x)}$$

and

$$f(x) = ix\cdot\alpha - i\frac{\gamma}{k}.$$

Since the scatterer Ω is convex, there are exactly two shadow points on the boundary for a fixed incoming plane wave. To describe the qualitative behavior of the slow density $\mu^s(x)$, one decomposes the boundary into four disjoint zones (see Figure 4.3): the illuminated region facing the incoming wave, the shadow region, and two boundary regions. In both the illuminated and shadow regions, $\mu^s(x)$ varies at the scale of $O(1)$ [82]. Therefore, in order to approximate $\mu^s(x)$ in the illuminated and shadow regions, one only needs to sample $\mu^s(x)$ in these two regions with a constant number of Nyström points. In each boundary region, $\mu^s(x)$ varies at the scale $O(k^{-1/3})$ [82]. However, since each boundary region is of size $O(k^{-1/3})$, we only need a constant number of Nyström points to approximate the density $\mu^s(x)$ in these two boundary regions as well.

Therefore, the total number of Nyström points required to represent

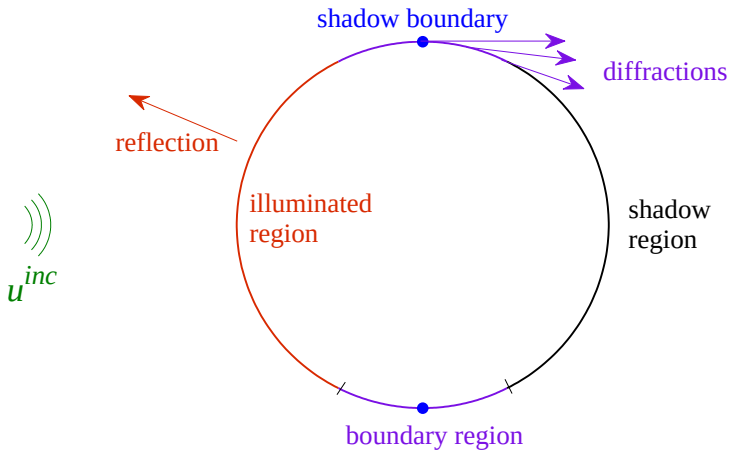

Figure 4.3: Reflections and diffractions in the scattering of an incoming wave u^I by a circular scatterer.

$\mu^s(x)$ is independent of k for a prescribed accuracy. This can be done by introducing a smooth boundary parameterization with derivative of order $O(k^{-1/3})$ at the shadow points and of order $O(1)$ away from the shadow regions (see Figure 4.4). The boundary Γ is then sampled uniformly according to this new parameterization with a constant number of Nyström points $\{x_i\}$.

4.3 Localized Integration

For $y \neq x$, the kernel K in the equation (4.4) behaves asymptotically as $e^{ik\phi(x,y)}$, where

$$\phi(x,y) = |x-y| + \alpha \cdot (y-x), \tag{4.5}$$

i.e. $\kappa(x,y)$, defined as $K(x,y)e^{-ik\phi(y)}$, is essentially a non-oscillatory function of y. This asymptotic property of K suggests a mechanism based on

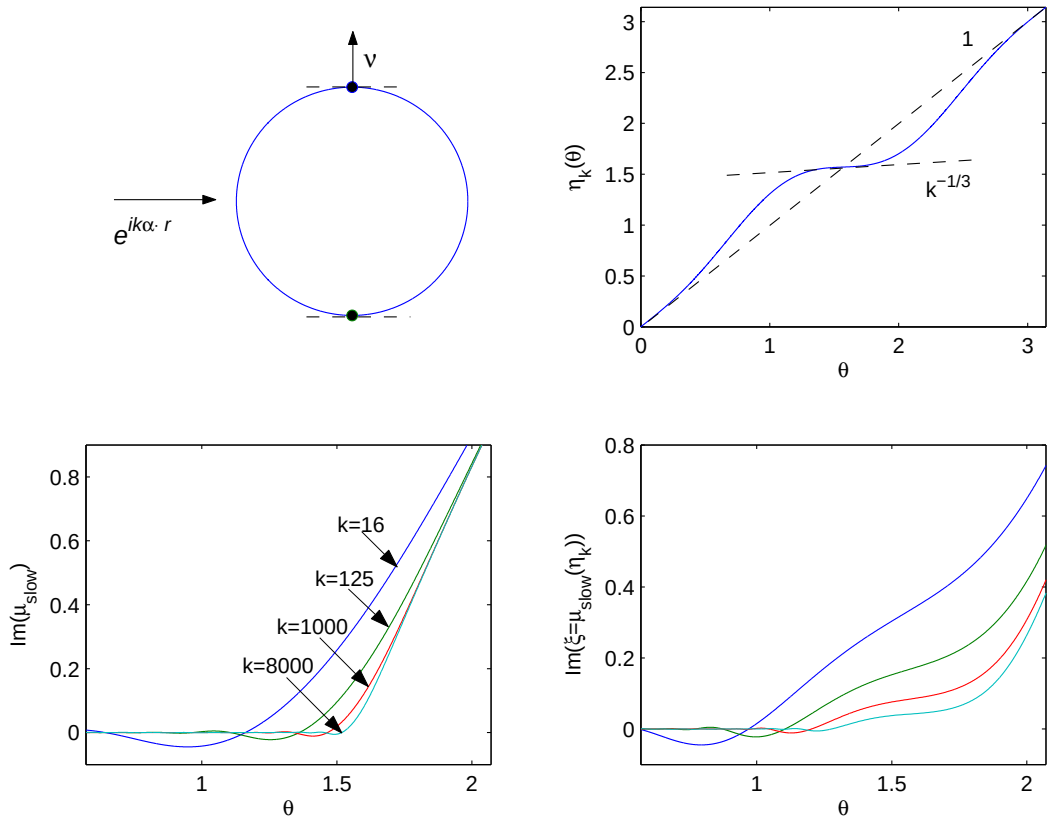

Figure 4.4: Top left: the shadow boundaries in the scattering by a circle. Top right: The boundary parameterization function has derivative of $O(k^{-1/3})$ at the shadow points. Bottom left: with uniform parameterization, the slow density μ^{s} has a sharp turn at $\frac{\pi}{2}$ when k is large. Bottom right: with the boundary parameterization in the above picture, μ^{S} has a bounded derivative with respect to k. For the circle geometry, we used $\xi(\theta) = \theta - \frac{k^{1/3}-1}{4} sin(2\theta) + \frac{k^{1/3}-1}{8} sin(4\theta)$ as the change of variable function, where θ corresponds to the parameterization angle of the boundary.

the classical "stationary phase method" (see [98, 15]) to rapidly evaluate the integral

$$\int_\Gamma e^{ik\phi(y)}\kappa(x,y)\mu(y)\,ds(y). \tag{4.6}$$

The stationary phase argument shows that the main contribution of this integral, away from from $y = x$, comes from the neighborhoods of the stationary points, *i.e.* the points y that satisfy

$$\frac{d}{ds(y)}\left(|x-y| + \alpha\cdot(y-x)\right) = 0.$$

Here, we will present the theory which makes rigorous this argument.

Definition 4.1. Define $w_A(t) = S(t, cA, A)\,(1 - S(t, -A, -cA))$, where

$$S(t,t_0,t_1) = \begin{cases} 1 & \text{if } t \le t_0, \\ \exp\left(\frac{2e^{-1/u}}{u-1}\right) & \text{if } t_0 < t < t_1,\ u = \frac{|t_1-t_0|}{t_1-t_0} \\ 0 & \text{if } t \ge t_1 \end{cases}$$

and c is a chosen constant in $(0,1)$. The function w_A is called a cutoff function (illustrated in Figure 4.5) and has two important properties

- w_A is infinitely differentiable and compactly supported on $[-A, A]$,
- $w_a(t) = w_A(\frac{A}{a}t)$.

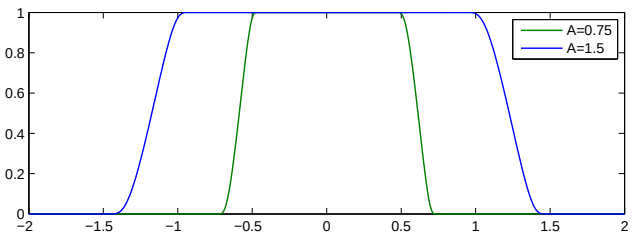

Figure 4.5: Cutoff function w_A for different values of A.

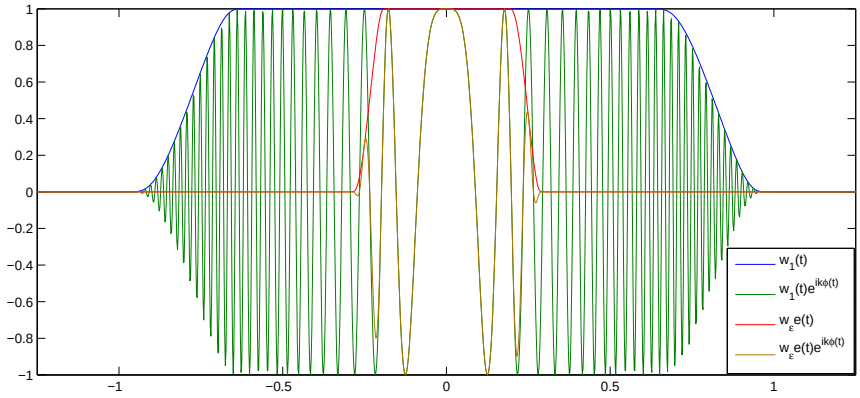

Figure 4.6: Localized integration: the integral of $e^{ik\phi(t)}w_1(t)$ (dense) on $[-1,1]$ can be approximated by that of $e^{ik\phi(t)}w_\epsilon(t)$ on $[-\epsilon,\epsilon]$.

Theorem 4.2. *Suppose $\phi(t)$ is a real analytic function which has only one stationary point of order p at 0, i.e. $\phi^{(j)}(0) = 0$ (for all $j = 1, 2, .., p$) but $\phi^{(p+1)}(0) \neq 0$. Let the cutoff functions w_1 and w_ϵ be defined as in Definition 4.1. Then:*

$$\int_{-1}^{1} e^{ik\phi(t)} w_1(t)\, dt = \int_{-\epsilon}^{\epsilon} e^{ik\phi(t)} w_\epsilon(t)\, dt + O\left(\left|k\epsilon^{p+1}\right|^{-n} \epsilon\right) \quad \forall n \geq 1.$$

Proof. Define $g(t) = w_1(t) - w_\epsilon(t)$. We consider the difference between the two integrals, restricted on the interval $[0, 1]$:

$$\begin{aligned} E_{k,\epsilon} &= \int_0^1 w_1(t) e^{ik\phi(t)}\, dt - \int_0^\epsilon w_\epsilon(t) e^{ik\phi(t)}\, dt \\ &= \int_{c\epsilon}^1 g(t) e^{ik\phi(t)}\, dt. \end{aligned}$$

(Due to symmetry, the analysis on the interval $[-1, 0]$ can be treated the same.)

Let $x = \phi(t)$. This change of variable is well-defined since ϕ is monotone in $(0, 1)$. Then $dt = z(t)dx$, where $z(t) = 1/\phi'(t)$. The error $E_{k,\epsilon}$ can be re-written as

$$E_{k,\epsilon} = \int_{\phi(c\epsilon)}^{\phi(1)} g(t) z(t) e^{ikx}\, dx.$$

Integrating by parts $n - 1$ times and using the fact that $g(t)$ and all of its derivatives vanish for $t = c\epsilon$ and $t = 1$, we get:

$$E_{k,\epsilon} = \frac{1}{(ik)^{n-1}} \int_{\phi(c\epsilon)}^{\phi(1)} \frac{d^{n-1}\left\{g(t)z(t)\right\}}{dx^{n-1}} e^{ikx}\, dx. \tag{4.7}$$

Notice that:

$$\frac{d^{n-1}\left\{g(t)z(t)\right\}}{dx^{n-1}} = \frac{d\frac{d^{n-2}\{g(t)z(t)\}}{dx^{n-2}}}{dx} = \frac{d\frac{d^{n-2}\{g(t)z(t)\}}{dx^{n-2}}}{dt} z(t).$$

Thus, by induction, we obtain:

$$\frac{d^{n-1}\left\{g(t)z(t)\right\}}{dx^{n-1}} = \sum c_j . g^{(j_0)} z^{(j_1)} z^{(j_2)} \ldots z^{(j_n)}(t), \tag{4.8}$$

where $\sum_{l=0}^{n} j_l = n - 1$.

On the one hand:

$$\begin{aligned} \left|g^{(j)}(t)\right| &= \left|\{w_1(t) - w_\epsilon(t)\}^{(j)}\right| \\ &= \left|w_1^{(j)}(t) - \frac{1}{\epsilon^j} w_1^{(j)}\left(\frac{t}{\epsilon}\right)\right| \\ &\leq \left|w_1^{(j)}(t)\right| + \frac{1}{\epsilon^j}\left|w_1^{(j)}\left(\frac{t}{\epsilon}\right)\right| \\ &\leq O(\epsilon^{-j}) \end{aligned} \tag{4.9}$$

since w is smooth and compactly supported.

On the other hand:

$$\begin{aligned} z(t) = \frac{1}{\phi'(t)} &= \frac{1}{\sum_{j=0}^{p-1} \frac{\phi^{(j+1)}(0)}{j!} t^j + \sum_{j \geq p} \frac{\phi^{(j+1)}(0)}{j!} t^j} \\ &= \frac{t^{-p}}{\sum_{j=0}^{\infty} c_j t^j}. \end{aligned} \tag{4.10}$$

If $z_0(t)$ is defined as $z_0(t) = \frac{1}{\sum_{j=0}^{\infty} c_j t^j}$, then $z(t) = z_0(t) t^{-p}$. Based on the assumptions of the theorem, the function $z_0(t)$ is infinitely differentiable on

$[0,1]$. The derivative of $z(t)$ can be decomposed as

$$\begin{aligned} z'(t) &= z_0'(t)t^{-p} + z_0(t)t^{-(p+1)} \\ &= \left\{z_0'(t)t + z_0(t)\right\} t^{-(p+1)} \\ &= z_1(t)t^{-(p+1)}, \end{aligned}$$

where $z_1(t) = z_0'(t)t + z_0(t)$ is also infinitely differentiable on $[0,1]$. Using induction, we can show that $z^{(j)}(t)$ has the decomposition representation $z_j(t)t^{-(p+j)}$, where $z_j(t) = z_{j-1}'(t)t + z_{j-1}(t)$ is infinitely differentiable on $[0,1]$. Then:

$$\left|z^{(j)}(t)\right| \leq \max_{0\leq t\leq 1} |z_j(t)| t^{-(p+j)} \leq O\left(\epsilon^{-(p+j)}\right), \quad t \in [c\epsilon, 1]. \tag{4.11}$$

The expansions (4.8), (4.9), and (4.11) imply

$$\left|\frac{d^{n-1}\left\{g(t)z(t)\right\}}{dx^{n-1}}\right| \leq O\left(\epsilon^{-(j_0+j_1+\ldots+j_n)-pn}\right) = O\left(\epsilon^{-pn-n+1}\right). \tag{4.12}$$

Therefore, from (4.7) and (4.12):

$$|E_{k,\epsilon}| \leq O\left((k\epsilon^{p+1})^{-n}\epsilon k \left|\int_{\phi(c\epsilon)}^{\phi(1)} e^{ikx}\, dx\right|\right).$$

Further observe that:

$$\left|\int_{\phi(c\epsilon)}^{\phi(1)} e^{ikx}\, dx\right| = \left|\int_{\phi(c\epsilon)}^{\tau} e^{ikx} dx + \int_{\tau}^{\phi(1)} e^{ikx}\, dx\right| = \left|\int_{\tau}^{\phi(1)} e^{ikx}\, dx\right| \leq \frac{2\pi}{k},$$

where τ satisfies $\tau - \phi(c\epsilon) = \frac{2\pi m}{k}$ and $\phi(1) - \tau < \frac{2\pi}{k}$. Hence

$$|E_{k,\epsilon}| = O\left((k\epsilon^{p+1})^{-n}\epsilon\right).$$

□

Remark 4.3. If the phase function ϕ has a stationary point at $s \neq 0$, the localized integration theorem is still applicable with the integration region and the cutoff functions shifted accordingly, *i.e.*

$$\int_{s-1}^{s+1} e^{ik\phi(t)} w_1(t-s)\,dt = \int_{s-\epsilon}^{s+\epsilon} e^{ik\phi(t)} w_\epsilon(t-s)\,dt + O\left(\left|k\epsilon^{p+1}\right|^{-n}\epsilon\right) \quad \forall n \geq 1.$$

Remark 4.4. Following [57], $\left|\int_{-1}^{1} w_1(t)h(t)e^{ik\phi(t)}\,dt\right|$ is asymptotically proportional to $k^{-\frac{1}{p+1}}$. Thus, if we fix $\rho = k\epsilon^{p+1}$ (*i.e.* ϵ proportional to $k^{-\frac{1}{p+1}}$), the localized integral $\int_{-\epsilon}^{\epsilon} f_\epsilon(t)h(t)e^{ik\phi(t)}dt$ is a good approximation of $\int_{-A}^{A} f_A(t)h(t)e^{ik\phi(t)}\,dt$ with a relative error of $O(\rho^{-n})$ for all $n \geq 1$.

Near 0, the phase $\phi(t)$ behaves like t^{p+1}. Thus, in its neighborhood of size ϵ, the integrand oscillates $O\left(\epsilon k^{\frac{1}{p+1}}\right)$ times. In the numerical evaluation of the integral, each oscillation requires a fixed number of quadrature points. Therefore, the number of points needed to obtain a prescribed accuracy for the localized integral is $O(\rho)$.

Figure 4.7 demonstrates the remark above. Using the Trapezoidal rule,

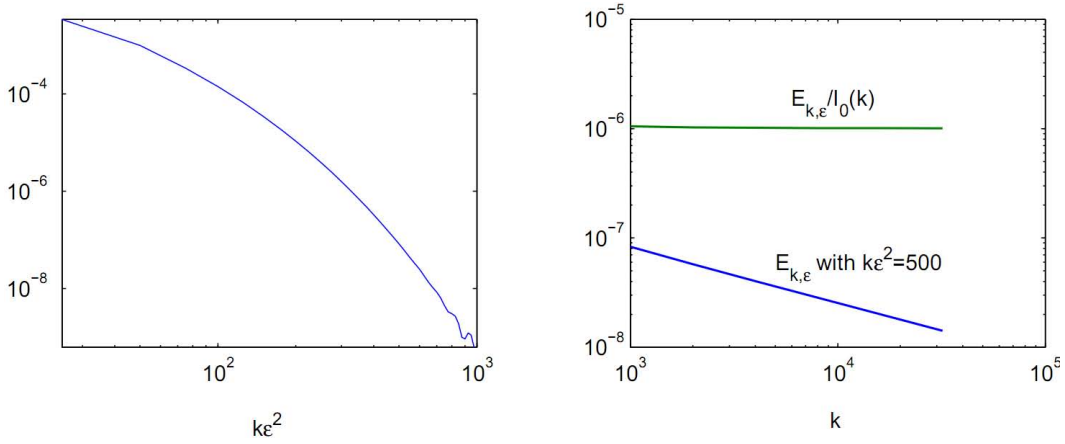

Figure 4.7: Convergence test of the localized integral approximation on $I(k)$ defined in (4.13). Left: with fixed k, the localized approximation of the integral has an error that decreases faster than any polynomial, with respect to $k\epsilon^2$, when ϵ increases. Right: with fixed $k\epsilon^2$, *i.e.* ϵ chosen to be proportional to $k^{-1/2}$, the error is approximately proportional to ϵ but the relative error stays constant.

we computed

$$I(k) = \int_{-1}^{1} w_1(t) e^{ik\cos(t)} \ln(t+\pi)\, dt \tag{4.13}$$

with $5k$ integration points and did the same for the localized integral with $5k\epsilon^2$ points. In this example, the stationary point of the phase $\phi(x) = \cos(x)$ is 0 with order $p = 1$.

This approximation result provides a numerical technique to compute the integral

$$\int_{\Gamma} e^{ik\phi(x,y)} \kappa(x,y) \mu^s(y)\, ds(y)$$

in fixed time for each x. Using partition of unity, this integral can be split and localized to the neighborhoods of critical points so that only a fixed number of

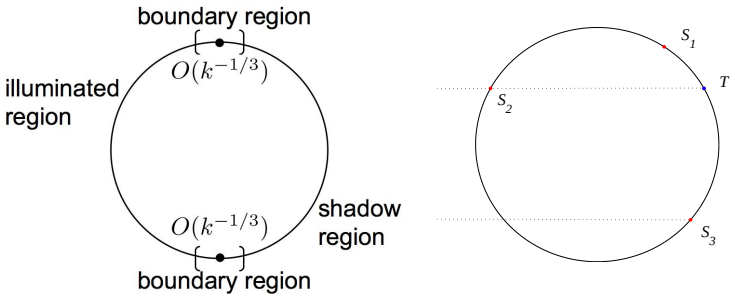

Figure 4.8: (a) The smooth convex scatterer is decomposed into four parts: illuminated region, shadow region, and two boundary regions. (b) For a target point T, the localized integrator is applied at the three stationary points S_1, S_2, and S_3.

integration points is required to obtain a prescribed accuracy for each critical region. The description of the critical points, for each Nyström point x, are described as follows (also illustrated in Figure 4.8 and Figure 4.9).

1. The target point (also called Nyström point) $y^t = x$, where the kernel $K(x, y^t)$ is hypersingular. This target point is covered by a region U_t of radius proportional to the wavelength λ ($p = 0$). Integrating around the target point requires special treatment since the kernel is hypersingular at this point.

2. The stationary points $\{y_j^{st}\}$ of the phase $\phi(x, y) = |x - y| + \alpha \cdot (y - x)$, at which ϕ has a vanishing derivative. Each stationary point is covered by a region U_s of size $O(k^{-1/2})$.

3. The shadow points $\{y_j^{sh}\}$, at which $n \cdot \alpha = 0$. These points are included in the set of critical points because in the region of size $O(k^{-1/3})$ around

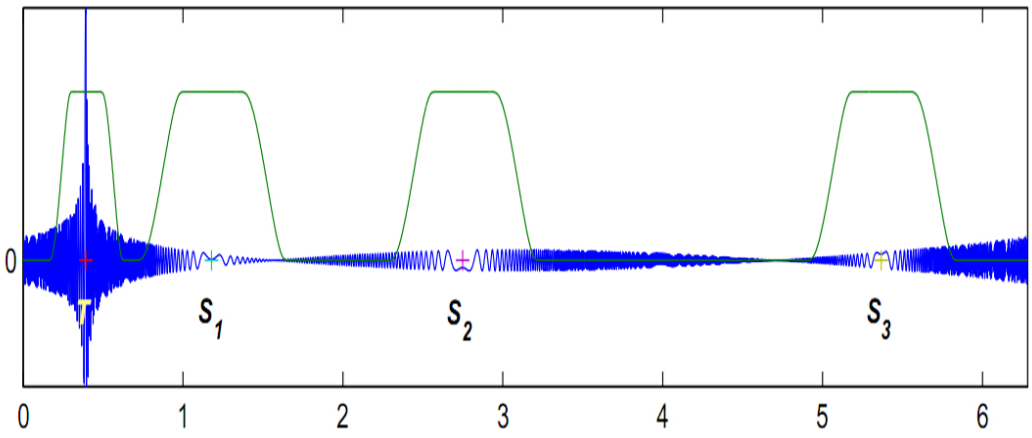

Figure 4.9: Using partition of unity, the integral of oscillatory function over the whole boundary can be split and localized to the neighborhoods of critical points.

the shadow points, $\mu^s(x)$ varies on the scale $O(k^{-1/3})$ [82].

Generalization of the localization theorem. For certain target points, the approximation described above is not sufficiently accurate due to the following.

1. In the expansion of $\phi'(t)$ as $t^p\left(c_0+\sum_{j=1}^{\infty}c_jt^j\right)$ (see the proof of Theorem 4.2), $|c_0|$ can be sufficiently small that $\phi'(t)$ is better approximated by a polynomial of higher-order than t^p. Hence some stationary points may require wider cutoff windows than the others.

2. The phase $\phi(t)$ behaves as a higher-order polynomial on one size of the stationary point, and as a lower-order polynomial on the other side. This is the case when the target point is close to one of the shadow points.

The kernel $e^{ik\phi(t)}$ oscillates at the rate $O(k)$ on one side of the target point, while oscillates at the rate $O\left(k^{1/3}\right)$ on the other side (see Figure 4.10).

To address these difficulties, we introduce an extension to Theorem 4.2.

Theorem 4.5. *Suppose $\phi(t)$ is a real analytic function which has only one stationary point of order p at 0, i.e. $\phi^{(j)}(0) = 0$ (for all $j = 1, 2, .., p$) but $\phi^{(p+1)}(0) \neq 0$. Let the cutoff functions w_1 and w_ϵ be defined as in Definition 4.1. Then:*

$$\int_{-1}^{1} w_1(t)e^{ik\phi(t)}\,dt = \int_{-\epsilon}^{\epsilon} w_\epsilon(t)e^{ik\phi(t)}\,dt + O\left(\left|k\epsilon\phi'(\epsilon)\right|^{-n}\epsilon\right) \quad \forall n \geq 1. \quad (4.14)$$

Proof. As in the proof of Theorem 4.2, we want to approximate $z^{(j)}(t)$, where $z(t) = [\phi(t)]^{-1}$. For the sake of induction, assume that $z^{(j)}(t)$ admits of the decomposition $z_j(t)t^{-j}[\phi(t)]^{-1}$, where $z_j(t) \in C^\infty$ on $[0, 1]$, for up to some $j \geq 0$. Then:

$$\begin{aligned} z^{(j+1)}(t) &= z_j'(t)t^{-j}[\phi(t)]^{-1} - jz_j(t)t^{-j-1}[\phi(t)]^{-1} - z_j(t)t^{-j}\phi''(t)[\phi(t)]^{-2} \\ &= \left\{z_j'(t)t - jz_j(t) - \frac{t\phi''(t)}{\phi'(t)}z_j(t)\right\}t^{-j-1}[\phi(t)]^{-1} \end{aligned}$$

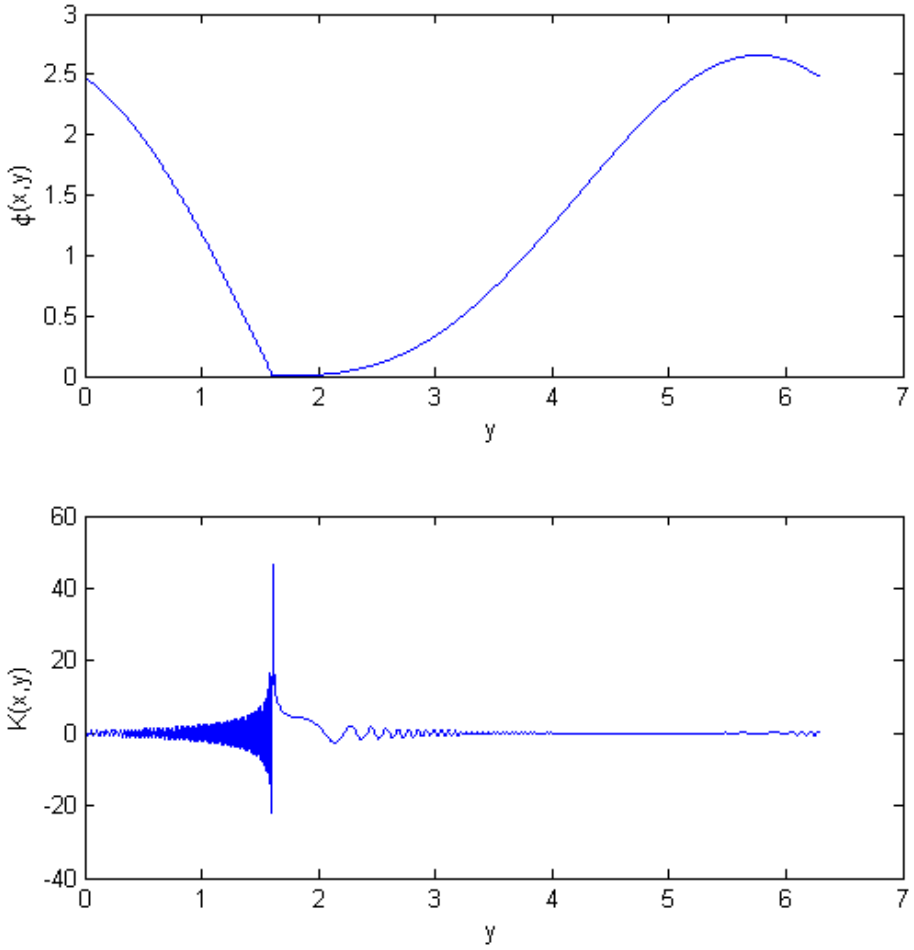

Figure 4.10: Top: The plot of the phase $\phi(x, y)$ as a function of y when x is near the shadow point $\pi/2$. The phase behaves like a linear function on the left side of x, while it behaves like a cubic function on the right side. Bottom: The plot of the kernel $K(x, y)$ under the same setting.

Taylor expanding $\frac{t\phi''(t)}{\phi'(t)}$, we obtain:

$$\frac{t\phi''(t)}{\phi'(t)} = \frac{\sum_{j=1}^{p-1} \frac{\phi^{(j+1)}(0)}{(j-1)!}t^j + \sum_{j\geq \max(p,1)} \frac{\phi^{(j+1)}(0)}{(j-1)!}t^j}{\sum_{j=0}^{p-1} \frac{\phi^{(j+1)}(0)}{j!}t^j + \sum_{j\geq p} \frac{\phi^{(j+1)}(0)}{j!}t^j} = \frac{\sum_{j\geq \max(p,1)} \frac{\phi^{(j+1)}(0)}{(j-1)!}t^j}{\sum_{j\geq p} \frac{\phi^{(j+1)}(0)}{j!}t^j}.$$

The sum $\sum_{j\geq p} \frac{\phi^{(j+1)}(0)}{j!}t^j$ is smooth and non-zero on $[0,1]$. Thus, $\frac{t\phi''(t)}{\phi'(t)}$ and the function $z_{j+1}(t)$, defined as

$$z_{j+1}(t) = \left\{ z_j'(t)t - jz_j(t) - \frac{t\phi''(t)}{\phi'(t)}z_j(t) \right\},$$

are infinitely differentiable on $[0,1]$. Therefore, by induction, the assumption above holds for all $j \geq 0$. Hence:

$$\left|z^{(j)}(t)\right| \leq \max_{c\epsilon\leq t\leq 1} |z_j(t)|t^{-j}[\phi'(t)]^{-1} \leq O\left(\epsilon^{-j}[\phi'(\epsilon)]^{-1}\right) \quad \forall t \in [c\epsilon, 1].$$

The rest of this proof is the same as for Theorem 4.2. □

The results of this theorem is illustrated in Figure 4.11. Note that unlike in Figure 4.9, the cutoff windows for the stationary points (or shadow points) are no longer symmetric and also do not have the same size. They are computed according to the behavior of the kernel near those critical points.

4.4 Algorithm Description

The algorithm to solve μ^s can be summarized as following.

Discretization. Sample points on the boundary Γ uniformly according to a boundary parameterization such that its derivative at the shadow points is of order $k^{-1/3}$ (see Figure 4.4). Let $\{\mu_i^s\}_{i=1}^N$ represent the slow density μ^s sampled at those points $\{x_i\}_{i=1}^N$ on the boundary. Now, we end up with a discrete system of N linear equations

$$\frac{1}{2}\mu^s(x_i) + \int_\Gamma K(x_i, y)\mu^s(y)\, ds(y) = f(x_i), \quad i = 1, 2, .., N, \tag{4.15}$$

where

$$K(x_i, y) = \left(\frac{\partial G(x_i, y)}{\partial n(x_i)} - i\gamma G(x_i, y)\right) e^{ik\alpha\cdot(y-x_i)}$$

and $f(x_i) = i\alpha \cdot x_i - i\frac{\gamma}{k}$.

Matrix solver. Run GMRES to solve the system of linear equations. At each iteration m, the following sub-steps need to be done for evaluating

$$\int_\Gamma K(x_i, y)\mu_{m-1}^s(y)\, ds(y).$$

1. Interpolate $\{\mu_{m-1}^s(x_i)\}_{i=1}^N$ to obtain $\mu_{m-1}^s(Y)$ on the boundary, where $Y \subset \Gamma$ is the set of integration points to be used later. Because the points in Y are not uniformly spaced, we can use cubic spline combined

with Fourier interpolation for an exponentially accurate interpolation.

2. Compute the stationary points for each target point x_i.

3. Generate a cutoff window for each critical point

   - The target point has covering of width $\epsilon = \frac{\rho}{k}$, where the parameter ρ is chosen according to the desired accuracy level.
   - For each stationary or shadow point y, the window $(y-\epsilon_l, y+\epsilon_r)$ is computed such that ϵ_l and ϵ_r solve $|k\epsilon_l\phi'(y-\epsilon_l)| = \rho$ and $|k\epsilon_r\phi'(y+\epsilon_r)| = \rho$ respectively. These algebraic equations can be solved via means of Newton's method. This is the result of Theorem (4.5).
   - Shrink and match the overlapping windows, if any (see Figure 4.11). It is essential that these overlapping partitions remain separated, rather than be merged, since the integrand varies at different scales over these regions.

4. Evaluate the integral for each window with a fixed number of integration points.

   - For each integration window covering a stationary point or a shadow point, standard Trapezoidal rule is used to leverage the property that the integrated function is smooth and periodic in that region.
   - Integrating around the target point requires a different method since the kernel converges to ∞. As shown in [31], the Hankel function

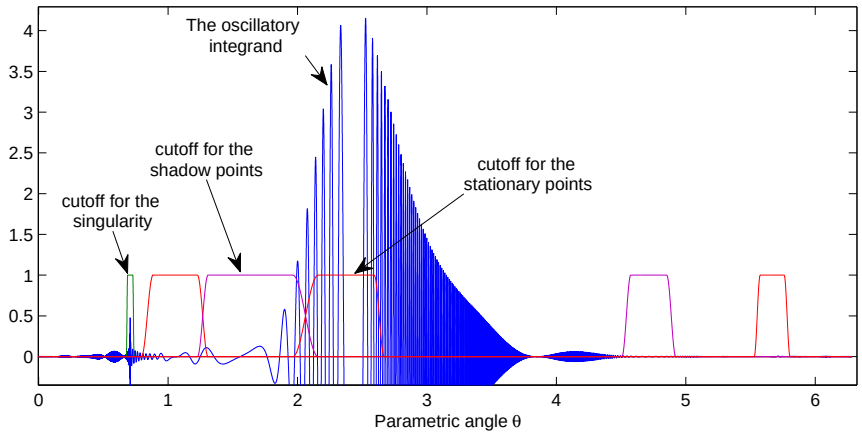

Figure 4.11: Illustration of the localized integration in implementation. The cutoff windows of different widths are generated using the results from Theorem 4.5. Overlapping windows are made thinner and aligned to avoid over evaluation of the integral.

behaves asymptotically as a log function near singularity:

$$H_0^{(1)}(k|x|) = \frac{2i}{\pi} \ln \frac{1}{|x|} + \frac{2i}{\pi} \ln \frac{1}{k} + C + O\left(|x|^2 \ln \frac{1}{|x|}\right) \qquad (4.16)$$

for $|x| \to 0$, where C is a constant. Hence, the integrated function in the target region has the form of $p(x) \ln \frac{1}{|x|} + q(x)$, where p and q are smooth functions. Following the discussion in [64], we devised a special Trapezoidal rule with discretization refinement in the neighborhood of singularity and with special quadrature weights in this neighborhood, which captures the behavior of the integrand near singularity.

Optimizations. Note that most of the operations in step 2, including

- the computations of critical points and their covering windows,
- the integration points as well as the kernels evaluated at those points,

are independent of the result μ_m^s obtained from each GMRES iteration. In fact, these computations are rather expensive compared to the rest of the GMRES matrix-vector multiplication. Hence, in our implementation, we used a precomputation scheme to cache all repeated computations in this operation.

4.5 Numerical Results

Here, we provide some numerical test results of the GO-BIE method.

We first examined the stability of the linear system when k grows. We did this by assembling the induced matrix elements and computed the matrix's condition number using MATLAB. We verified that the system is well-conditioned even for k very large, as shown in Table 4.1. The tested geometry is a unit circle.

Next, we present our numerical results on the the accuracy and performance of GO-BIE. For each wave frequency k, a reference solution, called $\mu^{s,exact}$, is computed using a direct simulation with $O(k^{1/3})$ points for the sampling of μ^s and 15 points per wavelength ($15\pi k$ total number of points) for each integral evaluation.[2] The reference solution is computed numerically us-

[2]The total computational complexity of this direct simulation is $O(k^{4/3})$.

	No. of sampling points			
k	20	40	60	80
50	11.2	12.3	12.8	13.4
100	10.9	12.4	12.9	14.2
200	11.6	12.6	13.5	14.4
400	12.1	12.7	13.8	15.2
800	12.4	13.1	13.8	15.5

Table 4.1: The condition number of the induced matrix A is stably small with respect to k.

ing a direct solver with highly dense discretization. The computed solutions are shown in Figure 4.12.

Table 4.2 demonstrates the bounded errors of μ^s corresponding to various wave frequencies, for scattering by a unit circle.

- For the prescribed error tolerance of $O(10^{-3})$:
  - 45 sampling points are used to represent μ^s.
  - The localization width ϵ is computed such that $|k\epsilon\phi'(x^c+\epsilon)| = \rho = 100$ for each critical point x^c.
  - The computation for each localized integral uses 100 discretization points.

  Using the implementation scheme described in the previous section, it takes approximately 30 seconds to compute the solution for every frequency k when running this method in MATLAB on a Pentium IV 2.8 GHz PC.

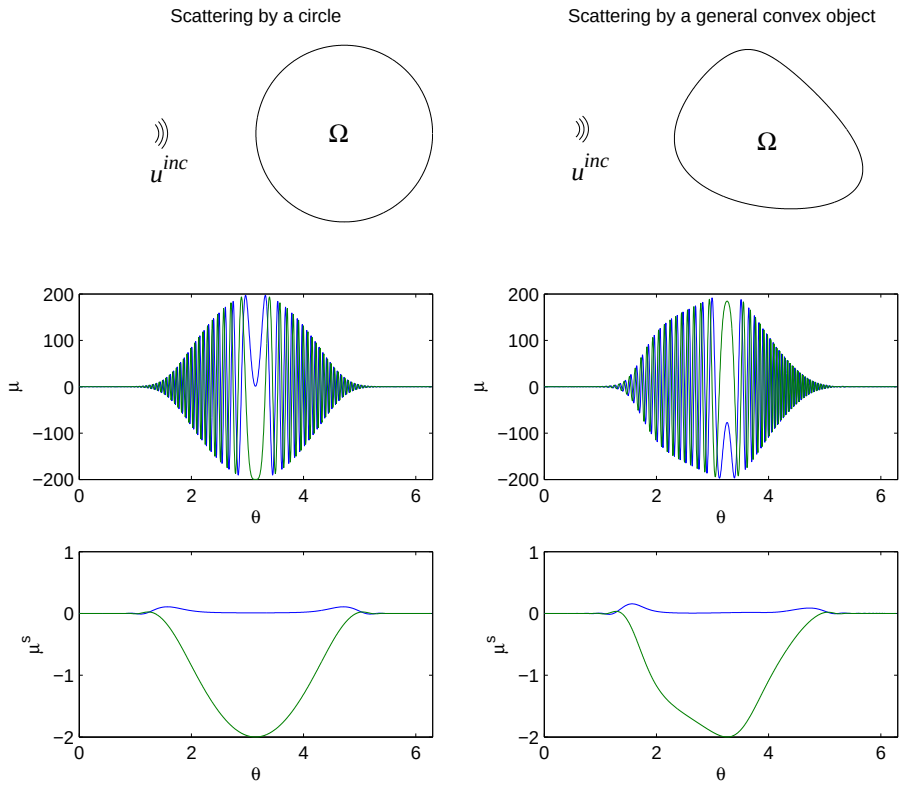

Figure 4.12: The solutions computed using the GO-BIE method for two different geometries. In both settings, the frequency $k = 100$. Left: scattering by a unit circle. Right: scattering by a more general convex object. Top row: illustration of the geometries. Middle row: the plot of the oscillatory density μ. Bottom row: the plot of the slow density μ^s.

	L_∞ errors of μ^s	
Frequency	45 unknowns	80 unknowns
1000	6.8e-4	7.1e-6
2000	4.4e-4	5.6e-6
4000	4.8e-4	4.5e-6
8000	6.4e-4	6.7e-6
16000	5.9e-4	5.7e-6
32000	3.2e-4	8.5e-6

Table 4.2: Accuracy of the GO-BIE method.

- For the prescribed error tolerance of $O(10^{-5})$:
  - 80 sampling points are used to represent μ^s
  - The accuracy parameter $\rho = 200$ is used.
  - Each localized integral needs 200 discretization points.

  To obtain this accuracy, the GO-BIE method takes slightly more than 2.5 minutes to produce μ^s under the same machine configuration.

The results shown in Table 4.2 are for scattering by a unit circle. We obtained similar results for scattering by general convex objects.

To further demonstrate the amazing efficiency of this method, we produced the result for $k = 100000$. Figure 4.13 shows the plot of μ^s for $k = 100000$. Direct numerical methods cannot be used at such high frequency.

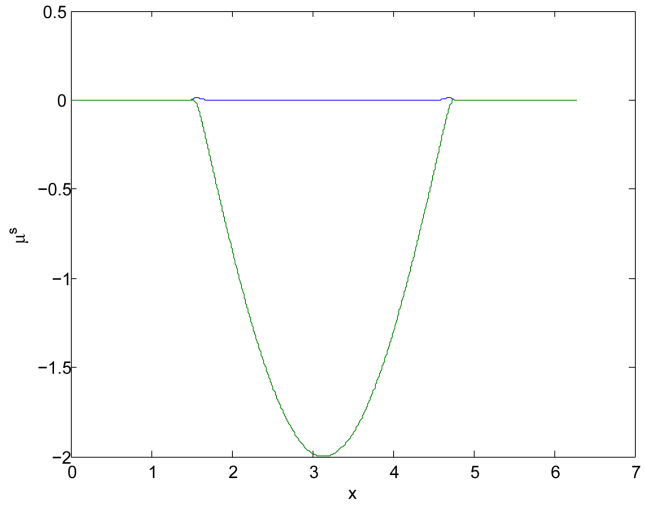

Figure 4.13: Real and imaginary parts of μ^s from the scattering by a circular cylinder at $k = 100000$. Note the sharp transition of the function from the shadow to illuminated region.

4.6 Extension to multibody scattering

4.6.1 Analysis

Our GO-BIE method at its current state of development, while being efficient, is only useful when solving wave scattering by convex objects. This is because for non-convex objects, the density $\mu(x) = \frac{\partial(u+u^I)}{\partial n}(x)$ no longer has a simple form $\mu(x) = \mu^s(x)ke^{ik\alpha\cdot x}$, where μ^s is a slowly varying function. However, μ still can be approximated as a superposition of the form

$$\mu(x) = \sum_{m=0}^{M} \mu_m^s(x) e^{ik\phi_m(x)},$$

where μ_m^s varies slowly on the boundary.

There have been many attempts at this approach for the multiple scattering problems [47, 36]. However, there has been no efficient and robust numerical methods available yet. We aim to provide such a method for the two-body scattering problem and then generalize the method to n-body settings.

Consider the problem of wave scattering by two disjoint convex objects $\{\Omega_i\}_{i=1,2}$ (see Figure 4.14 for example). Our goal is to solve $\mu = \{\mu_1, \mu_2\}$, where each μ_i is the normal derivative of the total field defined on the boundary $\Gamma_i = \partial\Omega_i$. Our proposed research is based on the following observations.

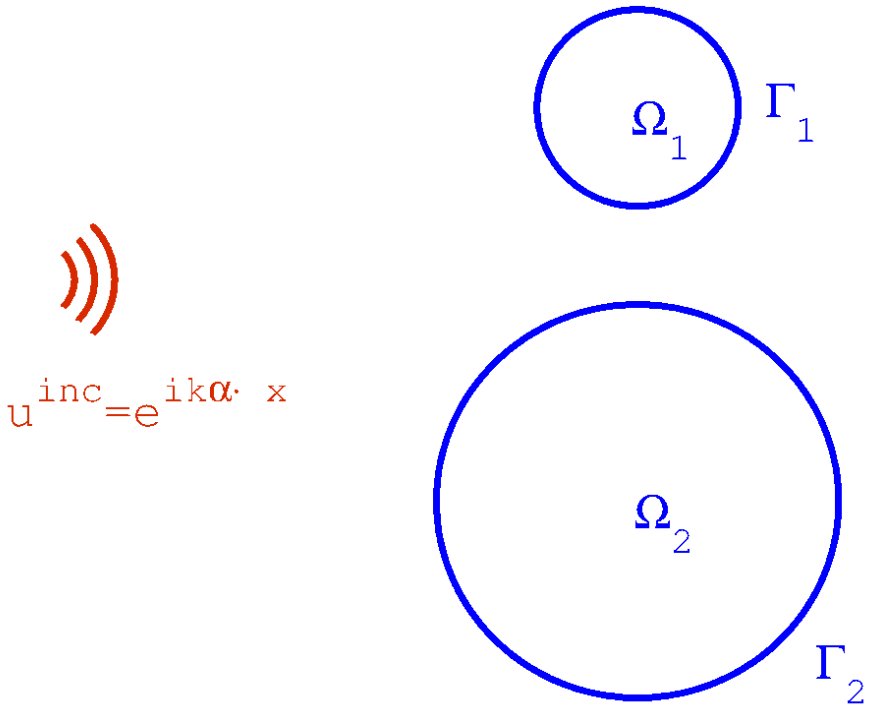

Figure 4.14: A plane wave incident scattered by two circular obstacles

Iterative scattering. The boundary density μ has an additive property and can be solved iteratively. The outgoing field from the first iteration can be treated as the incoming field for the second iteration and so on. The iterative scattered fields from these multiple reflections are added together to form μ. This property is described mathematically as

$$\mu = \sum_{m=0}^{\infty} \mu^m = \sum_{m=0}^{\infty} T^m g, \tag{4.17}$$

where

- $g = \mu^0$ is the normal derivative of the total field arising from the first reflection,
- $\mu^m = T\mu^{m-1}$ is the contribution of the wave after having gone through m geometrical optics reflections,
- and the operator T encodes the computation for each single reflection.

The detailed descriptions of T, g, and the derivation of (4.17) are provided in Appendix B.

Compact representation in each iteration. At each iteration, the density μ^m, composing of μ_1^m and μ_2^m, admits of a compact representation

$$\mu_i^m(x) = u_{j\to}^{m-1}(x)\eta_i^m(x), \quad i = 1, 2,$$

where η_i^m is expectedly a slowly varying function of x and $u_{j\rightarrow}^{m-1}$ is the scattered field generated by Ω_j in the previous iteration, which is given via the formula

$$\begin{aligned} u_{j\rightarrow}^{m-1}(x) &= -\int_{\Gamma_j} G(x,y)\mu_j^{m-1}(y)\, ds(y) \\ &= -\int_{\Gamma_j} G(x,y)u_{i\rightarrow}^{m-2}(y)\eta_j^{m-1}(y)\, ds(y) \end{aligned}$$

for $x \in \mathbb{R}^d \setminus \Omega_j$ and for each $j \neq i$. We expect that $\|u_{j\rightarrow}^{m-1}\|$ decays with respect to m, but $\|\eta_i^m\|$ stays fixed, approximately.

With the information obtained from the past iteration, such as $u_{i\rightarrow}^{m-2}$ and its phase, the scattered field at each point $x \in \mathbb{R}^d \setminus \Omega_j$ can be approximated in $O(1)$ using the localized integration technique described in Section 4.3.

However, solving for η_i^m via the GO-BIE method needs the phase information of the incoming field $u_{j\rightarrow}^{m-1}$ $(j \neq i)$, which is not readily available. Here, we propose to compute the phase of $u_{j\rightarrow}^{m-1}$ numerically from the field $u_{j\rightarrow}^{m-1}$ itself via a simple formula

$$\phi_{,j\rightarrow}^{m-1}(x) = (ik)^{-1} \ln\left(\frac{u_{j\rightarrow}^{m-1}(x)}{\left|u_{j\rightarrow}^{m-1}(x)\right|}\right). \tag{4.18}$$

Since the field $u_{j\rightarrow}^{m-1}(x)$ can be approximated accurately in $O(1)$ FLOPs, its phase can be determined in $O(1)$ FLOPs as well. This method of extracting the phase of a wave field is similar to the approach in [10]. Our way of determining $\phi_{j\rightarrow}^{m-1}$ is in contrast with the approach in [47], where the authors used geometrical ray tracing to approximate this phase function. We believe that

using ray tracing would cancel the ability of the GO-BIE method to control the error bound.

4.6.2 Preliminary results

We have generated some preliminary results for this multibody approach. Figure 4.14 illustrates the problem setting for the results shown below.

Figure 4.15 shows the density $\mu^s(x)$, defined as $\frac{\mu(x)}{ke^{ik\alpha\cdot x}}$, on the surface of two bodies. We can see that μ^s varies slowly on the top part of Γ_1 and on the bottom part of Γ_2 since these parts are almost unaffected from the multiple reflections. However, μ^s is oscillatory on the rest of Γ due to the strong multiple reflection effect.

Figure 4.16 shows that the energy of the scattered field decays exponentially after each geometrical optics reflection. This property allows us to solve μ in a practical number of iterations.

Figure 4.17 shows η_i^m $(i = 1, 2)$ in different iterations. In this test, for $x \in \Gamma_i$, we computed $u_{j\to}^{m-1}(x)$ and $\mu_i^m(x)$ using direct simulation and $\eta_i^m(x)$ was obtained by dividing $\mu_i^m(x)$ by $u_{j\to}^{m-1}(x)$. Our results confirm that η_i^m, for each object and each iteration, is indeed a smooth and slowly varying function on the object's boundary. We also see that the density η^m peaks at the left of the obstacles when $m = 1$, but it peaks at the bottom of Γ_1 and the top of Γ_2 when $m > 1$.

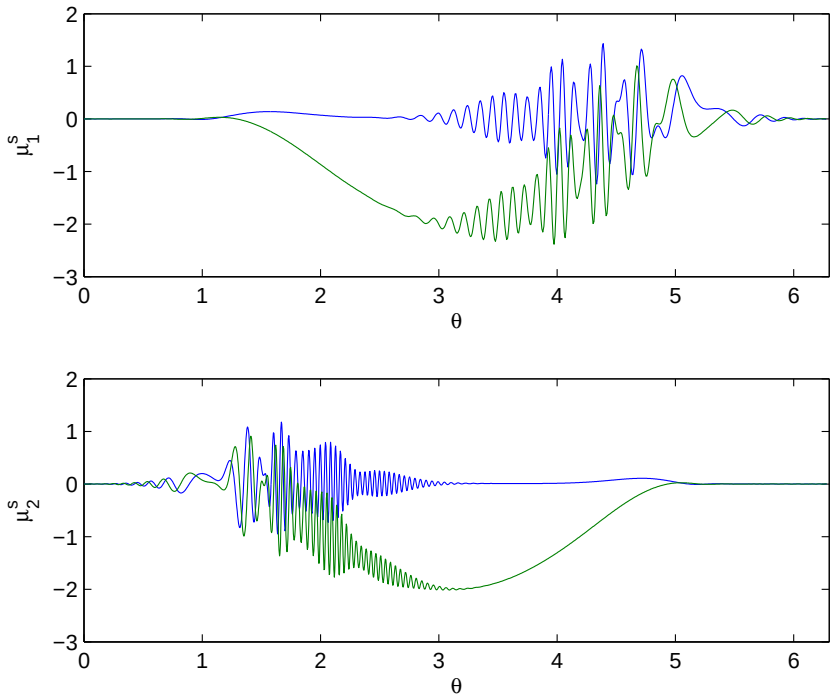

Figure 4.15: Solution to the problem setting described in Figure 4.14. For this test, $k = 50$, Ω_1 has radius 1, Ω_2 has radius 2, and the distance between the two circles is 2 (*i.e.* the distance between the two centers is 5). Top: the plot of μ_1^s against the parametric angle of Γ_1; Bottom: the plot of μ_2^s against the parametric angle of Γ_2. The boundary density $\mu^s = \{\mu_1^s, \mu_2^s\}$ shown here was obtained using full-scale direct simulation.

Table 4.3 and Figure 4.18 show the exponential convergence of the approximated solution with respect to the number of reflections. It also shows that adding more iterations is not very costly because the matrices involved depend only on the geometry and not on the values of incoming fields.

Finially, We tested the error sensitivity with respect to the frequency k,

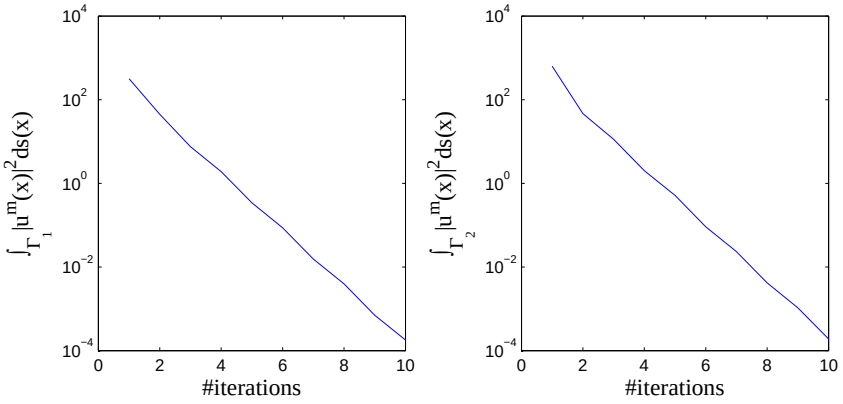

Figure 4.16: The L_2 energy of the wave field, restricted to Γ, decays exponentially with respect to the number of geometrical optics reflections it has gone through.

#iterations	Error	CPU Time
2	7.6e-1	6.2
4	2.3e-1	8.4
8	2.4e-2	12.6
16	4.3e-4	21.3
32	2.1e-7	36.9

Table 4.3: Error and CPU time (in seconds) for different numbers of iterations. The solvers were implemented in MATLAB and ran on a dual-core 2.8 GHz Intel CPU. The problem setting is described in Figure 4.15.

for the same number of geometrical optics iterations. Table 4.4 shows that the number of iterations required stays independent of k, for the same accuracy level.

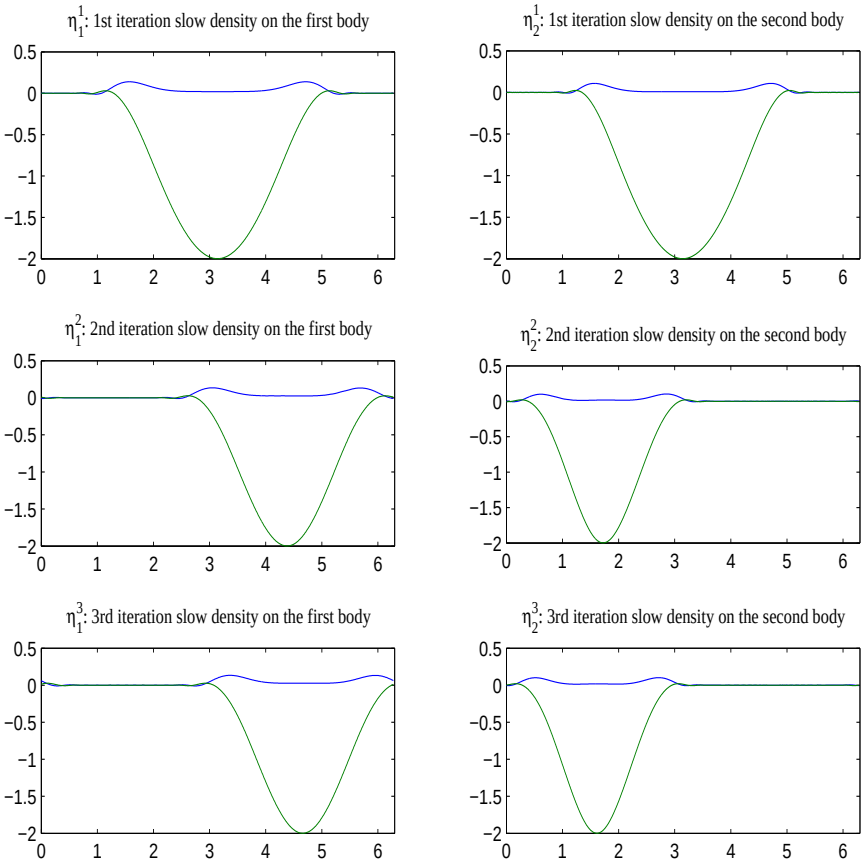

Figure 4.17: Plot η_i^m $(i = 1, 2)$ for different values of m and i.

4.7 Concluding Remarks

We have presented an $O(1)$ method for solving high frequency wave scattering by a convex obstacle. There are two key elements of this so-called GO-BIE method.

1. The normal derivative of the total field $\mu(x) = \frac{\partial(u+u^I)}{\partial n}(x)$ on the bound-

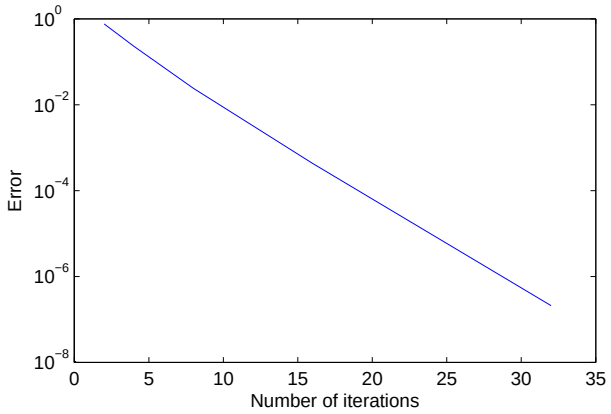

Figure 4.18: Convergence test of the iterative scattering method.

k	Error
10	4.8e-4
20	6.2e-4
40	3.0e-4
80	4.5e-4
160	6.0e-4

Table 4.4: Error of the iterative method, using 10 iterations, for different frequencies.

ary, which has its own BIE formulation, can be compactly represented as $\mu^s(x)ke^{ik\alpha\cdot x}$ for $x \in \Gamma$. One can thus use a finite number of points, irrespective of k, to sample the unknown function μ^s.

2. For each sampling point x_i, the corresponding integral (in the formulation of the induced matrix equation) can be approximated efficiently using the localized integration algorithm.

Our direct contribution to the GO-BIE method for single convex body scattering is the robust analysis and improvements to the localized integration

technique originally proposed in [18].

We also devised a new algorithm for high frequency scattering by multiple convex bodies and provided preliminary results for the 2-body scattering setting. This algorithm is based on two key observations.

1. We can solve μ in an iterative way, where the scattered field from the first geometrical optics reflection is treated as the incoming field for the second reflection and so on.

2. The sub-scattered field in each geometrical optics reflection can be solved using the GO-BIE method for one convex body scattering.

For future work, we propose the following improvements to this GO-BIE method.

1. One is a hybrid method that combines the GO-BIE method where it is applicable and a fine-scale method based on DFMM on the other part of the geometry, where the solution does not admit of a compact representation. Chapter 5 describes one such method.

2. The other is a multiple scattering GO-BIE method to handle the case of multiple wave reflections and diffractions. This extension to our current 2-body scattering solver can be done in a few independent ways.

   (a) Use of the GO-BIE technique in each geometrical optics reflection for optimal performance. Our preliminary results show that each

η_i^m is indeed smooth and slowly varying. Hence, the GO-BIE can be utilized to solve η_i^m, and hence μ_i^m.

(b) Extension to handle self-reflections by a non-convex object (see Figure 4.2).

(c) Extension to handle n bodies, where n can be greater than 2. One difficulty associated with a general n-body scattering solver is that to solve for μ_i^m ($i \in \mathcal{I}$), we have to consider a large number of incoming fields that have gone through different paths of reflection. The number of paths increases exponentially with respect to m. In the 2-body case, for each i and m, there is only one incoming field corresponding to one path of reflection.[3]

(d) Extension to handle corners and edges. The superposition of μ should account for the diffraction terms produced by non-smooth parts of the surfaces $\{\Gamma_i\}$. An analysis for this type of problem can be found in [27].

[3] There is only one path of reflection in a back-and-forth fashion: $\Omega_i \to \Omega_j \to \Omega_i \to \ldots$ ($j \neq i$) for each $i = 1, 2$.

Chapter 5

Hybrid Method

5.1 Motivation

In many applications that involve high frequency scattering, the scatterers can often be represented as a combination of a large convex part and a complicated small structure (see Figure 5.1 for example). One typical example is the structure of a building with an antenna at the top. For scatterers of this type, the ansatz introduced in (4.2) is valid only on the large convex part, while the density $\mu(x)$ can be arbitrarily oscillatory at the small structure. As a result, neither method discussed in the previous two chapters is optimal when k is very large and the small structure is relatively small compared to the whole domain. The method based on DFMM samples the whole boundary densely even though the density $\mu(x)$ has an efficient representation for the most part of the boundary; on the other hand the GO based integral equation method faces difficulty in handling the non-convex part.

In this chapter, we discuss a hybrid method that combines the advantages of both methods [40]. The main idea is simply to partition the boundary into two overlapping parts and then use the appropriate representation for

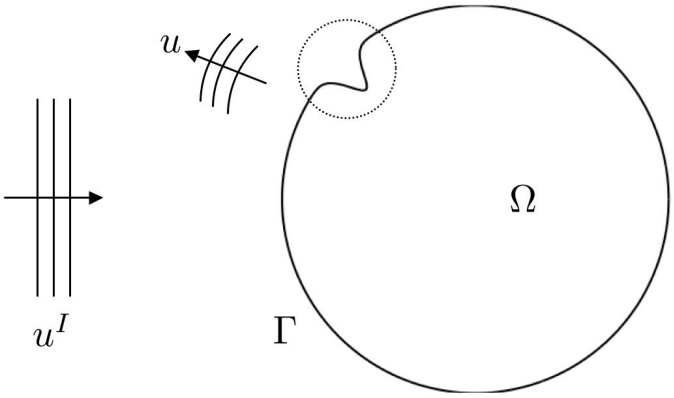

Figure 5.1: The scattering of an incoming plane wave u^I from a smooth scatterer Ω. u is the scattered field. The boundary Γ of the scatterer can be represented as a large convex part and a small structure (inside the small circle).

each part.

5.2 Analysis

Domain decomposition. For simplicity, suppose that we are solving the problem of high frequency wave scattering from a unit circular cylinder which has a small indentation of size L much smaller than the size of the overall object. Note that, due to the problem of multiple scatterings in the non-convex part, μ^s as defined in (4.2) is no longer slowly varying (see Figure 5.2).

We start by introducing an open covering $\{\Gamma_1, \Gamma_2\}$ of the boundary Γ (see Figure 5.3(a)). The non-convex region Γ_1 covers the small structure and is of size $O(L)$ while Γ_2 contains the large convex part. The overlapping region

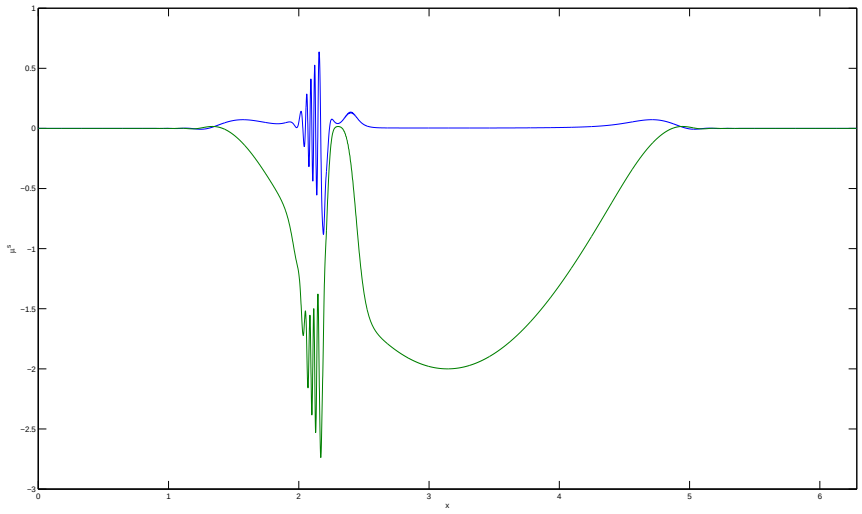

Figure 5.2: Real and imaginary parts of μ^s for the scattering from a circular cylinder with a small non-convex structure: μ^s is highly oscillatory in the small non-convex region due to multiple reflections.

$\Gamma_1 \cap \Gamma_2$ is of size $O(\frac{1}{k})$. Next, we construct a partition of unity $\{\alpha_1(x), \alpha_2(x)\}$ subordinate to the covering $\{\Gamma_1, \Gamma_2\}$, *i.e.* $\alpha_1(x) + \alpha_2(x) = 1$, $\text{supp}(\alpha_1) \subset \Gamma_1$, and $\text{supp}(\alpha_2) \subset \Gamma_2$.

Let us estimate how many samples are required in order to approximate accurately the unknown density $\mu(x) = \frac{\partial(u+u^I)}{\partial n}(x)$ for $x \in \Gamma$. We first consider the restriction of $\mu(x)$ to Γ_1. The oscillation of $\mu(x)$ on Γ_1 can be quite arbitrary, depending on the shape of the small structure. However, since Γ_1 is of size $O(L)$, we only need to sample $\mu(x)$ at a set of $O(kL)$ equally spaced points, called P_1, on Γ_1 (see Figure 5.3(b)). Second, $\mu(x)$ on Γ_2 has a simple form $\mu(x) = ke^{ik\alpha\cdot x}\mu^s(x)$, where μ^s is slowly varying on Γ_2. Therefore, in order

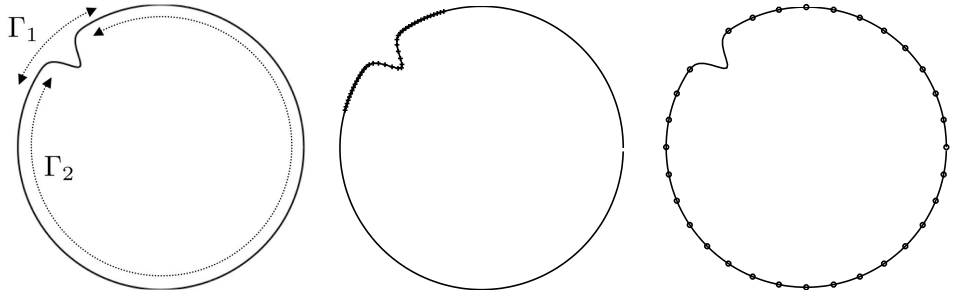

Figure 5.3: (a) The boundary of the scatterer is represented as the union of Γ_1 and Γ_2. Γ_1 contains the small structure while Γ_2 covers the large convex part. (b) The sample points P_1 in the dense discretization of Γ_1. (c) The sample points P_2 in the sparse discretization of Γ_2.

to approximate $\mu(x)$ on Γ_2, we can sample $\mu(x)$ on Γ_2 with a fixed number of points, called P_2. Putting the sets P_1 and P_2 together, one only needs $O(1+kL)$ Nyström points to represent $\mu(x)$.

Since this sampling scheme relies on the fact that μ^s can be represented accurately on Γ_2 by a fixed number of data points, we will consider boundary integral equation (4.4), *i.e.*

$$\frac{1}{2}\mu^s(x) + \int_\Gamma K(x,y)\mu^s(y)\,ds(y) = ix\cdot\alpha - i\frac{\gamma}{k}, \quad x\in\Gamma,$$

and adopt the Nyström approach for the discretization of the equation.

Hybrid integration. The next step in the Nyström method is the evaluation of the integral

$$\int_\Gamma K(x,y)\mu^s(y)\,ds(y) = \int_\Gamma \left(\frac{\partial G(x,y)}{\partial n(x)} - i\gamma G(x,y)\right) e^{ik\alpha\cdot(y-x)}\mu^s(y)\,ds(y) \tag{5.1}$$

given the values of $\mu^s(x)$ at $x_i \in P_1 \cup P_2$. Based on the partition of unity $\{\alpha_1(x), \alpha_2(x)\}$, we can represent $\mu^s(x)$ as a sum of two parts:

$$\mu^s(x) = \alpha_1(x)\mu^s(x) + \alpha_2(x)\mu^s(x).$$

Let us denote $\mu_1(x) = \alpha_1(x)\mu^S(x)$ and $\mu_2 = \alpha_2(x)\mu^s(x)$. Using this decomposition to rewrite the integral (5.1) gives

$$\begin{aligned}
&\int_\Gamma K(x,y)\mu^s(y)\,ds(y) \\
&= \int_\Gamma K(x,y)\alpha_1(y)\mu^s(y)\,ds(y) + \int_\Gamma K(x,y)\alpha_2(y)\mu^s(y)\,ds(y) \\
&= \int_{\Gamma_1} K(x,y)\mu_1^s(y)\,ds(y) + \int_{\Gamma_2} K(x,y)\mu_2^s(y)\,ds(y),
\end{aligned} \tag{5.2}$$

where in the last step we use the facts that support of $\alpha_t(y)$ is Γ_t for $t = 1, 2$.

To calculate the first integral:

- DFMM is used to compute the interaction of the points in P_1. Since there are $O(kL)$ points in P_1, evaluating this integral for all $x_i \in P_1$ using DFMM takes at most $O\left(kL\log(kL)\right)$ floating-point operations (FLOPs). This is equivalent to $O(1)$ operations if the size of the perturbation is

$L = O(1/k)$ or $O\left(k^{1/2}\log k\right)$ if $L = O(k^{-1/2})$.

- for the interaction of the points in Γ_1 (sources) and $\Gamma \setminus \Gamma_1$ (targets), we use direct computation. Since there are only a fixed number of sampling points in $\Gamma \setminus \Gamma_1$, evaluating this integral for points in $\Gamma \setminus \Gamma_1$ would take at most $O(kL)$ operations.

Hence, we can evaluate the first integrals $\left\{\int_{\Gamma_1} K(x_i, y)\mu_1^s(y)\, ds(y)\right\}_{x_i \in P_1 \cup P_2}$ in $O\left(kL\log(kL)\right)$ time.

For the evaluation of the second integral, we can resort to the localized integration technique discussed in the previous chapter. However, this can get tricky when L is small since α_2 (and hence μ_2) would change rapidly in the transition regions (see Figure 5.4).

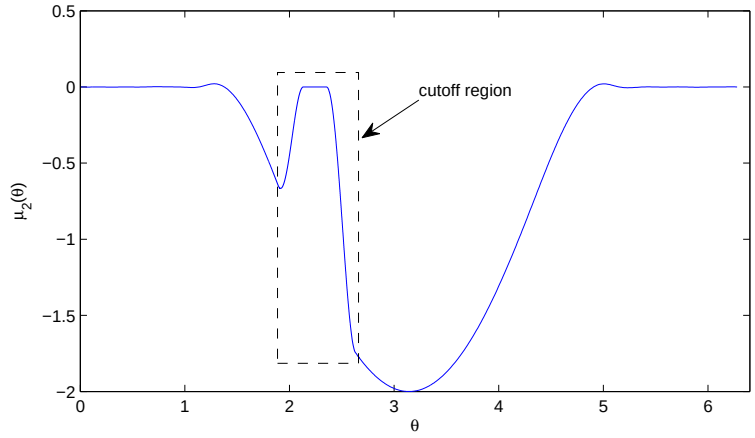

Figure 5.4: The function $\mu_2(x)$ changes rapidly in the transition regions, due to the sharp cutoff $\alpha_2(x)$.

Due to this reason, the oscillatory integrals $\int e^{ik\phi(y)}\mu_2(y)\, ds(y)$ over

these transition regions no longer vanish when $k \to \infty$. A simple fix is to treat these regions similar to the way we treat the shadow regions. Therefore, for each target point x_i, there are 4 types of critical regions that we need to cover with localized integration:

- the target point region, the stationary regions, and shadow regions, all of which were detailed in the previous chapter;
- and the two transition regions, which have size of $O(k^{-1})$.

Since we can use a fixed number of points to cover the transitions regions, the extra costs do not change the $O(1)$ complexity of the localized integration method. Since there are $O(kL)$ sampling points in $P_1 \cup P_2$, evaluating the second integral takes $O(kL)$ FLOPs.

To summarize, using this hybridization idea, it is possible to evaluate (5.1) fast, *i.e.* in $O\left(kL \log(kL)\right)$ FLOPs.

5.3 Algorithm

Here, we present the detailed description of the algorithm.

1. **Discretization.** Recall that we need to collocate points smoothly on the boundary such that they are dense in Γ_1 because of the highly oscillatory behavior of μ^s in this region, and get coarser in the region $\Gamma \setminus \Gamma_1$ as μ^s is slowly varying here. In order to do that, we introduce a smooth and

periodic transformation function τ that maps a uniform discretization $\{z_i\}_{i=1}^N \subset \Gamma$ onto the same boundary such that $\{\tau(z_i)\}_{i=1}^M$ $(M < N)$ are uniform and dense in Γ_1 but that $\{\tau(z_i)\}_{i=M+1}^N$ lie sparsely in $\Gamma \setminus \Gamma_1$. Figure 5.5 illustrates the the behavior of the transformation function τ.

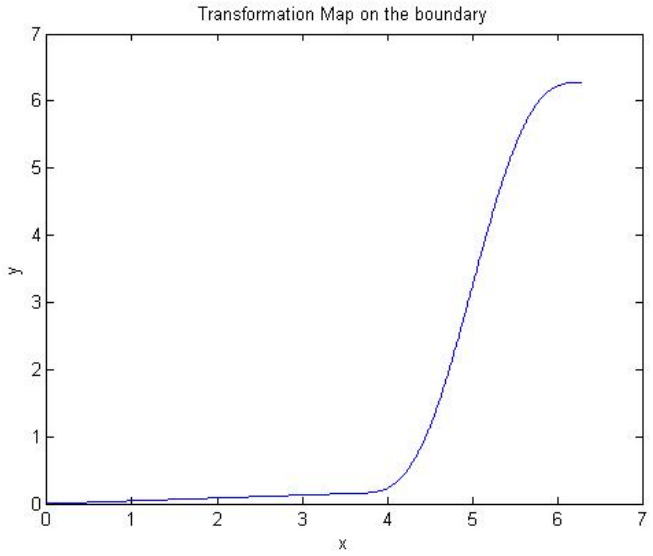

Figure 5.5: The smooth and periodic function τ transforms $M + N$ points sampled uniformly on Γ onto the same domain in a non-linear fashion. The first M points are squeezed into the small region Γ_1 while the rest are mapped into a much larger region.

Let us call $x_i = \tau(z_i)$ for all $i = 1..N$. Our goal is to compute μ^s valued at these points.

2. **Integrators Setup.** Construct the localized integration method and DFMM. This involves pre-computations of the integrations points and the kernels at those points, as well as the tree construction for DFMM.

3. **Matrix solver.** At each GMRES iteration:

   - **Interpolation.** Interpolate μ^s obtained from the previous GMRES iteration. This can be done efficiently by first using FFT refinement on a denser grid and then using a cubic spline for the evaluation of u^s at arbitrary points on the boundary accurately.

   - **Dense integrator.** Run DFMM once to compute

$$\int_{\Gamma_1} K(x_i, y)\mu^s(y)\alpha_1(y)\, ds(y) \tag{5.3}$$

     for all $i = 1..M$. Note that DFMM does not take into account that K is singular when y is in the neighborhood of x_i. Hence, a singularity correction is needed to correct the results obtained from DFMM. The idea of the singularity correction rule is similar to the one discussed in the previous chapter, *i.e.* using local refinement and special quadrature weights.

   - **Sparse integrator.** Evaluate (5.3) for each $i = M+1, .., N$ using the localized integration method to compute the integral

$$\int_{\Gamma_2} K(x_i, y)\mu^s(y)\alpha_2(y)\, ds(y) \tag{5.4}$$

   - **Aggregation.** Combine the results above, via the formula

$$\frac{1}{2}\mu^s(x_i) + \int_{\Gamma_1} K(x_i, y)\mu^s(y)\alpha_1(y)\, ds(y) + \int_{\Gamma_2} K(x_i, y)\mu^s(y)\alpha_2(y)\, ds(y)$$

to complete each matrix-vector multiplication (in the framework of GMRES).

5.4 Numerical Results

In this section, we present some numerical and performance results for the hybrid method described above. Our test example is the object displayed in Figure 5.3 and its boundary is parametrized by $x = (r(\theta)\cos\theta, r(\theta)\sin\theta)$, with

$$r(\theta) = 1 - Le^{-\frac{(\pi - \bmod(\theta+0.9, 2\pi))^2}{2L^2}} \tag{5.5}$$

for $\theta \in [0, 2\pi)$.

Figure 5.6 shows the true solution $\mu^s(x)$ for $k = 100$ and $L = O(k^{-1/2})$, computed using a DFMM-based solver (DFMM solver for short) on the whole boundary with highly dense discretization.

To study the performance of the hybrid method, we benchmarked it against a DFMM-based solver,[1] which is the current state-of-the-art method for general high frequency wave scattering. We tested for a wide range values of k and for $L = O(k^{-1/2})$ as well as for $L = O(k^{-1})$. Both the DFMM and hybrid methods are implemented and optimized in C++ and ran on a 3.2 GHz Intel i5 CPU.

[1] For the tested geometries, which is a smooth object in two dimensions, we used the Nyström discretization scheme due to its simplicity and efficiency.

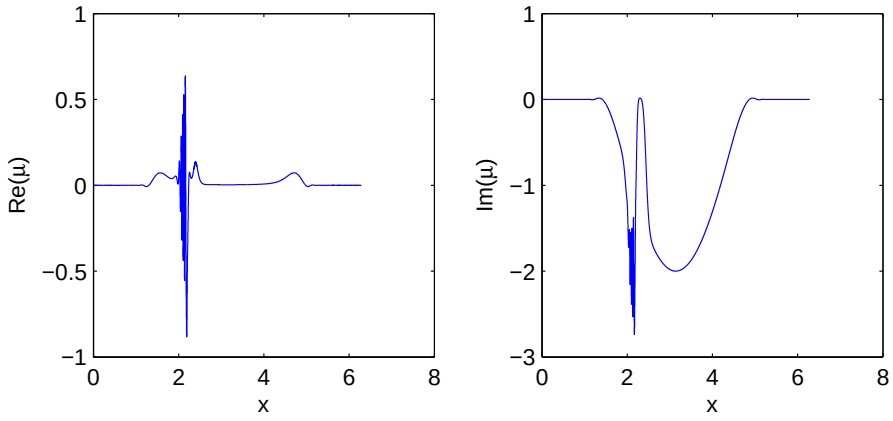

Figure 5.6: The real and imaginary parts of $\mu^s(x)$. The scatterer is given by the formula in (5.5) and $k = 100$. $\mu^s(x)$ is smooth in the convex part and is highly oscillatory in the small non-convex structure.

Table 5.1 and Figure 5.7 display the benchmarking results regarding the CPU time for each GMRES iteration and the number of such iterations, while Table 5.2 and Figure 5.9 compare the two respective solvers in regards of their total runtimes, when $L = (k^{-1/2})$.

The numerical results for the described setting and for $L = O(k^{-1/2})$ show that:

1. The integral evaluation using the hybrid method takes only $O(\sqrt{k}\log k)$ versus $O(k\log k)$ if using full-scale DFMM.

2. The advantage of the hybrid method is not limited to just the number of sampling points and lower quadrature cost. The results show that the iterative solver also converges faster using the hybridized discretization scheme. It also implies that the condition number of the system depends

k	Time per iteration		#iterations	
	Hybrid	DFMM	Hybrid	DFMM
50	0.14	1.19	15	24
100	0.21	2.46	16	29
200	0.33	5.23	17	36
400	0.51	10.80	19	45
800	0.74	23.30	22	54
1600	1.08	49.05	23	68
3200	1.57	105.49	26	84
6400	2.26	222.13	28	104
12800	3.39	476.20	31	n/a

Table 5.1: Performance comparison between the hybrid and DFMM solvers with respect to the averaged time spent per iteration (in seconds) and also the number of iterations needed, when $L = O(k^{-1/2})$. The 10^{-3} error bound control was specified for both methods. While the DFMM solver did not finish the computations within a reasonable time frame, the hybrid solver produced the solution within a relatively short time.

not only on the geometry but also on the discretization scheme. With a reasonable discretization scheme, we have verified numerically that the condition number of the discrete system grows as $O(N^{1/3})$, where N is the number of sampling points.

3. Combining the two points above, the hybrid method significantly boosts the performance when solving wave scattering by a near-convex object.

We also tested our hybrid solver when $L = O(k^{-1})$. Table (5.3) shows the accuracy level of the solutions computed by the hybrid method with fixed numbers of sampling and integration points. The results verify that the hybrid method effectively has $O(1)$ complexity if the small structure has size scale of a few wavelengths.

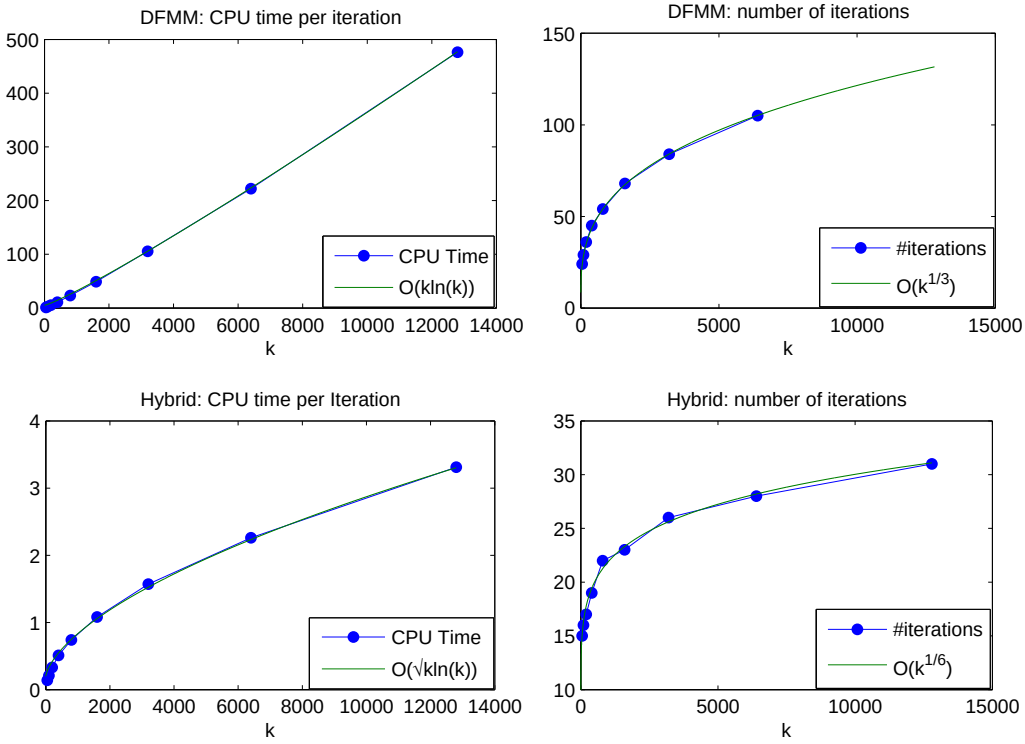

Figure 5.7: Growth of the CPU time per iteration and the number of iterations with respect to k, for both the DFMM and hybrid solvers, when $L = k^{-1/2}$. Top left: the CPU time per iteration using the DFMM solver has $O(k \log k)$ growth rate; Top right: the number of iterations using the DFMM solver has $O(k^{1/3})$ growth rate; Bottom left: the CPU time per iteration using the hybrid solver has $O(\sqrt{k} \log k)$ growth rate; Bottom right: the number of iterations using the hybrid solver has $O(k^{1/6})$ growth rate. The data used are from Table 5.1.

In each case, the growth is plotted in comparison with its theoretical growth $O(\sigma(k))$. Two parameters a and b are used for fitting the theoretical growth function $a\sigma(k) + b$.

k	Error of Hybrid method	CPU Time (seconds) Hybrid	DFMM
50	2.3 e-4	3.87	28.96
100	3.5 e-4	4.43	71.51
200	4.1 e-4	6.91	188.55
400	4.2 e-4	11.31	486.38
800	5.4 e-4	18.34	1259.56
1600	4.7 e-4	27.83	3340.74
3200	4.9 e-4	45.12	8873.70
6400	5.7 e-4	68.22	23572.40
12800	n/a	111.99	n/a

Table 5.2: Numerical error of the hybrid method and its CPU time, in comparison with the performance of the DFMM solver, for the case $L = O(k^{-1/2})$. The numerical error of a simulated solution is the discrete L_∞ error when compared against the corresponding "exact" solution, which was computed using the DFMM-based direct solver with 10^{-5} error tolerance. See Figure 5.8 for the plot of errors on the whole boundary. As in Table 5.1, we were not able to produce the results using the DFMM solver for $k > 6400$.

5.5 Concluding Remarks

Two efficient existing numerical techniques for high frequency wave scattering and their hybridization have been presented. The directional fast multipole method can handle general geometries and general scattered fields but requires a large number of unknowns. The GO-BIE method described in the previous chapter requires much fewer unknowns but only applies to simpler geometries and scattered fields.

Our analysis and simple two-dimensional numerical model problem show the potential for hybridization of the two methods into a technique that exploits the best properties of both methods. The low computational complex-

k	Errors of μ^s	
	120 points	240 points
50	6.5e-4	3.7e-6
100	5.8e-4	4.2e-6
200	7.0e-4	5.6e-6
400	7.5e-4	5.9e-6
800	8.2e-4	7.6e-6
1600	9.1e-4	8.1e-6
3200	8.7e-4	6.1e-6
6400	9.6e-4	7.6e-6

Table 5.3: Accuracy of the hybrid method when $L = O(k^{-1})$ and when a fixed number of unknowns were used. The errors are computed by comparing the approximate solutions against the reference solutions, via the discrete L_2 error formula. The reference solutions were generated with the DFMM solver with 10^{-6} accuracy level. To obtain L_2 errors of less than 10^{-3}, we used 120 sampling points. The method converged in 14-15 iterations for all k and the whole program took on average 4.2 seconds to finish. For 10^{-5} error bound, we used 240 sampling points. The method converged in 19-21 iterations and the program finished within 38 seconds on average.

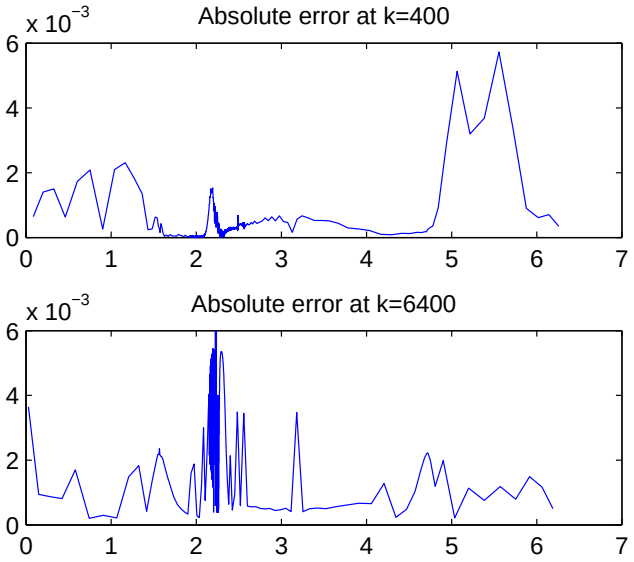

Figure 5.8: The absolute value of the difference between the true solution and the solution computed using the hybrid method, at $k = 400$ and at $k = 3200$.

ity of the geometrical optics based method can essentially be retained even if the geometry locally requires a fine discretization.

We would like to point out that the classical geometrical optics methods applied to a sub domain of the scattering object have been hybridized in the literature, especially in the field of antennas, with full numerical boundary integral techniques applied to the remaining part of the scatterer. The name method of moments (MM) is often used instead of numerical boundary integral methods. Examples of the coupling GTD-MM and of PO-MM are given in [21, 60, 56]. The new aspects of our hybrid method are:

1. Two very recent and highly efficient methods are involved. While DFMM

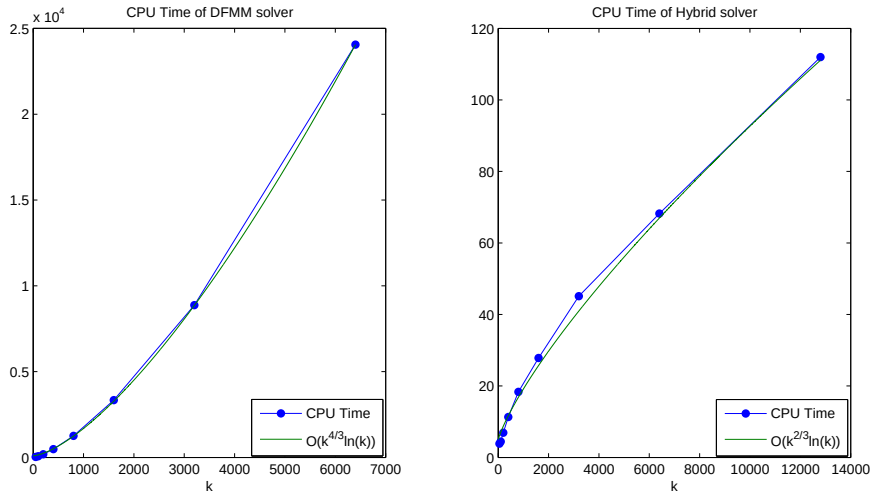

Figure 5.9: Growth of the total CPU time for both the DFMM and hybrid solvers. Left: the total CPU time using the DFMM solver has $O(k^{4/3}\log k)$ growth rate; Right: the total CPU time using the hybrid solver has $O(k^{2/3}\log k)$ growth rate; The data used are from Table 5.2. In each case, the growth is plotted in comparison with its theoretical growth $O(\sigma(k))$. Two parameters a and b are used for fitting the theoretical growth function $a\sigma(k)+b$.

accelerates the computations where small-scale resolution is needed, the GO-BIE method provides the control of the accuracy of the solution where high frequency asymptotics is used.

2. The technique naturally provides smooth transition between the two regions through the partition of unity and also allows a full two-way coupling in the overlapping regions.

We can even further speed up the computation of the second integral of (5.2) if the small structure in Γ_1 is relatively smooth. In the current form, our hybrid

method performs the localized integration for each Nyström point $x_i \in P_1$. Instead, we can perform the localized integrator at few points in Γ_2 and then use Fourier or spline interpolation to obtain the values at the Nyström points $x_i \in P_1$.

Chapter 6

Conclusions and Future Work

6.1 Results.

In this dissertation we have presented three new fast numerical techniques for high frequency wave scattering problems.

A. The first is a distributed directional fast multipole method (DFMM) algorithm and an accelerated boundary integral equation (BIE) solver based on this distributed DFMM. Such a parallel solver is particularly important for the simulation of wave propagation in a cluttered environment, where both asymptotic and full-scale methods are too costly in time and memory to be used on a single machine.

Despite the complexity of the directional multilevel algorithm, which has several different phases and combines many independent techniques, we were able to achieve good scalability for up to hundreds or thousands of processors (for realistic problem settings). The scalability is reduced when the local space that each processor owns is too small compared to the global domain.

The new fast parallel solver also allowed us to examine the convergence

properties of iterative solvers when solving high frequency wave propagation in cluttered environments.

B. The second is a geometrical optics boundary integral equation (GO-BIE) algorithm that can solve high frequency wave scattering by a convex object within a fixed time, independent of k, for a prescribed error bound. The convexity of the geometry is an important assumption since it gives rise to a compact representation of the solution based on asymptotic analysis. A localized integration technique is then used for fast integral computations. These ideas, when combined, yield an $O(1)$ method with respect to the wave frequency. For single convex object scattering, our main contribution is the analysis and improvement on the localized integration technique originally proposed in [18].

We also devised an algorithm and provided preliminary numerical results for the 2-body scattering. The idea is to separate the wave field into different components, where the scattered field from one geometrical optics reflection is treated as the incoming field for the next reflection. We verified numerically that the field component from each reflection has a compact representation. This gives rise to the use of the GO-BIE method on each object during each geometrical optics iteration.

C. The third is a hybrid method that combines the GO-BIE method where it is applicable and a full scale DFMM-based method on the rest of the geometry, where the solution does not admit of a compact asymptotic representation. The new aspects of our hybrid method in this dissertation are:

- Two very recent and highly efficient methods are involved. While DFMM accelerates the computations where small-scale resolution is needed, the GO-BIE method provides minimal computational cost and control of the accuracy of the solution where high frequency asymptotics is used.

- The technique naturally provides smooth transition between the two regions through the partition of unity and also allows a full two-way coupling in the overlapping regions.

6.2 Future Work

The methods described above are not without limitations. The common theme is that they are efficient but only for specific problem settings. Our future work will focus on improvements to these methods so that they are more robust and applicable to wider classes of high frequency wave problems. In particular, three new lines of research will be considered.

A more robust parallel DFMM. In our current implementation of the parallel DFMM, each processor starts constructing its local tree structure at the box occupying the space partition that it owns. When the geometric partition that a processor owns is smaller than the mother boxes, defined as the largest boxes one needs to visit in the DFMM framework, the scalability of the parallel DFMM starts degrading. This is because we cannot exploit the multilevel property of DFMM at the few box levels above the partitioning

level. This multilevel characteristic is a key element of DFMM that brings the cost of matrix-vector multiplication down to $O(N \log N)$ from $O(N^2)$, where N is the number of points.

To remedy this problem, we propose a dynamic processor allocation scheme, described as following.

- The tree construction starts at the mother boxes, which are processed by a subset of processors, if there are more processors than mother boxes.
- At lower levels, the active processors allocate jobs to more processors since there are more boxes involved. This allocation process stops when all processors start their computations.

Another improvement to the parallel algorithm is to use a heterogeneous parallel architecture. One increasingly popular hybrid architecture is the combination of MPI (Message Passing Interface) with GPU technology (Graphics Processing Unit).

A new preconditioner for the boundary integral equation. Our results in Chapter 3 shows that when propagating in a cluttered or porous medium, the wave field can be trapped. The number of iterations needed for an iterative solver to converge can become very large when the obstacles are densely packed. This high condition number results in a large number of iterations and prevents us from being able to solve very large scale problems where there are a lot of obstacles.

Hence, one of our next projects is to design an efficient preconditioner for the wave boundary integral equation. There has been extensive study on wave propagation in porous media (see for example [14, 105, 24, 5]). We hope to incorporate some of those results in the design of this preconditioner.

A parallel DFMM-based BIE solver for the Maxwell's equation. Although our development has been limited to the scalar Helmholtz equation, our proposed techniques can be applied to other models as well. Thus, one of our planned development is to extent the parallel algorithm in order to solve vector Helmholtz equations. This extension would be particularly useful in electromagnetic applications.[1]

A new multibody GO-BIE method. The current-state GO-BIE is efficient but only for solving scattering by a convex object, for which there is only one geometrical optics reflection. We have devised a more general algorithm which can handle 2-body scattering. Our next step is to extend this algorithm to make it more robust. The extension can be done in a few independent directions.

1. Implementation of the GO-BIE technique in each geometrical optics reflection for optimal performance. The fact that each sub-scattered field has a compact asymptotic representation makes this step feasible.

[1] As the linear scalar wave equation can be transformed into a series of scalar Helmholtz equations in the frequency domain, time-domain Maxwell's equations be transformed into a series of vector Helmholtz equations in the frequency domain.

2. Extension to handle self-reflections by a non-convex object.

3. Extension to handle n bodies, where n can be more than 2. One difficulty associated with a general n-body scattering solver is that for solving u_i^m ($i \in \mathcal{I}$) at iteration, we have to consider a large number of incoming fields that have gone through different paths of reflection. The number of paths increases exponentially with respect to m. In the 2-body case, for each i and m, there is only one incoming field corresponding to one path of reflection.

4. Extension to handle corners and edges. The superposition for $\frac{\partial(u+u^I)}{\partial n}(x)$ on the surface Γ should account for the diffraction terms produced by non-smooth parts of the surface. An analysis for this type of problem can be found in [27].

Appendix

Appendix A

Implementation considerations for DFMM

Here are some optimization techniques that were used in the implementation of the directional fast multipole method.

1. Pre-computations. At each GMRES iteration, the only variable that is updated is the approximated solution to the matrix equation. Therefore, we can pre-compute most of the DFMM operations. These pre-computations include:

   (a) the tree construction and the interaction lists for each node of the tree, which depend only on the problem geometry;

   (b) the encoding of far-to-far, far-to-local, and local-to-local translations; and

   (c) the kernel evaluations for the points that have interactions via the DFMM scheme. Note that since the number of kernel evaluations is very large, pre-computations of the kernel may or may not be practical depending on the memory limit.

2. Since evaluations of the Green's function (complex exponential function in 3D) are relatively costly and in fact dominate the program's runtime, a desirable optimization technique is to use architecture-dependent and highly optimized vector functions provided by chip designers. For example, one can use vectorized exponential functions in Intel's Math Kernel Library or AMD's Core Math Library.

Appendix B

Neumann series in multibody scattering

Our derivation of the Neumann series (4.17) closely follows the description in [36]. The integral equation for μ, equation (4.1), can be recast as

$$(I - R)\mu = f, \tag{B.1}$$

where I denotes the identity operator, $f = \{f_i\}_{i\in\mathcal{I}}$ denotes the boundary data for each $i \in \mathcal{I}$ and $x \in \Gamma_i$ as

$$f_i(x) = 2\left(\frac{\partial u_i^I(x)}{\partial n(x)} - i\gamma u_i^I(x)\right),$$

and the operator $R = \{R_{ij}\}_{i,j\in\mathcal{I}}$ is defined for each (i, j) as

$$(R_{ij}\mu_j)(x) = 2\int_{\Gamma_j}\left(\frac{\partial G(x,y)}{\partial n(x)} - i\gamma G(x,y)\right)\mu_j(y)\,ds(y), \quad x \in \Gamma_i.$$

Inverting the diagonal part of (B.1) yields the equivalent relation

$$(I - T)\mu = g, \tag{B.2}$$

with

$$g_i = (I - R_{ii})^{-1} f_i,$$

and

$$T_{ij} = \begin{cases} (I - R_{ii})^{-1} R_{ij} & \text{if } i \neq j \\ 0 & \text{otherwise.} \end{cases}$$

If $||T|| < 1$ then μ admits of a Neumann series form

$$\mu = \sum_{m=0}^{\infty} \mu^m = \sum_{m=0}^{\infty} T^m g.$$

This expansion of μ explains the mechanism for solving multiple reflections in an iterative way since each term μ^m, defined as $T^m g$, equals $T\mu^{m-1}$ and corresponds exactly to the contribution from the wave that has gone through m geometrical reflections.

Refer to to the paper [36] for a deeper analysis of the operator T.

Nomenclature

Acronyms

BIE	Boundary Integral Equation
DFMM	Directional Fast Multipole Method
FFT	Fast Fourier Transform
FLOPs	Floating Point Operations
GO	Geometrical Optics
GO-BIE	Geometrical Optics based Boundary Integral Equation
GPU	Graphics Processing Unit
HF-FMM	High Frequency Fast Multipole Method (developed by Rokhlin et al.)
MPI	Message Passing Interface: a standardized and portable message-passing system for distributed computing
PDE	Partial Differential Equation

Mathematical Symbols

γ	Coupling parameter used in the combined layer boundary integral equation to make the system well-conditioned , see equation (1.11)
$\Gamma = \partial\Omega$	Boundary of the scatterer
$\lambda = \frac{2\pi}{k}$	Wavelength
Ω	Space occupied by the impenetrable scatterer
Ω^E	exterior domain ($\Omega^E = \mathbb{R}^d \setminus \bar{(}\Omega)$
$\{f_q^{B,l}\}$	Directional outgoing equivalent sources of box B in direction l , page 42
$\{u_q^{B,l}\}$	Directional incoming check potentials of box B in direction l , page 42
$G(x, y)$	Green's function , see equation (1.2)
k	Wave frequency
$K(x, y)$	Integral kernel in a BIE method
$W^{B,l}$	Directional wedge , page 40

Bibliography

[1] T. Abboud, J.-C. Nédélec, and B. Zhou, *Méthode des équations intégrales pour les hautes fréquences*, C. R. Acad. Sci. Paris Sér. I Math., 318 (1994), pp. 165–170. 4.1

[2] B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin, *Wavelet-like bases for the fast solutions of second-kind integral equations*, SIAM J. Sci. Comput., 14 (1993), pp. 159–184. 1.2.2

[3] S. Amini, *On the choice of the coupling parameter in boundary integral formulations of the exterior acoustic problem*, Applicable Analysis, 35 (1990), pp. 75–92. 1.2.2

[4] C. R. Anderson, *An implementation of the fast multipole method without multipoles*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 923–947. 2.1.4

[5] G. Bal and K. Ren, *Transport-based imaging in random media*, SIAM Journal of Applied Mathematics, 68 (2008), pp. 1738–1762. 6.2

[6] S. Balay, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. C. Mcinnes, B. Smith, and H. Zhang, *PETSc Users Manual*. 3.3

[7] M. Bebendorf, *Approximation of boundary element matrices*, Numer. Math., 86 (2000), pp. 565–589. 2.2.2

[8] M. Bebendorf and S. Rjasanow, *Adaptive low-rank approximation of collocation matrices*, Computing, 70 (2003), pp. 1–24. 2.2.2

[9] J.-D. Benamou, *Big ray tracing: Multivalued travel time field computation using viscosity solutions of the eikonal equation*, J. Comput. Phys., 128 (1996), pp. 463–474. 1.2.3

[10] J.-D. Benamou, F. Collino, and O. Runborg, *Numerical microlocal analysis of harmonic wavefields*, J. Comput. Phys., 199 (2004), pp. 717–741. 4.6.1

[11] J.-P. Berenger, *A perfectly matched layer for the absorption of electromagnetic waves*, J. Comput. Phys., 114 (1994), pp. 185–200. 1, 1a

[12] T. Betcke, S. N. Chandler-Wilde, I. G. Graham, S. Langdon, and M. Lindner, *Condition number estimates for combined potential integral operators in acoustics and their boundary element discretisation*, Numerical Methods for Partial Differential Equations, 27 (2011), p. 24. 1.2.2, 3.3.2

[13] G. Beylkin, R. Coifman, and V. Rokhlin, *Fast wavelet transforms and numerical algorithms. I*, Comm. Pure Appl. Math., 44 (1991), pp. 141–183. 1.2.2

[14] M. A. Biot, *Mechanics of deformation and acoustic propagation in porous media*, Journal of Applied Physics, 33 (1962), pp. 1482 – 1498. 6.2

[15] N. Bleistein and R. A. Handelsman, *Asymptotic Expansions of Integrals*, Dover, 1986. 4.3

[16] B. Bradie, R. Coifman, and A. Grossmann, *Fast numerical computations of oscillatory integrals related to acoustic scattering. I*, Appl. Comput. Harmon. Anal., 1 (1993), pp. 94–99. 1.2.2

[17] A. Brandt and A. A. Lubrecht, *Multilevel matrix multiplication and fast solution of integral equations*, J. Comput. Phys., 90 (1990), pp. 348–370. 1.2.2

[18] O. P. Bruno, C. A. Geuzaine, J. A. Monro, Jr., and F. Reitich, *Prescribed error tolerances within fixed computational times for scattering problems of arbitrarily high frequency: the convex case*, Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 362 (2004), pp. 629–645. 2, 4.1, 4.7, 6.1

[19] O. P. Bruno and L. A. Kunyansky, *High-order algorithm for the solution of surface scattering problems: Basic implementation, tests, and applications*, Journal of Computational Physics, 169 (2001), pp. 80–110. 1.2.2

[20] O. M. Bucci and G. Franceschetti, *On the degrees of freedom of scattered fields*, IEEE Trans. Antennas Propagat., 37 (1989), pp. 918–926. 2.2.2

[21] W. Burnside, C. Yu, and R. Marhefka, *A technique to combine the geometrical theory of diffraction and the moment method*, The Antennas and Propagation Society International Symposium, 11 (1973), pp. 253–256. 5.5

[22] A. J. Burton, *Numerical solution of helmholtz radiation problems*, NPL Report OC5/535, (1976). 1, 1b

[23] F. X. Canning, *Sparse approximation for solving integral equations with oscillatory kernels*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 71–87. 1.2.2

[24] J. M. Carcione, *Wave Fields in Real Media: Wave Propagation in Anisotropic, Anelastic, Porous and Electromagnetic Media*, Elsevier, 2nd ed., 2007. 6.2

[25] J. Carrier, L. Greengard, and V. Rokhlin, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 669–686. 1.2.2

[26] V. Cerveny, I. A. Molotkov, and I. Psencik, *Ray Methods in Seismology*, Univ. Karlova Press, 1977. 1.2.3

[27] S. N. Chandler-Wilde and S. Langdon, *A Galerkin boundary element method for high frequency scattering by convex polygons*, SIAM J. Numer. Anal., 45 (2007), pp. 610–640 (electronic). 4.1, 4.2, 1, 2d, 4

[28] H. Cheng, W. Crutchfield, Z. Gimbutas, L. Greengard, J. Huang, V. Rokhlin, N. Yarvin, and J. Zhao, *Remarks on the implementation of the wideband FMM for the helmholtz equation in two dimensions*, in Inverse problems, multi-scale analysis and effective medium theory, vol. 408 of Contemp. Math., Amer. Math. Soc., Providence, RI, 2006, pp. 99–110. 1.2.2, 2.2.1

[29] H. Cheng, L. Greengard, and V. Rokhlin, *A fast adaptive multipole algorithm in three dimensions*, J. Comput. Phys., 155 (1999), pp. 468–498. 1.2.2

[30] W. Chew, E. Michielssen, J. M. Song, and J. M. Jin, eds., *Fast and efficient algorithms in computational electromagnetics*, Artech House, Inc., Norwood, MA, USA, 2001. 2.2.1

[31] D. Colton and R. Kress, *Inverse acoustic and electromagnetic scattering theory*, vol. 93 of Applied Mathematical Sciences, Springer-Verlag, Berlin, second ed., 1998. 1.2.2, 4

[32] E. Darve, *The fast multipole method: numerical implementation*, J. Comput. Phys., 160 (2000), pp. 195–240. 2.2.1

[33] E. Darve and P. Havé, *A fast multipole method for Maxwell equations stable at all frequencies*, Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 362 (2004), pp. 603–628. 2.2.1

[34] V. Domínguez, I. G. Graham, and V. P. Smyshlyaev, *A hybrid numerical-asymptotic boundary integral method for high-frequency acoustic scattering*, Numer. Math., 106 (2007), pp. 471–510. 4.1

[35] M. G. Duffy, *Quadrature over a pyramid or cube of integrands with a singularity at a vertex*, SIAM Journal on Numerical Analysis, 19 (1982), pp. 1260–1262. 3.2.2

[36] F. Ecevit and F. Reitich, *Analysis of multiple scattering iterations for high-frequency scattering problems. i: the two-dimensional case*, Numer. Math., 114 (2009), pp. 271–354. 4.6.1, B, B

[37] B. Engquist and A. Majda, *Absorbing boundary conditions for the numerical simulation of waves*, Mathematics of Computations, 31 (1977), pp. 629–651. 1, 1a

[38] B. Engquist and O. Runborg, *Computational high frequency wave propagation*, Acta Numerica, 12 (2003), pp. 181–266. 1.2.1

[39] B. Engquist, O. Runborg, and A.-K. Tornberg, *High frequency wave propagation by the segment projection method*, J. Comput. Phys., 178 (2002), pp. 373–390. 2, 1.2.3

[40] B. Engquist, K. Tran, and L. Ying, *Fast hybrid algorithms for high frequency scattering*, in Mathematical Modeling of Wave Phenomena: 3rd Conference on Mathematical Modeling of Wave Phenomena, 20th Nordic Conference on Radio Science and Communications, B. Nilsson, L. Fishman, A. Karlsson, and S. Nordebo, eds., vol. 1106, AIP, 2009, pp. 3–17. 5.1

[41] B. Engquist and L. Ying, *Fast directional multilevel algorithms for oscillatory kernels*, SIAM Journal on Scientific Computing, 29 (2007), pp. 1710–1737. 2.2.1, 2.2.2, 2.2.5

[42] B. Engquist and L. Ying, *Fast directional computation for the high frequency Helmholtz kernel in two dimensions*, tech. report, University of Texas at Austin, 2008. 2.2.1, 2.2.2

[43] E. Fatemi, B. Engquist, and S. J. Osher, *Numerical solution of the high frequency asymptotic expansion for the scalar wave equation*, J. Comput. Phys., 120 (1995), pp. 145–155. 1.2.3

[44] G. T. D. Francia, *Degrees of freedom of an image*, J. Opt. Soc. Am., 59 (1969), pp. 779–804. 2.2.2

[45] M. Frigo and S. G. Johnson, *The design and implementation of FFTW3*, Proceedings of the IEEE, 93 (2005), pp. 216–231. Special issue on "Program Generation, Optimization, and Platform Adaptation". 3.3

[46] K. Fürlinger and D. Skinner, *Capturing and visualizing event flow graphs of mpi applications*, in Proceedings of the 2009 international conference on Parallel processing, Euro-Par'09, Berlin, Heidelberg, 2010, Springer-Verlag, pp. 218–227. 3.3.1

[47] C. A. Geuzaine, Member, IEEE, O. P. Bruno, and F. Reitich, *On the o(1) solution of multiple-scattering problems*, IEEE Transactions on Magnetics, 41 (2005). 4.6.1, 4.6.1

[48] E. Giladi and J. B. Keller, *A hybrid numerical asymptotic method for scattering problems*, J. Comput. Phys., 174 (2001), pp. 226–247. 4.1

[49] A. S. Glassner, *An Introduction to Ray Tracing*, Xerox PARC, Academic Press, 1989. 1.2.3

[50] L. Greengard, *The rapid evaluation of potential fields in particle systems*, ACM Distinguished Dissertations, MIT Press, Cambridge, MA, 1988. 1b, 2.1.1, 3.2.1

[51] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348. 1b, 1.2.2, 2.1.1, 3.2.1

[52] G. W. Griffiths and W. E. Schiesser, *Linear and nonlinear waves*, Scholarpedia, (2009). 1.2.1

[53] I. Harari and T. J. R. Hughes, *A cost comparison of boundary element and finite element methods for problems of time-harmonic acoustics*, Comput. Methods Appl. Mech. Eng., 97 (1992), pp. 77–102. 1.3

[54] B. Hariharan, S. Aluru, and B. Shanker, *A scalable parallel fast multipole method for analysis of scattering from perfect electrically conducting surfaces*, in In Proceedings of Supercomputing, The SCxy Conference series, ACM/IEEE, 2002. 3.1

[55] J. G. Harris, *Linear elastic waves*, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, 2001. 1.2.1

[56] R. Hodges and Y. Rahmat-Samii, *An iterative current-based hybrid method for complex structures*, IEEE Transactions on Antennas and Propagation, 45 (1997), pp. 265–276. 5.5

[57] L. V. Hörmander, *The analysis of linear partial differential operators. I-IV*, Springer-Verlag, Berlin, 1983–1985. 4.4

[58] D. Huybrechs and S. Vandewalle, *A sparse discretization for integral equation formulations of high frequency scattering problems*, SIAM J. Sci. Comput., 29 (2007), pp. 2305–2328 (electronic). 1.2.3, 4.1

[59] Intel, *Intel's Math Kernel Library Reference Manual*. 3.3

[60] U. Jakobus, J. Christ, and F. Landstorfer, *Po-mom analysis of cavity-backed antennas*, The Eighth IEEE International Conference on Antennas and Propagation, 1 (1993), pp. 111–114. 1.2.1, 5.5

[61] S. Jin, H. Liu, S. Osher, and R. Tsai, *Computing multi-valued physical observables for the high frequency limit of symmetric hyperbolic systems*, J. Comput. Phys., 210 (2005), pp. 497–518. 2, 1.2.3

[62] B. R. Julian and D. Gubbins, *Three-dimensional seismic ray tracing*, J. Geophys. Res., 43 (1977), pp. 95–114. 1.2.3

[63] S. Kapur and L. D. E., *IES3: a fast integral equation solver for efficient 3-dimensional extraction*, in ICCAD '97: Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design, Washington, DC, USA, 1997, IEEE Computer Society, pp. 448–455. 2.2.2

[64] S. Kapur and V. Rokhlin, *High-order corrected trapezoidal quadrature rules for singular functions*, SIAM Journal on Numerical Analysis, 34 (1997), pp. 1331–1356. 4

[65] G. Karypis and V. Kumar, *Parallel multilevel k-way partitioning scheme for irregular graphs*, 1996. 3.2.1

[66] ——, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM Journal on Scientific Computing, 20 (1998), pp. 359–392. 3.2.1

[67] J. B. Keller, *Geometrical theory of diffraction*, J. Opt. Soc. Amer., 52 (1962), pp. 116–130. 2, 1.2.3, 1.2.3, 4.2

[68] R. E. Kleinmann and G. F. Roach, *Boundary integral equations for the three-dimensional helmholtz equation*, SIAM Review, 16 (1974), pp. 214–236. 1, 1b

[69] M. Kline and I. W. Kay, *Electromagnetic theory and geometrical optics*, Pure and applied mathematics, Interscience publishers, 1965. 2, 1.2.3

[70] R. Kress, *Minimizing the condition number of boundary integral operators in acoustic and electromagnetic scattering*, Inst. für Numerische und Angewandte Mathematik, Univ., 1983. 1.2.2

[71] R. Kress, *Linear Integral Equations*, Springer, NY, 1989. 1.2.2

[72] J. Kurzak and B. M. Pettitt, *Massively parallel implementation of a fast multipole method for distributed memory machines*, J. Parallel Distrib. Comput., 65 (2005), pp. 870–881. 3.1

[73] R. T. Langan, I. Lerche, and R. T. Cutler, *Tracing of rays through heterogeneous media: An accurate and efficient procedure*, Geophysics, 50 (1985), pp. 1456–1465. 1.2.3

[74] I. Lashuk, A. Chandramowlishwaran, H. Langston, T.-A. Nguyen, R. Sampath, A. Shringarpure, R. Vuduc, L. Ying, D. Zorin, and G. Biros, *A massively parallel adaptive fast-multipole method on heterogeneous architectures*, in Proceedings of the Conference

on High Performance Computing Networking, Storage and Analysis, SC '09, New York, NY, USA, 2009, ACM, pp. 58:1–58:12. 3.1, 2

[75] Y. Li and H. Ling, *Investigation of wave propagation in a dielectric rod array: Toward the understanding of hf/vhf propagation in a forest*, Antennas and Propagation, IEEE Transactions on, 58 (2010), pp. 4025 –4032. 3.1

[76] Y. Li, M.-F. Wu, A. Yilmaz, and H. Ling, *Investigation of short-range radiowave propagation at hf/vhf frequencies in a forested environment*, Antennas and Wireless Propagation Letters, IEEE, 8 (2009), pp. 1182 –1185. (document), 3.1, 3.1

[77] E. Lindman, *Free-space boundary conditions for the time dependent wave equation*, J. Comput. Phys., 18 (1975), pp. 66–78. 1a

[78] H. Liu, S. Osher, and R. Tsai, *Multi-valued solution and level set methods in computational high frequency wave propagation*, Comm. Comp. Phys., 1 (2006), pp. 765–804. 2, 1.2.3

[79] S. Lloyd, *Least squares quantization in pcm*, Information Theory, IEEE Transactions on, 28 (1982), pp. 129–137. 3.2.1, 3.2.2

[80] P.-G. Martinsson, V. Rokhlin, and M. Tygert, *A randomized algorithm for the approximation of matrices*, tech. report, Yale University, 2006. 2.2.2

[81] L. N. Medgyesi-Mitschang and D.-S. Wang, *Hybrid methods for analysis of complex scatterers*, P. IEEE, 77 (1989), pp. 770–779. 1.2.1

[82] R. B. Melrose and M. E. Taylor, *Near peak scattering and the corrected Kirchhoff approximation for a convex obstacle*, Adv. in Math., 55 (1985), pp. 242–315. 2, 1.2.3, 1.2.3, 1.2.3, 4.2, 4.2, 3

[83] E. Michielssen and A. Boag, *A multilevel matrix decomposition algorithm for analyzing scattering from large structures*, IEEE Transactions on Antennas and Propagation, 44 (1996), pp. 1086–1093. 2.2.2

[84] G. Mur, *Absorbing boundary conditions for the finite difference approximation of the time-domain electromagnetic field equations*, Transactions on Electromagnetic Compatibility, (1981), pp. 377–382. 1a

[85] S. Nayar, K.Ikeuchi, and T. Kanade, *Surface Reflection: Physical and Geometrical Perspectives*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (1991), pp. 611–634. 2, 1.2.3

[86] M. Nieto-Vesperinas, *Scattering and Diffraction in Physical Optics*, Wiley, New York, 1991. 1.2.3

[87] N. Nishimura, *Fast multipole accelerated boundary integral equation methods*, Applied Mechanics Reviews, 55 (2002), pp. 299–324. 2.2.1

[88] M. O'Neil and V. Rokhlin, *A new class of analysis-based fast transforms*, tech. report, Yale University. YALE/DCS/TR1384, 2007. 1.2.2

[89] S. Osher, L.-T. Cheng, M. Kang, H. Shim, and Y.-H. Tsai, *Geometric optics in a phase-space-based level set and eulerian framework*, Journal of Computational Physics, 179 (2002), pp. 622–648. 2, 1.2.3

[90] S. J. Osher and J. A. Sethian., *Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations*, J. Comput. Phys., 79 (1988), pp. 12–49. 2, 1.2.3

[91] J. J. Ottusch, M. A. Stalzer, J. L. Visher, and S. M. Wandzura, *Scalable electromagnetic scattering calculations on the sgi origin 2000*, in Proceedings of the 1999 ACM/IEEE conference on Supercomputing (CDROM), Supercomputing '99, New York, NY, USA, 1999, ACM. 3.1

[92] A. Peterson, S. Ray, and R. Mittra, *Computational Methods for Electromagnetics*, University Press, 2001. 1c

[93] V. Rokhlin, *Rapid solution of integral equations of scattering theory in two dimensions*, J. Comput. Phys., 86 (1990), pp. 414–439. 1.2.2, 2.2.1

[94] V. Rokhlin, *Diagonal forms of translation operators for the Helmholtz equation in three dimensions*, Appl. Comput. Harmon. Anal., 1 (1993), pp. 82–93. 2.2.1

[95] Y. Saad, *Iterative methods for sparse linear systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, second ed., 2003. 2.1.1

[96] F. E. Sevilgen and S. Aluru, *A unifying data structure for hierarchical methods*, in Proceedings of the 1999 ACM/IEEE conference on Supercomputing (CDROM), Supercomputing '99, New York, NY, USA, 1999, ACM. 2

[97] J. P. Singh, C. Holt, J. L. Hennessy, and A. Gupta, *A parallel adaptive fast multipole method*, in In Proceedings of Supercomputing 93, 1993, pp. 54–65. 3.1

[98] E. M. Stein, *Harmonic analysis: real-variable methods, orthogonality, and oscillatory integrals*, vol. 43 of Princeton Mathematical Series, Princeton University Press, Princeton, NJ, 1993. With the assistance of Timothy S. Murphy, Monographs in Harmonic Analysis, III. 4.3

[99] TACC, *Lonestar user guide*. 3.3

[100] T. Tamir, *On radio-wave propagation in forest environments*, IEEE Trans. Antennas Propagat, (1967), pp. 806–817. 3.1

[101] N. Tanushev, B. Engquist, and R. Tsai, *Gaussian beam decomposition of high frequency wave fields*, J. Comput. Phys., 228 (2009), pp. 8856–8871. 2, 1.2.3

[102] S.-H. Teng, *Provably good partitioning and load balancing algorithms for parallel adaptive n-body simulation*, SIAM J. Sci. Comput., 19 (1998), pp. 635–656. 3.2.1

[103] G. A. Thiele and T. H. Newhouse, *A hybrid technique for combining moment methods with a geometrical theory of diffraction*, IEEE T. Antenn. Propag., 23 (1975), pp. 62–69. 1.2.1

[104] P. Tsuji and L. Ying, *A fast directional algorithm for high-frequency electromagnetic scattering*, J. Comput. Phys., 230 (2011), pp. 5471–5487. 3.1

[105] J. Virieux, *Sh-wave propagation in heterogeneous media: Velocity-stress finite-difference method*, Geophysics, 49 (1984), pp. 1933–1942. 6.2

[106] M. S. Warren and J. K. Salmon, *A parallel hashed oct-tree n-body algorithm*, in Proceedings of the 1993 ACM/IEEE conference on Supercomputing, Supercomputing '93, New York, NY, USA, 1993, ACM, pp. 12–21. 3.1

[107] J. J. Wen and M. A. Breazeale, *A diffraction beam field expressed as the superposition of gaussian beams*, The Journal of the Acoustical Society of America, 83 (1988), pp. 1752–1756. 2, 1.2.3

[108] K. S. Yee, *Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media*, Transactions on Antennas and Propagation, AP-14 (1966), pp. 302–307. 1

[109] L. Ying, *Fast algorithms for boundary integral equations*, in Multiscale Methods in Science and Engineering, B. Engquist, O. Runborg, and

P. Lötstedt, eds., vol. 66 of Lecture Notes in Computational Science and Engineering, Springer Verlag, 2005, pp. 139–194. (document), 1.2.2, 2.1.1, 2.2, 2.3, 2.4

[110] ——, *A kernel independent fast multipole algorithm for radial basis functions*, J. Comput. Phys., 213 (2006), pp. 451–457. 2.1.4

[111] L. Ying, G. Biros, and D. Zorin, *A kernel-independent adaptive fast multipole algorithm in two and three dimensions*, J. Comput. Phys., 196 (2004), pp. 591–626. 2.1.4

[112] L. Ying, G. Biros, D. Zorin, and H. Langston, *A new parallel kernel-independent fast multipole method*, in SC '03: Proceedings of the 2003 ACM/IEEE conference on Supercomputing, Washington, DC, USA, 2003, IEEE Computer Society, p. 14. 3.1, 2