Fast Algorithms for Boundary Integral Equations

Lexing Ying

Department of Mathematics, University of Texas, Austin, TX 78712, USA,
lexing@math.utexas.edu

Summary. This article reviews several fast algorithms for boundary integral equations. After a brief introduction of the boundary integral equations for the Laplace and Helmholtz equations, we discuss in order the fast multipole method and its kernel independent variant, the hierarchical matrix framework, the wavelet based method, the high frequency fast multipole method, and the recently proposed multidirectional algorithm.

Keywords. Boundary integral equations, Laplace equation, Helmholtz equation, fast algorithms, fast multipole method, hierarchical matrices, wavelets, multiscale methods, multidirectional methods.
AMS subject classifications. 45A05, 65R20.

1 Outline

Many physical problems can be formulated as partial differential equations (PDEs) on certain geometric domains. For some of them, the PDEs can be reformulated using the so-called boundary integral equations (BIEs). These are integral equations which only involve quantities on the domain boundary. Some advantages of working with the BIEs are automatic treatments of boundary condition at infinity, better condition numbers, and fewer numbers of unknowns in the numerical solution. On the other hand, one of the major difficulties of the BIEs is that the resulting linear systems are dense, which is in direct contrast to the sparse systems of the PDEs. For large scale problems, direct solution of these dense linear systems becomes extremely time-consuming. Hence, how to solve these dense linear systems efficiently has become one of the central questions. Many methods have been developed in the last twenty years to address this question. In this article, we review some of these results.

We start in Sect. 2 with a brief introduction of the boundary integral formulation with the Laplace and Helmholtz equations as our examples. A major difference between these two equations is that the kernel of the Laplace equation is non-oscillatory while the one of the Helmholtz equation is oscillatory. For the non-oscillatory kernels, we discuss the fast multipole method (FMM) in Sect. 3 and its kernel indepen-

dent variant in Sect. 4, the hierarchical matrices frame in Sect. 5, and the wavelet based methods in Sect. 6. For the oscillatory kernels, we review the high frequency fast multipole method (HF-FMM) in Sect. 7 and the recently developed multidirectional method in Sect. 8.

The purpose of this article is to provide an introduction to these methods for advanced undergraduate and graduate students. Therefore, our discussion mainly focuses on algorithmic ideas rather than theoretical estimates. For the same reason, we mostly refer only to the original papers of these methods and keep the size of the reference list to a minimum. Many important results are not discussed here due to various limitations and we apologize for that.

2 Boundary Integral Formulation

Many linear partial differential equation problems have boundary integral equation formulations. In this section, we focus on two of the most important examples and demonstrate how to transform the PDE formulations into the BIE formulations. Our discussion mostly follows the presentation in [11, 18, 20]. We denote $\sqrt{-1}$ with i and assume that all functions are sufficiently smooth.

2.1 Laplace equation

Let D be a bounded domain with smooth boundary in $\mathbb{R}^d (d = 2,3)$. n is the exterior normal to D. The Laplace equation on D with Dirichlet boundary condition is

$$-\Delta u = 0 \quad \text{in } D \tag{1}$$

$$u = f \quad \text{on } \partial D \tag{2}$$

where f is defined on ∂D. The geometry of the problem is shown in Fig. 1. We seek to represent $u(x)$ for $x \in D$ in an integral form which uses only quantities on the boundary ∂D.

The Green's function for the Laplace equation is

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

Some of the important properties of $G(x,y)$ are

- $G(x,y)$ is symmetric in x and y,
- $G(x,y)$ is non-oscillatory, and
- $-\Delta_x G(x,y) = \delta_y(x)$ and $-\Delta_y G(x,y) = \delta_x(y)$

where Δ_x and Δ_y take the derivatives with respect x and y, respectively, and δ_x is the Dirac function located at x. The following theorem is a simple consequence of Stokes' theorem.

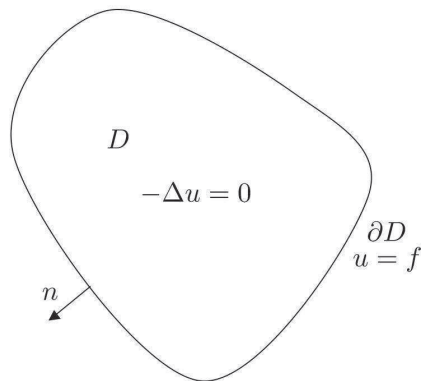

Fig. 1. Domain of the Dirichlet boundary value problem of the Laplace equation.

Theorem 1. *Let u and v to be two sufficiently smooth functions on $\bar{D}$. Then*

$$\int_D (u\Delta v - v\Delta u)dx = \int_{\partial D} \left(u\frac{\partial v(y)}{\partial n(y)} - v\frac{\partial u(y)}{\partial n(y)} \right) ds(y).$$

A simple application of the previous theorem gives the following result.

Theorem 2. *Let u be a sufficiently smooth function on $\bar{D}$ such that $-\Delta u = 0$ in D. For any x in D,*

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

Proof. Pick a small ball B at x that is contained in D (see Fig. 2). From the last theorem, we have

$$\int_{D\setminus B} (u(y)\Delta G(x,y) - G(x,y)\Delta u(y))ds(y) = \int_{\partial(D\setminus B)} \left(u(y)\frac{\partial G(x,y)}{\partial n(y)} - G(x,y)\frac{\partial u(y)}{\partial n(y)} \right) ds(y).$$

Since $-\Delta u(y) = 0$ and $-\Delta G(x,y) = 0$ for $y \in D \setminus B$, the left hand side is equal to zero. Therefore,

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

where n points towards x on ∂B. Now let the radius of the ball B go to zero. The first term of the right hand side goes to $-u(x)$ while the second term approaches 0.

From the last theorem, we see that $u(x)$ for x in D can be represented as a sum of two boundary integrals. In the boundary integral formulation, we seek to represent $u(x)$ using only one of them. This degree of freedom gives rise to the following two approaches.

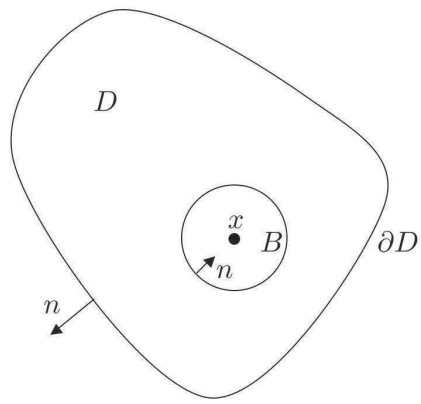

Fig. 2. Proof of Theorem 2.

Method 1

We represent $u(x)$ for $x \in D$ using the integral that contains $G(x,y)$

$$u(x) = \int_{\partial D} \varphi(y) G(x,y) ds(y) \tag{4}$$

where φ is an unknown density on ∂D. This formulation is called the *single layer form* and φ is often called the *single layer density*. One can show that any sufficiently nice $u(x)$ can be represented using the single layer form (see [20] for details). Letting x approach $z \in \partial D$, we get

$$f(z) = u(z) = \int_{\partial D} \varphi(y) G(z,y) ds(y),$$

which is an integral equation that involves only boundary quantities φ and f. Therefore, the steps to solve the Laplace equation using the single layer form are:

1. Find $\varphi(z)$ on ∂D such that

$$f(z) = \int_{\partial D} \varphi(y) G(z,y) ds(y). \tag{5}$$

   This equation is a *Fredholm equation of the first kind* (see [20]).
2. For x in D, compute $u(x)$ by

$$u(x) = \int_{\partial D} \varphi(y) G(x,y) ds(y). \tag{6}$$

Method 2

We can also represent $u(x)$ for $x \in D$ using the integral that contains $\frac{\partial G(x,y)}{\partial n(y)}$

$$u(x) = -\int_{\partial D} \varphi(y) \frac{\partial G(x,y)}{\partial n(y)} ds(y) \tag{7}$$

where φ is again an unknown density on ∂D. This formulation is called the *double layer form* and φ is the *double layer density*. In fact, the double layer form is capable of representing any sufficiently nice $u(x)$ in D [20]. If we now let x approach $z \in \partial D$, we obtain the following equation on the boundary:

$$f(z) = u(z) = \frac{1}{2}\varphi(z) - \int_{\partial D} \frac{\partial G(z,y)}{\partial n(y)} \varphi(y) ds(y).$$

The extra $\frac{1}{2}\varphi(z)$ term comes up because the integral (7) is not uniformly integrable near $z \in \partial D$. Hence, one cannot simply exchange the limit and integral signs. Since the boundary ∂D is smooth, the integral operator with the kernel $\frac{\partial G(z,y)}{\partial n(y)}$ is a compact operator. The steps to solve the Laplace equation using the double layer form are:

1. Find $\varphi(z)$ on ∂D such that

$$f(z) = \frac{1}{2}\varphi(z) - \int_{\partial D} \frac{\partial G(z,y)}{\partial n(y)} \varphi(y) ds(y). \tag{8}$$

   This equation is a *Fredholm equation of the second kind*.
2. For x in D, compute $u(x)$ with

$$u(x) = -\int_{\partial D} \frac{\partial G(x,y)}{\partial n(y)} \varphi(y) ds(y). \tag{9}$$

Between these two approaches, we often prefer to work with the double layer form (Method 2). The main reason is that the Fredholm equation of the second kind has a much better condition number, thus dramatically reducing the number of iterations required in a typical iterative solver.

2.2 Helmholtz equation

We now turn to the Helmholtz equation. Let D be a bounded domain with smooth boundary in $\mathbb{R}^d (d = 2,3)$ and n be the exterior normal to D. The unbounded Helmholtz equation on $\mathbb{R}^d \setminus \bar{D}$ $(d = 2,3)$ with Dirichlet boundary condition describes the scattering field of a sound soft object:

$$-\Delta u - k^2 u = 0 \quad \text{in } \mathbb{R}^d \setminus \bar{D} \tag{10}$$

$$u(x) = -u^{inc}(x) \quad \text{for } x \in \partial D \tag{11}$$

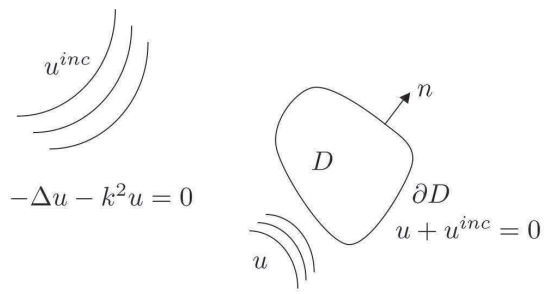

Fig. 3. Domain of the Dirichlet boundary value problem of the Helmholtz equation.

$$\lim_{r\to\infty} r^{\frac{d-1}{2}}\left(\frac{\partial u}{\partial r} - \mathrm{i}ku\right) = 0 \tag{12}$$

where k is the wave number, u^{inc} is the incoming field and u is the scattering field. The last equation is called the *Sommerfeld radiation condition* which guarantees that the scattering field propagates to infinity. The geometry of the problem is described in Fig. 3. Our goal is again to represent $u(x)$ for $x \in \mathbb{R}^d \setminus \bar{D}$ in an integral form which uses quantities defined on the boundary ∂D.

The Green's function of the Helmholtz equation is

$$G(x,y) = \begin{cases} \frac{\mathrm{i}}{4} H_0^1(k|x-y|) & (d=2) \\ \frac{1}{4\pi}\frac{\exp(\mathrm{i}k|x-y|)}{|x-y|} & (d=3) \end{cases} \tag{13}$$

Some of the important properties of $G(x,y)$ are

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

Theorem 3. *Let C be a bounded domain with smooth boundary. Suppose that u is sufficiently smooth in $\bar{C}$ and satisfies $(-\Delta - k^2)u = 0$ in C. Then for any x in C*

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

Proof. Pick a small ball B centered at x. Then we have

$$\int_{C\setminus B}(u(y)\Delta G(x,y) - G(x,y)\Delta u(y))dy = \int_{\partial(C\setminus B)}\left(u(y)\frac{\partial G(x,y)}{\partial n(y)} - G(x,y)\frac{\partial u(y)}{\partial n(y)}\right)ds(y).$$

The left hand side is equal to

$$\int_{C\setminus B}(u\cdot(\Delta G + k^2 G) - G(\Delta u + k^2 u))dy = 0.$$

The rest of the proof is the same as the one of Theorem 2.

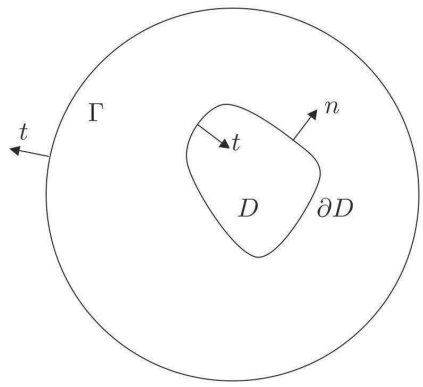

Fig. 4. Proof of Theorem 4.

The above theorem addresses a bounded domain C. However, what we are really interested in is the unbounded domain $\mathbb{R}^d \setminus \bar{D}$.

Theorem 4. *Suppose that u is sufficiently smooth and satisfies $(-\Delta - k^2)u = 0$ in $\mathbb{R}^d \setminus D$. Then for any x in $\mathbb{R}^d \setminus D$,*

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

Proof. Pick a large ball Γ that contains D. Consider the domain $\Gamma \setminus \bar{D}$ (see Fig. 4). Let t be the exterior normal direction of $\Gamma \setminus \bar{D}$. From the previous theorem, we have

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

Using the Sommerfeld condition at infinity, one can show that the integral over $\partial \Gamma$ goes to zero as one pushes the radius of Γ to infinity [11]. Noting that $t = -n$ on ∂D, we have

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

From the last theorem, we see that $u(x)$ for x in $\mathbb{R}^d \setminus \bar{D}$ can be represented as a sum of two integrals. In the boundary integral formulation of the Helmholtz equation, one option is to represent $u(x)$ by the *double layer form*:

$$u(x) = \int_{\partial D} \frac{\partial G(x,y)}{\partial n(y)} \varphi(y) ds(y)$$

Different from the double layer form of the Laplace equation, the double layer form of the Helmholtz equation is not capable of representing arbitrary field $u(x)$ for $x \in \mathbb{R}^d \setminus \bar{D}$. If k is one of the internal resonant numbers such that the internal Neumann problem with zero boundary condition has non-trivial solution, then this double layer form is singular (see [11]). In practice, we use

$$u(x) = \int_{\partial D} \left(\frac{\partial G(x,y)}{\partial n(y)} - \mathrm{i}\eta G(x,y) \right) \varphi(y) ds(y).$$

where η is a real number (for example, $\eta = k$). As we let x approach z on ∂D, we get

$$-u^{inc}(z) = u(z) = \frac{1}{2}\varphi(z) + \int_{\partial D} \left(\frac{\partial G(z,y)}{\partial n(y)} - \mathrm{i}\eta G(z,y) \right) \varphi(y) ds(y)$$

where the extra term $\frac{1}{2}\varphi(z)$ is due to the fact that the integral is improper at $z \in \partial D$.

The steps to solve the Helmholtz equation using this double layer form are:

1. Find a function $\varphi(z)$ on ∂D such that

$$-u^{inc}(z) = \frac{1}{2}\varphi(z) + \int_{\partial D} \left(\frac{\partial G(z,y)}{\partial n(y)} - \mathrm{i}\eta G(z,y) \right) \varphi(y) ds(y). \tag{14}$$

2. For point x in $\mathbb{R}^3 \setminus D$, compute $u(x)$ with

$$u(x) = \int_{\partial D} \left(\frac{\partial G(x,y)}{\partial n(y)} - \mathrm{i}\eta G(x,y) \right) \varphi(y) ds(y). \tag{15}$$

We have seen the derivations of the BIEs for the interior Laplace Dirichlet boundary value problem and the exterior Helmholtz Dirichlet boundary value problem. Though both cases use the Green's functions of the underlying equation and the Stokes' theorem, the derivation for the Helmholtz equation is complicated by the existence of the internal resonant numbers. For other elliptic boundary value problems, the derivations of the BIE formulations often differ from case to case.

2.3 Discretization

In both BIEs discussed so far, we need to solve a problem of the following form: find $\varphi(x)$ on ∂D such that

$$f(x) = \varphi(x) + \int_{\partial D} K(x,y)\varphi(y) ds(y), \quad i.e., \quad f = (I+K)\varphi$$

or

$$f(x) = \int_{\partial D} K(x,y)\varphi(y) ds(y), \quad i.e., \quad f = K\varphi.$$

where $K(x,y)$ is either the Green's function or its derivative of the underlying PDE. In order to solve these equations numerically, we often use one of the following three discretization methods: the Nyström method, the collocation method, and the Galerkin method. Let us discuss these methods briefly using the Fredholm equation of the second kind.

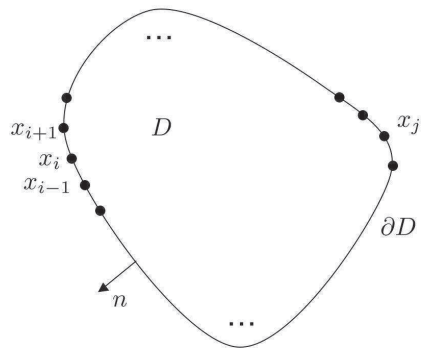

Fig. 5. Nyström method

Nyström method

The idea of the Nyström method is to approximate integral operators with quadrature operators. The steps are:

1. Approximate the integral operator $(K\varphi)(x) := \int K(x,y)\varphi(y)dy$ with the quadrature operator
$$(K_N\varphi)(x) := \sum_{j=1}^{N} K(x,x_j)\lambda_j\varphi(x_j)$$
where $\{x_j\}$ are the quadrature points and $\{\lambda_j\}$ are the quadrature weights (see Fig. 5). Here we make the assumption that $\{\lambda_j\}$ are independent of x. In practice, $\{\lambda_j\}$ often depend on x when x_j is in the neighborhood of x if the kernel $K(x,y)$ has a singularity at $x = y$.
2. Find $\varphi(x)$ such that $\varphi + K_N\varphi = f$. We write down the equation at $\{x_i\}$:
$$\varphi_i + \sum_{j=1}^{n} K(x_i,x_j)\lambda_j\varphi_j = f_i, \quad i = 1,\cdots,N \tag{16}$$
and solve for $\{\varphi_i\}$. Here $f_i = f(x_i)$.
3. The value of $\varphi(x)$ at $x \in \partial D$ is computed using
$$\varphi(x) = f(x) - \sum_{j=1}^{n} K(x,x_j)\lambda_j\varphi_j. \tag{17}$$

Collocation method

The idea of the collocation method is to use subspace approximation. The steps are:

1. Approximate $\varphi(x)$ by $\sum_{j=1}^{N} c_j\varphi_j(x)$ where $\{\varphi_j(x)\}$ are basis functions on ∂D. Let $\{x_j\}$ be a set of points on ∂D (see Fig. 6), where x_j is often the center of $supp(\varphi_j)$.

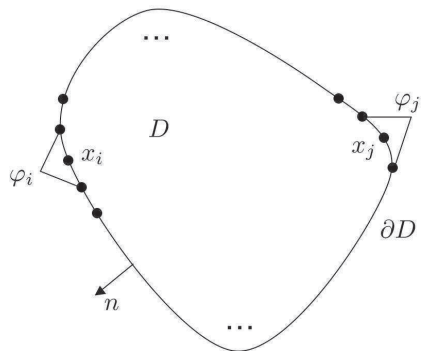

Fig. 6. Collocation and Galerkin methods

2. Find $\{c_j\}$ such that $\varphi + K\varphi = f$ is satisfied at $\{x_j\}$, i.e.,

$$\sum_{j=1}^{N} c_j \varphi_j(x_i) + (K(\sum_{j=1}^{N} c_j \varphi_j))(x_i) = f(x_i), \quad i = 1, \cdots, N \tag{18}$$

Galerkin method

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

1. Approximate $\varphi(x)$ by $\sum_{j=1}^{N} c_j \varphi_j(x)$ where $\{\varphi_j(x)\}$ are often localized basis functions on ∂D.
2. Find $\{c_j\}$ such that $\varphi + K\varphi - f$ is orthogonal to all the subspace generated by $\varphi\ (x\}$.

$$\langle \varphi_i, \sum_{j=1}^{N} c_j \varphi_j + K(\sum_{j=1}^{N} c_j \varphi_j) - f \rangle = 0, \quad i = 1, \cdots, N \tag{19}$$

2.4 Iterative solution

The following discussion is in the setting of the Nyström method. The situations for the other methods are similar. In the matrix form, the linear system that one needs to solve is

$$(\boldsymbol{I} + \boldsymbol{K}\boldsymbol{\Lambda})\boldsymbol{\varphi} = \boldsymbol{f}$$

where $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{K}$ is the matrix with entries $K(x_i, x_j)$, $\boldsymbol{\Lambda}$ is the diagonal matrix with the diagonal entries equal to $\{\lambda_j\}$, $\boldsymbol{\varphi}$ is the vector of $\{\varphi_j\}$, and $\boldsymbol{f}$ is the vector of $\{f_j\}$.

Since $\boldsymbol{K}$ is a dense matrix, the direct solution of this equation takes $O(N^3)$ steps. For large N, this becomes extremely time-consuming and solving this system directly is not feasible. Therefore, we need to resort to iterative solvers.

Since the integral operator K is compact, its eigenvalues decay to zero. This is also true for the discretized version, the matrix $\boldsymbol{K}$. Therefore, the condition number of $\boldsymbol{I}+\boldsymbol{K}\boldsymbol{\Lambda}$ is small and independent of the number of quadrature points N. As a result, the number of iterations is also independent of N. In each iteration, one computes $\boldsymbol{K}\boldsymbol{\psi}$ for a given vector $\boldsymbol{\psi}$. Since $\boldsymbol{K}$ is dense, a naive implementation of this matrix-vector multiplication takes $O(N^2)$ steps, which can be still quite expensive for large values of N. How to compute the product $\boldsymbol{K}\boldsymbol{\psi}$ is the question that we will address in the following sections.

Before we move on, let us compare the PDE and BIE formulations. For the PDE formulations, a numerical solution often requires $O((1/h)^d)$ unknowns for a given discretization size h. Special care is necessary for unbounded exterior problems. Since the resulting linear system is sparse, each iteration of the iterative solver is quite fast though the number of iterations might be large. Finally, the PDE formulations work for domains with arbitrary geometry and problems with variable coefficients.

For the BIE formulations, a numerical solution involves only $O((1/h)^{d-1})$ unknowns on the domain boundary for a given discretization size h. No special care is needed for exterior domains. The resulting system is always dense, so fast algorithms are necessary for efficient iterative solutions of the BIE formulations. As we have seen already, the Green's functions are fundamental in deriving the integral equations. Since the Green's functions are often unknown for problems with variable coefficients, most applications of the BIE formulations are for problems with constant coefficient.

3 Fast Multipole Method

In each step of the iterative solution of a BIE formulation, we face the following problem. Given a set of charges $\{f_i, 1 \le i \le N\}$ located at points $\{p_i, 1 \le i \le N\}$ (see Fig. 7) and the Green's function $G(x,y)$ of the underlying equation, we want to compute at each p_i the potential

$$u_i = \sum_{j=1}^{N} G(p_i, p_j) f_j. \tag{20}$$

As we pointed earlier, a naive algorithm takes $O(N^2)$ steps, which can be quite expensive for large values of N. In this section, we introduce the fast multipole method by Greengard and Rokhlin [15, 16] for the Green's function of the Laplace equation. This remarkable algorithm reduces the complexity from $O(N^2)$ to $O(N)$ for any fixed accuracy ε.

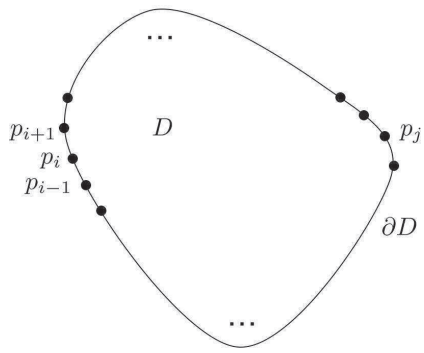

Fig. 7. Distribution of quadrature points $\{p_i\}$ on the boundary of the domain D.

3.1 Geometric part

Two sets A and B are said to be *well-separated* if the distance between A and B are greater than their diameters. Let us consider the interaction from a set of points $\{y_j\}$ in B to a set of points $\{x_i\}$ in A, where both $\{y_j\}$ and $\{x_i\}$ are subsets of $\{p_i\}$. The geometry is shown in Fig. 8.

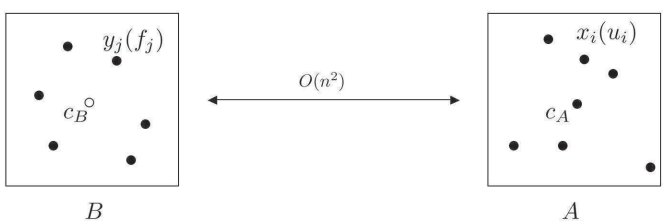

Fig. 8. Two boxes A and B are well-separated. Direct computation takes $O(N^2)$ steps.

Suppose that $\{f_j\}$ are the charges at $\{y_j\}$. Let us consider the following approximation for the potential u_i at each x_i

$$u_i \approx u(c_A) = \sum_j G(c_A, y_j) f_j \approx G(c_A, c_B) \sum_j f_j. \tag{21}$$

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 distant galaxies. However, for two sets A and B which are merely well-separated (the distance between them is comparable to their diameters), this approximation introduces significant error. Let us not worry too much about the accuracy

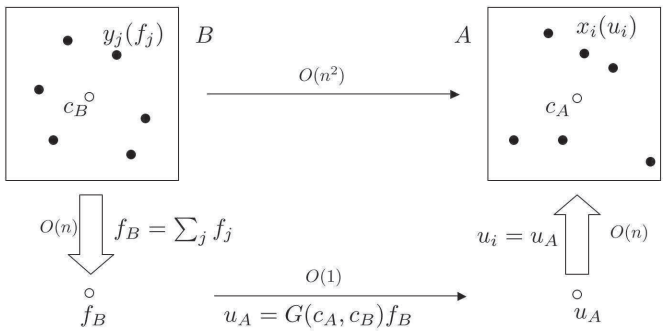

Fig. 9. The three steps of the approximate procedure. The total number operations is $O(N)$.

at this moment and we will come back to this point later. A geometric description of this approximation is given in Fig. 9.

We have introduced two representations in this simple approximation:

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

The computation of u_A from f_B

$$u_A = G(c_A, c_B) f_B$$

is called a *far-to-local* translation. Assuming both $\{y_j\}$ and $\{x_i\}$ contain $O(n)$ points, the naive direct computation of the interaction takes $O(n^2)$ steps. The proposed approximation is much more efficient:

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

Hence, the complexity of this three step procedure is $O(n)$. Viewing the interaction between A and B in a matrix form, we see that this interaction is approximately low rank if A and B are well-separated. In fact, in the above approximation, a rank-1 approximation is used.

However, in the problem we want to address, all the points $\{p_i\}$ are mixed together and each p_i is both a source and a target. Therefore, one cannot apply the above procedure directly. The solution is to use an adaptive tree structure, namely the octree in 3D or the quadtree in 2D (see Fig. 10). We first choose a box that contains all the points $\{p_i\}$. Starting from this top level box, each box of the quadtree is recursively partitioned unless the number of points inside it is less than a prescribed constant (in practice this number can vary from 50 to 200). Assuming that the points $\{p_i\}$ are distributed quite uniformly on ∂D, the number of levels of the quadtree is

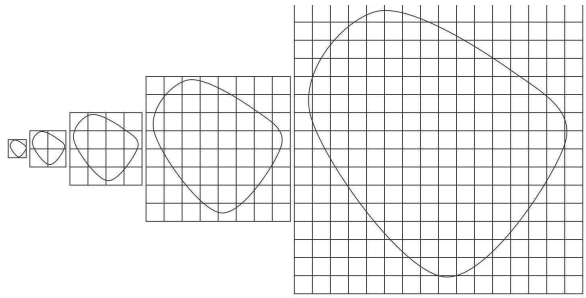

Fig. 10. The quadtree generated from the domain in Fig. 7. Different levels of the quadtree are shown from left to right.

$O(\log N)$. 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*. The *interaction list* of B contains the boxes on the same level that are *in B's far field but not the far field of B's parent*. It is not difficult to see that the size of the interaction list is always $O(1)$.

No computation is necessary at the zeroth and the first levels. At the second level (see Fig. 11), each box B has $O(1)$ well-separated boxes (e.g. A). These boxes are colored in gray and in B's interaction list. The interaction between B and each box in its interaction list can be approximated using the three step procedure described above. The same computation is repeated over all the boxes on this level.

To address the interaction between B and its adjacent boxes, we go to the next level (see Fig. 12). Suppose that B' is a child of B. Since the interaction between B' and B's far field has already been taken care of in the previous level, we only need

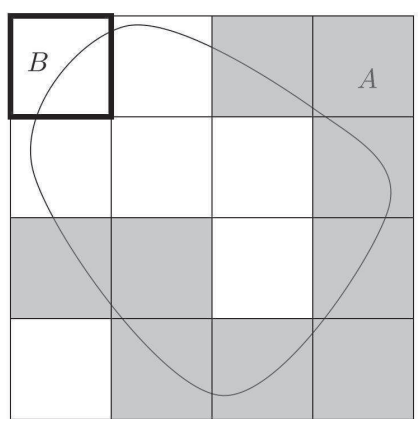

Fig. 11. Computation at the second level.

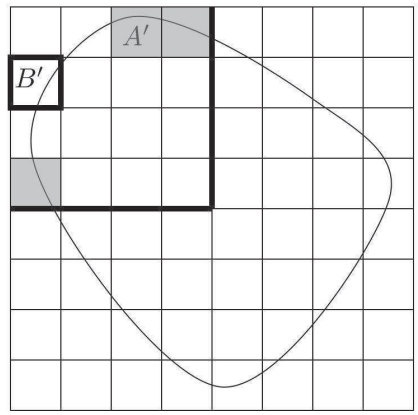

Fig. 12. Computation at the third level.

to address the interaction between B' and the boxes in B''s interaction list (e.g. A'). These boxes are also colored in gray and the interaction between B' and each one of them can be approximated again using the three step procedure described above.

To address the interaction between B' and its adjacent boxes, we again go to the next level (see Fig. 13). B'' (a child of B') has $O(1)$ boxes in its interaction list. The interaction between B'' and each one of them (e.g. A'') is once again computed using the three step procedure described above. Suppose now that B'' is also a leaf box. We then need to address the interaction between B'' and its adjacent boxes. Since the number of points in each leaf box is quite small, we simply use the direct computation for this.

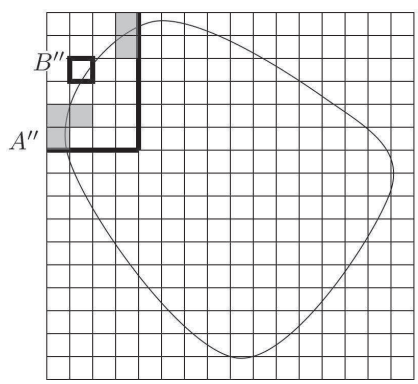

Fig. 13. Computation at the fourth (last) level.

The algorithm is summarized as follows:

1. At each level, for each box B, compute $f_B = \sum_{p_j \in B} f_j$.
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 is the far-to-local translation.
3. At each level, for each box A, add u_A to u_j for each $p_j \in A$.
4. At the final level, for each leaf box B, compute the interaction with its adjacent boxes directly.

The complexity of this algorithm is $O(N \log N)$ based on the following considerations.

1. Each point belongs to one box in each of $O(\log N)$ levels. The complexity of the first step is $O(N \log N)$.
2. There are $O(N)$ boxes in the octree. Each box has $O(1)$ boxes in the interaction list. Since each far-to-local translation takes $O(1)$ operations, the complexity of the second step is $O(N)$.
3. Each point belongs to one box in each of $O(\log N)$ levels. The complexity is $O(N \log N)$.
4. There are $O(N)$ leaf boxes in total. Each one has $O(1)$ neighbors. Since each leaf box contains only $O(1)$ points, the direct computation costs $O(N)$ steps.

As we have mentioned earlier, the goal is $O(N)$. Can we do better? The answer is yes. Let us take a look at a box B and its children $B_1, \cdots, B_4$. Based on the definition of f_B, we have

$$f_B = \sum_{p_j \in B} f_j = \sum_{p_j \in B_1} f_j + \sum_{p_j \in B_2} f_j + \sum_{p_j \in B_3} f_j + \sum_{p_j \in B_4} f_j = f_{B_1} + f_{B_2} + f_{B_3} + f_{B_4}.$$

Therefore, once $\{f_{B_i}\}$ are all known, f_B can be computed using only $O(1)$ operations. This step is called a *far-to-far translation*. The dependence between f_B and $\{f_{B_i}\}$ suggests that we traverse the quadtree bottom-up during the construction of the far field representations.

Similarly, instead of putting u_A to each of its points, it is sufficient to add u_A to $\{u_{A_i}\}$ where $\{A_i\}$ are the children of A. The reason is that u_{A_i} will eventually be added to the individual points. This step of adding u_A to $\{u_{A_i}\}$ obviously takes $O(1)$ operations as well and it is called a *local-to-local translation*. Since $\{u_{A_i}\}$ now depend on u_A, we need to traverse the octree top-down during the computation of the local field representations.

Combining the far-to-far and local-to-local translations with the above algorithm, we have the complete the description of the geometric structure of the FMM.

1. Bottom-up traversal of the octree. 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 far field representations of its children.
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.
3. Top-down traversal of the octree. 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.

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

Compared with the previous version, the only changes are made in the first and the third steps, while the second and the fourth steps remain the same. Let us estimate its complexity. It is obvious that we perform one far-to-far translation and one local-to-local translation to each of the $O(N)$ boxes in the octree. 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 clearly $O(N)$. Therefore, the overall complexity of the algorithm is $O(N)$.

3.2 Analytic part

In the discussion of the geometric part of the FMM, we did not worry too much about the accuracy. In fact, simply taking the far field representation $f_B = \sum_{p_j \in B} f_j$ and the local field representation $u_A = G(c_A, c_B) f_B$ gives very low accuracy. Next, we discuss the analytic part of the FMM, which provides efficient representations and translations that achieve any prescribed accuracy ε. In fact one can view the $f_B = \sum_{p_j \in B} f_j$ to be the zeroth moment of the charge distribution $\{f_j\}$ at $\{p_j\}$ in B. The idea behind the analytic part of the FMM is simply to utilize the higher order moments and represent them compactly using the property of the underlying PDE.

2D case

In the two dimensional case, we can regard $\{p_i\}$ to be points in the complex plane. Up to a constant,

$$G(x,y) = \ln|x-y| = \mathrm{Re}(\ln(x-y))$$

for $x, y \in \mathbb{C}$. Therefore, we will regard the kernel to be $G(x,y) = \ln(x-y)$ and throw away the imaginary part at the end of the computation.

Far field representation

Suppose that $\{y_j\}$ are source points inside a box (see Fig. 14) and $\{f_j\}$ are charges located at $\{y_j\}$.

Since

$$G(x,y) = \ln(x-y) = \ln x + \ln\left(1 - \frac{y}{x}\right) = \ln x + \sum_{k=1}^{\infty} \left(-\frac{1}{k}\right) \frac{y^k}{x^k},$$

we have for any x in the far field of this box

$$u(x) = \sum_j G(x, y_j) f_j = \left(\sum_j f_j\right) \ln x + \sum_{k=1}^{p} \left(-\frac{1}{k} \sum_j y_j^k f_j\right) \frac{1}{x^k} + O(\varepsilon)$$

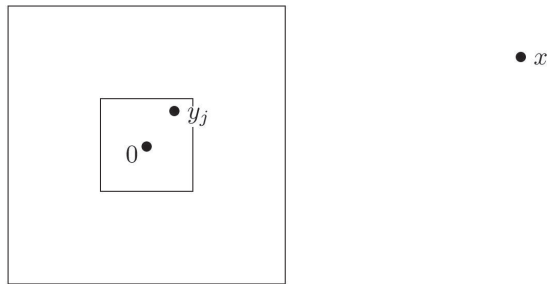

Fig. 14. Far field representation.

where $p = O(\log(1/\varepsilon))$ because $|y_j/x| < \sqrt{2}/3$. We define the far field representation to be the coefficients $\{a_k, 0 \le k \le p\}$ given by

$$a_0 = \sum_j f_j \quad \text{and} \quad a_k = -\frac{1}{k}\sum_j y_j^k f_j \quad (1 \le k \le p). \tag{22}$$

It is obvious that from $\{a_k\}$ we can approximate the potential for any point x in the far field efficiently within accuracy $O(\varepsilon)$. This representation clearly has complexity $O(\log(1/\varepsilon))$ and is also named the *multipole expansion*.

Local field representation

Suppose that $\{y_j\}$ are source points in the far field of a box (see Fig. 15) and $\{f_j\}$ are charges located at $\{y_j\}$.

From the Taylor expansion of the kernel

$$G(x,y) = \ln(x-y) = \ln(-y) + \ln\left(1 - \frac{x}{y}\right) = \ln(-y) + \sum_{k=1}^{\infty}\left(-\frac{1}{k}\right)\frac{x^k}{y^k},$$

we have, for any x inside the box,

$$u(x) = \sum_j G(x,y_j)f_j = \sum_j \ln(-y_j)f_j + \sum_{k=1}^{p}\left(-\frac{1}{k}\sum_j \frac{f_j}{y_j^k}\right)x^k + O(\varepsilon)$$

where $p = O(\log(1/\varepsilon))$ because $|x/y_j| < \sqrt{2}/3$. We define the local field representation to be the coefficient $\{a_k, 0 \le k \le p\}$ given by

$$a_0 = \sum_j \ln(-y_j)f_j \quad \text{and} \quad a_k = -\frac{1}{k}\sum_j \frac{f_j}{y_j^k} \quad (1 \le k \le p). \tag{23}$$

Based on $\{a_k\}$, we can approximate the potential for any point x inside the box efficiently within accuracy $O(\varepsilon)$. This representation has complexity $O(\log(1/\varepsilon))$ and is also named the *local expansion*.

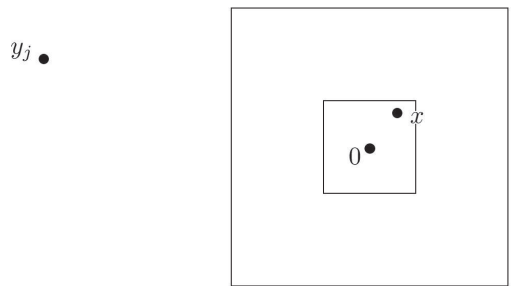

Fig. 15. Local field representation.

Far-to-far translation

Let us now consider the far-to-far translation which transforms the far field representation of a child box B' to the far field representation of its parent box B (see Fig. 16). We assume that B' is centered at a point z_0 while B is centered the origin. Suppose that the far field representation of child B' is $\{a_k, 0 \le k \le p\}$, i.e.,

$$u(z) = a_0 \ln(z - z_0) + \sum_{k=1}^{p} a_k \frac{1}{(z - z_0)^k} + O(\varepsilon)$$

for any z in the far field of B'. The far field representation $\{b_l, 0 \le l \le p\}$ of B is given by

$$b_0 = a_0 \quad \text{and} \quad b_l = -\frac{a_0 z_0^l}{l} + \sum_{k=1}^{l} a_k z_0^{l-k} \binom{l-1}{k-1} \quad (1 \le l \le p) \tag{24}$$

and for any z in the far field of B

$$u(z) = b_0 \ln z + \sum_{l=1}^{p} b_l \frac{1}{z^l} + O(\varepsilon).$$

From the definition of $\{b_l\}$, it is clear that each far-to-far translation takes $O(p^2) = O(\log^2(1/\varepsilon))$ operations.

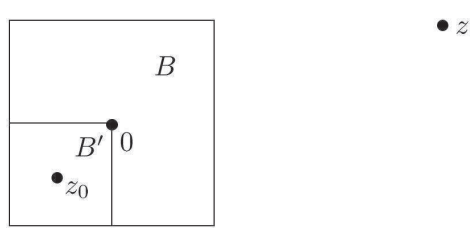

Fig. 16. Far-to-far translation.

Far-to-local translation

The far-to-local translation transforms the far field representation of a box B to the local field representation of a box A in B's interaction list. We assume that B is centered at z_0 while A is centered at the origin (see Fig. 17). Suppose that the far

Fig. 17. Far-to-local translation.

field representation of B is $\{a_k, 0 \le k \le p\}$, i.e.,

$$u(z) = a_0 \ln(z - z_0) + \sum_{k=1}^{p} a_k \frac{1}{(z - z_0)^k} + O(\varepsilon)$$

for any z in the far field of B. The local field representation $\{b_l, 0 \le l \le p\}$ of A is given by

$$b_0 = a_0 \ln(-z_0) + \textstyle\sum_{k=1}^{p} (-1)^k \frac{a_k}{z_0^k} \quad \text{and}$$
$$b_l = -\frac{a_0}{l z_0^l} + \frac{1}{z_0^l} \textstyle\sum_{k=1}^{p} \frac{a_k}{z_0^k} \binom{l+k-1}{k-1} (-1)^k \quad (1 \le l \le p).$$

and for any z in A

$$u(z) = \sum_{l=0}^{p} b_l z^l + O(\varepsilon).$$

It is clear that each far-to-local translation takes $O(p^2) = O(\log^2(1/\varepsilon))$ operations as well.

Local-to-local translation

The local-to-local translation transforms the local field representation of a parent box A to the local field representation of its child A'. We assume that the center of A is z_0 while the center of A' is the origin (see Fig. 18). Suppose that the local field representation of A is $\{a_k, 0 \le k \le p\}$, i.e.,

$$u(z) = \sum_{k=0}^{p} a_k (z - z_0)^k + O(\varepsilon)$$

for any z in A. Then the local field representation $\{b_l, 0 \le l \le p\}$ at A' is given by

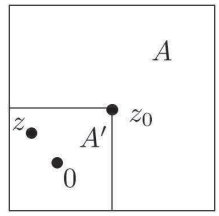

Fig. 18. Local-to-local translation.

$$b_l = \sum_{k=l}^{n} a_k \binom{k}{l} (-z_0)^{k-l} \quad (0 \le l \le p)$$

and for any z in A'

$$u(z) = \sum_{l=0}^{p} b_l z^l + O(\varepsilon).$$

The complexity of a local-to-local translation is again $O(p^2) = O(\log^2(1/\varepsilon))$. To summarize the 2D case, both the far and local field representations are of size $O(p) = O(\log(1/\varepsilon))$ for a prescribed accuracy ε. All three translations are of complexity $O(p^2) = O(\log^2(1/\varepsilon))$. Therefore, the complexity of the FMM algorithm based on these representations and translations is $O(N)$ where the constant depends on ε in a logarithmic way.

3D case

Up to a constant, the 3D Green's function of the Laplace equation is

$$G(x,y) = \frac{1}{|x-y|}.$$

For two points $x = (r,\theta,\varphi)$ and $x' = (r',\theta',\varphi')$ in spherical coordinates, we have an important identity

$$\frac{1}{|x-x'|} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} (r')^n Y_n^{-m}(\theta',\varphi') \frac{1}{r^{n+1}} Y_n^m(\theta,\varphi)$$

for $r \ge r'$.

Far field representation

Suppose that $\{y_j = (r_j,\theta_j,\varphi_j)\}$ are source points with charges $\{f_j\}$ inside a box centered at the origin. Let us consider the potential generated by $\{y_j\}$ at a point $x = (r,\theta,\varphi)$ in the far field (see Fig. 14). Using the given identity, we get

$$u(x) = \sum_j G(x, y_j) f_j = \sum_{n=0}^{p} \sum_{m=-n}^{n} \left(\sum_j f_j r_j^n Y_n^{-m}(\theta_j, \varphi_j) \right) \frac{1}{r^{n+1}} Y_n^m(\theta, \varphi) + O(\varepsilon)$$

where $p = \log(1/\varepsilon)$ because $|y_j/x| < \sqrt{3}/3$. We define the far field representation to be the coefficients $\{\alpha_n^m, 0 \le n \le p, -n \le m \le n\}$ given by

$$\alpha_n^m = \sum_j f_j r_j^n Y_n^{-m}(s_j).$$

From these coefficients $\{\alpha_n^m\}$, one can approximate $u(x)$ for any x in the far field efficiently.

Local field representation

Suppose that $\{y_j = (r_j, \theta_j, \varphi_j)\}$ are source points with charges $\{f_j\}$ in the far field of a box. Let us consider the potential generated by $\{y_j\}$ at a point x inside the box. We assume that the box is centered at the origin (see Fig. 15). Following the above identity, we have at x

$$u(x) = \sum_j G(x, y_j) = \sum_{n=0}^{p} \sum_{m=-n}^{n} \left(\sum_j f_j \frac{1}{r_k^{n+1}} Y_n^m(\theta_j, \varphi_j) \right) r^n Y_n^m(\theta, \varphi) + O(\varepsilon)$$

where $p = \log(1/\varepsilon)$ because $|x/y_j| < \sqrt{3}/3$. We define the local field representation to be the coefficients $\{\beta_n^m, 0 \le n \le p, -n \le m \le n\}$ given by

$$\beta_n^m = \sum_j f_j \frac{1}{r_k^{n+1}} Y_n^m(s_j).$$

It is clear that, from these coefficients $\{\beta_n^m\}$, one can approximate $u(x)$ for any x inside the box efficiently.

Far-to-far, far-to-local and local-to-local translations

Similar to the 2D case, we have explicit formulas for the three translations. The derivation of these formulas depend heavily on special function theories. We point to [16] for the details.

Since both the far field and local field representations have $O(p^2)$ coefficients, a naive implementation of these translations requires $O(p^4)$ operations, which is quite large even for moderate values of p. If we take a look at the FMM closely, we discover that the most time-consuming step is to perform the far-to-local translations. This is due to the fact that for each box B there can be as many as $6^3 - 3^3 = 189$ boxes in its interaction list. For each of these boxes, a far-to-local translation is required. Therefore, computing the far-to-local translations with a much lower complexity is imperative for the success of a 3D FMM implementation.

In [9], Cheng et al. introduce highly efficient ways for computing these translations. For the far-to-far and local-to-local translations, a "point and shoot" method is

used to reduce the complexity from $O(p^4)$ to $O(p^3)$. Let us consider for example the far-to-far translation between a child box B' and its parent B. The main idea is that if the z axes of the spherical coordinate systems at B' and B coincided, the transformation from the far field representation of B' to the ones of B would be computed in $O(p^3)$ steps. Therefore, the far-to-far translation is partitioned into three steps:

- "Rotate" the coordinate system at B' so that the z axis points to the center of B. The far field representation at B' is transformed accordingly. This step takes $O(p^3)$ operations.
- Perform the far-to-far translation from B' to B in the rotated coordinate system. This step takes $O(p^3)$ operation as well.
- Finally, "rotate" the coordinate system at B back to the original configuration and transform the far field representation at B accordingly. This step takes $O(p^3)$ operations as well.

For the far-to-local translation, the main idea is to use the plane wave (exponential) expansion, which diagonalizes the far-to-local translation. Given two well-separated boxes A and B, the steps are

- Transform the far field representation to six plane wave expansions, one for each of the six directions $\pm x, \pm y, \pm z$. This step has $O(p^3)$ complexity.
- Depending on the location of A, use one of the six plane wave expansions to compute the far-to-local translation from B to A. After this step, the local field representation at A is stored in the plane wave form. Since the plane wave expansion diagonalizes the far-to-local translation, the complexity of this step is $O(p^2)$.
- Transform the plane wave expansions at A back to the local field representation. Notice that at A there are also six plane wave expansions for six different directions. This step takes $O(p^3)$ operations as well.

Since the first step is independent of the target box A, one only needs to perform it once for each box B. The same is true for the last step as it is independent of the source box B. On the other hand, the second step, which can be called as many as 189 times for each box, is relatively cheap as its complexity is $O(p^2)$.

4 Kernel Independent Fast Multipole Method

The FMM introduced in the previous section is highly efficient yet quite technical. As we have seen, both the representations and translations in the 3D case depend heavily on the results from special functions and their derivations are far from trivial. The Laplace equation is only one of the elliptic PDEs with non-oscillatory kernels: other examples include the Stokes equations, the Navier equation, the Yukawa equation and so on. Deriving expansions and translations for the kernels of these equations one by one can be a tedious task. In this section, we introduce the kernel independent fast multipole method which addresses all these kernels in a unified framework [27]. Some of the ideas in this framework appeared earlier in [1, 4].

The geometric part of the kernel independent fast multipole method is exactly the same as the standard FMM. Hence, our discussion focuses only on the analytic part. We will start with the 2D case and then comment on the difference for the 3D case.

4.1 2D case

Far field representation

Let us consider a simple physics experiment first (see Fig. 19). Suppose that B is a box with radius r and that we have a set of charges $\{f_j\}$ at $\{y_j\}$ inside B. These charges generate a non-zero potential in the far field. Let us now put a metal circle of radius $\sqrt{2}r$ around these charges and connect this metal circle to the ground. As a result, a charge distribution would appear on this metal circle to cancel out the potential field generated by the charges inside the box. Due to the linearity of the problem, we see that the potential field due to the charges inside the box can be reproduced by the charge distribution on the circle if we flip its sign. This experiment shows that the volume charges inside the box can be replaced with an equivalent surface charge distribution on the circle if one is only interested in the potential in the far field.

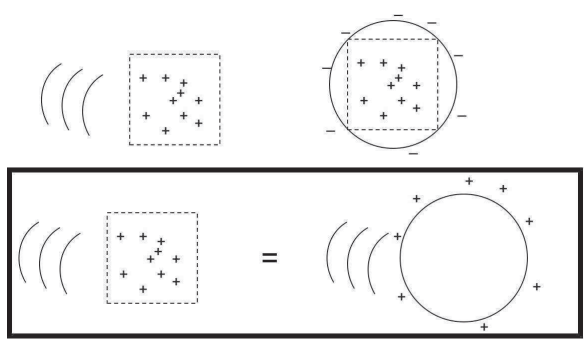

Fig. 19. The existence of an equivalent charge distribution.

A natural question to ask is, given a prescribed accuracy ε, how many degrees of freedom one needs to describe the equivalent charge distribution. Let us recall that the far field representation is only needed for the far field. It is well-known that the potential generated by the high frequency modes of the charge distribution on the circle dies out very quickly in the far field: it decays like $(\sqrt{2}/3)^n$ for the nth mode. As a result, we only need to capture the low frequency modes of the charge distribution. Our solution is to place $O(\log 1/\varepsilon)$ equally spaced points $\{y_k^{B,F}\}_k$ on the circle. The equivalent charges $\{f_k^{B,F}\}_k$ supported at these points are used as the far field representation (see Fig. 20).

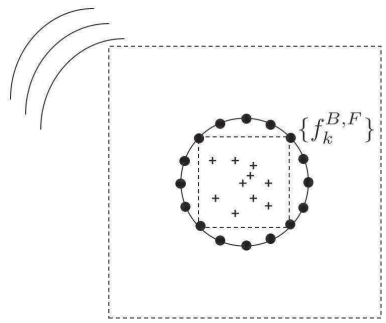

Fig. 20. The equivalent charges $\{f_k^{B,F}\}_k$ of the box B.

The next question is how to construct the equivalent charges $\{f_k^{B,F}\}_k$. One of the solutions is to pick a large circle of radius $(4-\sqrt{2})r$, the exterior of which contains the far field of B. If the potential fields generated by the source charges and $\{f_k^{B,F}\}_k$ are identical on this circle, then they have to match in the far field as well due to the uniqueness of the exterior problem of the Laplace equation. Based on this observation, the procedure of constructing $\{f_k^{B,F}\}_k$ consists of two steps (see Fig. 21).

- Pick $O(\log(1/\varepsilon))$ equally spaced locations $\{x_k^{B,F}\}_k$ on the large circle. Use kernel evaluation to compute the potentials $\{u_k^{B,F}\}_k$ at these locations generated by the charges inside B.
- Invert the interaction matrix between $\{y_k^{B,F}\}_k$ and $\{x_k^{B,F}\}_k$ to find $\{f_k^{B,F}\}_k$ so that they generate the potentials $\{u_k^{B,F}\}_k$. This inversion problem might be ill-posed, so one might need to regularize it with Tikhonov regularization [20].

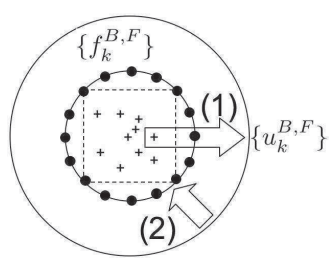

Fig. 21. The construction of the equivalent charges $\{f_k^{B,F}\}_k$.

Local field representation

Suppose that A is a box with radius r and that we have a set of charges $\{f_j\}$ located at points $\{y_j\}$ in the far field of A. To represent the potential field generated by these charges inside A, we first put a circle of radius $\sqrt{2}r$ around A (see the following figure). Let us call the potential on the circle the check potential field. From the uniqueness property of the interior problem of the Laplace equation, we know that, if we are able to capture the check potential field, we then can construct the potential everywhere in the box.

Similar to the case of the equivalent charge distribution, the next question is how many degrees of freedom we need to represent the check potential field. Since the potential is generated by points in the far field, it is quite smooth on the circle as the high frequency modes die out very quickly. Therefore, we only need a few samples to capture the check potential field. We put $O(\log(1/\varepsilon))$ samples $\{x_k^{A,L}\}_k$ on the circle. The potentials $\{u_k^{A,L}\}_k$ at these locations are taken to be the local field representation.

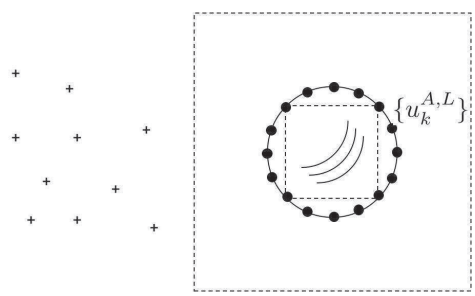

Fig. 22. The check potentials $\{u_k^{A,L}\}_k$ of the box B.

In order to reconstruct the potential inside the box A from the check potentials $\{u_k^{A,L}\}_k$, we first take a look at the example in Fig. 23. As before, the charges in the far field of A produce a potential field inside A. Let us now put a large metal circle of radius $(4-\sqrt{2})r$ around the box A and connect it to the ground. As a result, a charge distribution will appear on the large circle to cancel the potential field generated by the far field charges inside A. Again due to the linearity of the problem, we conclude that the potential due to the charges in the far field can be reproduced by the charge distribution on the large circle if we flip the sign of the surface charge distribution. This experiment shows that, if one can find the appropriate surface charge distribution on the large circle, the potential inside the box A can then be reconstructed.

Motivated by this example, we propose the following procedure to compute the potential inside A given the check potentials $\{u_k^{A,L}\}_k$ (see Fig. 24).

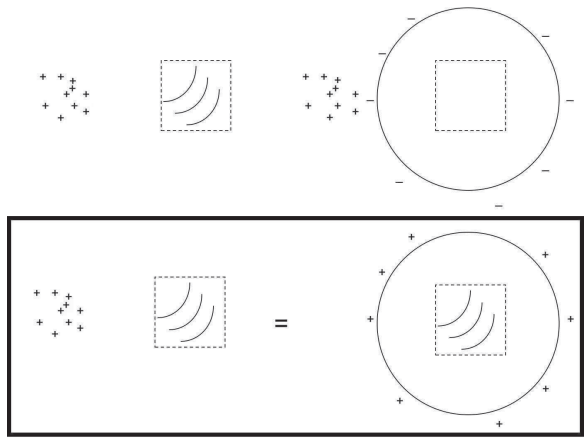

Fig. 23. The existence of equivalent charges for the local field representation.

- Pick $O(\log(1/\varepsilon))$ points $\{y_k^{A,L}\}_k$ on the large ring. Invert the interaction matrix between $\{y_k^{A,L}\}_k$ and $\{x_k^{A,L}\}_k$ to find the charges $\{f_k^{A,L}\}_k$ that produce $\{u^{A,L}\}_k$. This inversion might be ill-posed, so one might need to regularize it with Tikhonov regularization.
- Use the kernel evaluation to compute the potential inside A using the charges $\{f_k^{A,L}\}_k$.

To summarize, we have used the equivalent charges as the far field representation and the check potentials as the local field representation. Now let us consider the three translations of the kernel independent FMM.

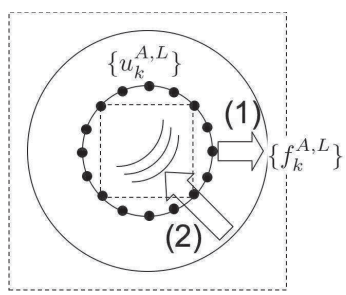

Fig. 24. The evaluation of the local field from the check potentials $\{u_k^{A,L}\}_k$.

Far-to-far translation

Given the equivalent charges of a child box B', the far-to-far translation computes the equivalent charges of the parent box B. The situation is similar to the construction of the equivalent charges that is described before if one is willing to consider the equivalent charges of B' as the source charges inside B. The steps of this translation are:

- Use the equivalent charges $\{f_k^{B',F}\}_k$ as source charges to evaluate the potential $\{u_k^{B,F}\}_k$ at $\{x_k^{B,F}\}_k$ (see Fig. 25).
- Invert the interaction between $\{y_k^{B,F}\}_k$ and $\{x_k^{B,F}\}_k$ to find the equivalent charges $\{f_k^{B,F}\}_k$. This step might again be ill-posed, so Tikhonov regularization might be needed.

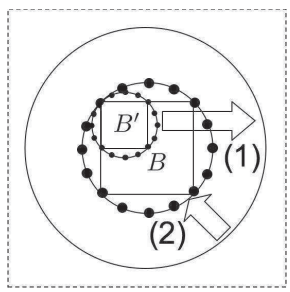

Fig. 25. Far-to-far translation.

Far-to-local translation

Given the equivalent charges of a box B, the far-to-local translation transforms them to the check potentials of a box A in B's interaction list (see the following figure). This translation is particularly simple for the kernel independent FMM. It consists of only a single step:

- Evaluate the potential $\{u_k^{A,L}\}_k$ using the equivalent charges $\{f_k^{B,F}\}_k$ (see Fig. 26).

Local-to-local translation

Given the check potentials of a parent box A, the local-to-local translation transforms them to the check potentials of its child box A'. The steps of the local-to-local translation are:

- Invert the interaction between $\{y_k^{A,L}\}_k$ and $\{x_k^{A,L}\}_k$ to find the equivalent charges $\{f_k^{A,L}\}_k$ that produce the check potentials $\{u_k^{A,L}\}_k$ (see Fig. 27). Tikhonov regularization is invoked whenever necessary.

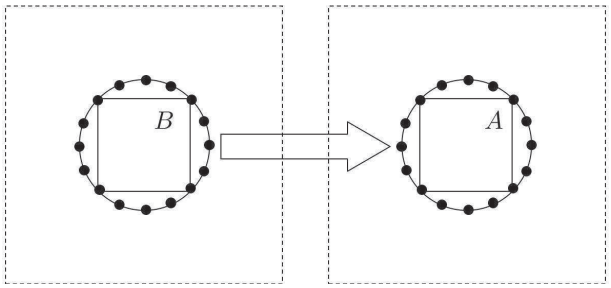

Fig. 26. Far-to-local translation.

- Check potentials $\{u_k^{A',L}\}_k$ are then computed using kernel evaluation with $\{f_k^{A,L}\}_k$ as the source charges.

Since the matrices used in the far-to-far and local-to-local translations only depend on the size of the boxes, their inversions can be precomputed and stored. Therefore, the kernel independent FMM algorithm only uses matrix vector multiplications and kernel evaluations. This general framework works well not only for PDE kernels such as the Green's functions of the Laplace equation, the Stokes equations, the Navier equation and Yukawa equation, but also for various radial basis functions after a slight modification.

4.2 3D case

In 3D, we need $O(p^2) = O(\log^2 1/\varepsilon)$ points to represent the equivalent charge distribution and the check potential field. If we put these points on a sphere, the three translations would require $O(p^4)$ operations. This poses the same problem we faced in the discussion of the 3D FMM algorithm.

In order to reduce this complexity, we choose to replace the sphere with the boundary of a box. This box is further discretized with a Cartesian grid and both

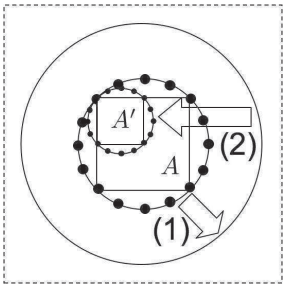

Fig. 27. Local-to-local translation.

the equivalent charges and the check potentials are located at the boundary points of the Cartesian grid. The main advantage of choosing the Cartesian grid is that the far-to-local translation, which is the most frequently used step, becomes a discrete convolution operator since the Green's function of the underlying PDE is translation invariant. This discrete convolution can be accelerated using the standard FFT techniques, and the resulting complexity of these translation operators are reduced to $O(p^3 \log p)$.

5 Hierarchical Matrices

Let us recall the computational problem that we face in each step of the iterative solution. Given a set of charges $\{f_i, 1 \le i \le N\}$ located at points $\{p_i, 1 \le i \le N\}$ (see Fig. 28) and the Green's function $G(x,y)$ of the Laplace equation, we want to compute at each p_i the potential

$$u_i = \sum_{j=1}^{N} G(p_i, p_j) f_j.$$

From the discussion above, we know that, if two sets A and B are well-separated, the interaction interaction $G(x,y)$ for $x \in A$ and $y \in B$ is approximately low rank. The hierarchical matrix framework puts this observation into an algebraic form. Our presentation in this section is far from complete, and we refer to [6] for a comprehensive treatment.

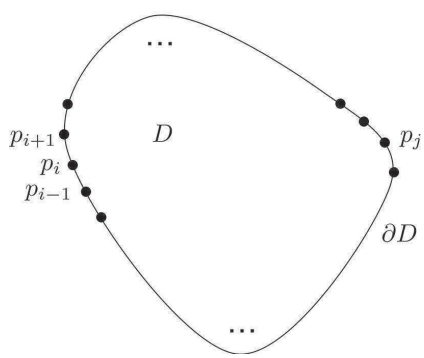

Fig. 28. Distribution of quadrature points $\{p_i\}$ on the boundary of the domain D.

5.1 Construction

Let us consider the following simple example where the domain D is a 2D disk. The boundary ∂D is subdivided into a hierarchical structure such that each internal node has two children and each leaf contains $O(1)$ points in $\{p_i\}$.

At the beginning level, ∂D is partitioned into with 4 large segments (see the following figure). Some pairs (e.g., A and B) are well-separated. Suppose the points $\{p_i\}$ are ordered according to their positions on the circle. As a result, the interaction from B to A corresponds to a subblock of the full interaction matrix $\boldsymbol{G} = (G(p_i, p_j))_{1 \le i,j \le N}$. Since the interaction between B and A is approximately low rank, this subblock can be represented in a low rank compressed form. All the subblocks on this level which have low rank compressed forms are colored in gray (see Fig. 29).

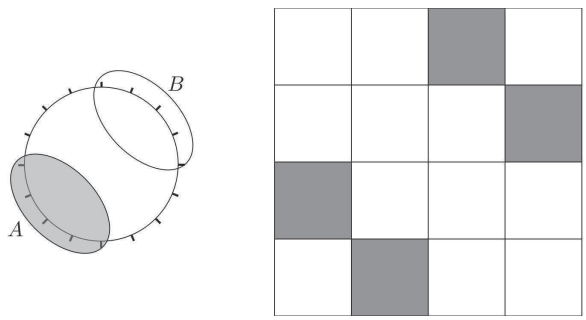

Fig. 29. Left: two well-separated parts on the second level of the hierarchical partition. In the matrix form, the interaction between them corresponds to a off-diagonal subblock. Right: all the blocks that correspond to well-separated parts on this level.

In order to consider the interaction between B and its neighbors, we go down to the next level. Suppose B' is a child of B. Similar to the case of the FMM, the interaction between B' and B's far field has already been taken care of in the previous level. We now need to consider the interaction between B' and the segments that are in the far field of B' but not the far field of B. There are only $O(1)$ segments in this region (colored in gray as well) and A' is one of them. As B' and A' are now well-separated, the interaction from B' to A' is approximately low rank. Therefore, the subblock that corresponds to this interaction can be stored in a low rank compressed form. All the subblocks on this level which have low rank compressed forms are again colored in gray (see Fig. 30).

We go down one level further to address the interaction between B' and its neighbors. For the same reason, B'' (a child of B') has only $O(1)$ segments in its far field but not in B''s far field. The subblocks that correspond to the interaction between B'' and these segments can be stored in low rank compressed forms (see Fig. 31). Suppose now that B'' is also a leaf segment. Since the interaction between B'' and its adjacent segments are not necessarily low rank, the subblocks corresponding to these interactions are stored densely. Noticing that each leaf segment contains only $O(1)$ points, the part of $\boldsymbol{G}$ that requires dense storage is $O(N)$.

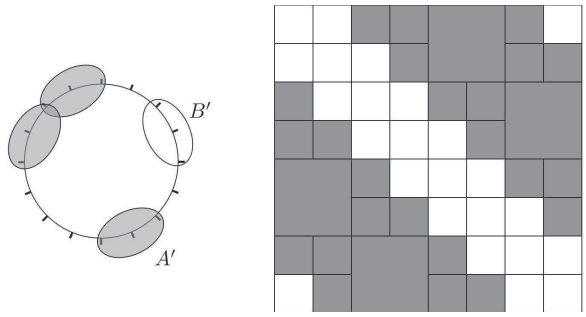

Fig. 30. At the third level.

From this simple example, we see that the hierarchical matrix framework is a way to partition the full interaction matrix into subblocks based on a hierarchical subdivision of the points. The off-diagonal blocks of a hierarchical matrix are compressed in low rank forms, while the diagonal and the next-to-diagonal blocks are stored densely.

A natural question at this point is which low rank compressed form one should use to represent the off-diagonal blocks. A first answer is to construct of these off-diagonal blocks first and then perform the truncated singular value decomposition (SVD) to compress them. The resulting form gives the best compression for a prescribed accuracy ε as the singular value decomposition is optimal in compressing matrices. However, there are two major disadvantages. First, the SVD usually requires one to construct the off-diagonal blocks first, which costs at least $O(N^2)$ operations. Second, since the singular vectors resulted from the SVD are not directly related to the vectors of the subblocks of $\boldsymbol{G}$, storing these vectors requires a lot of memory space. To overcome these two problems, we resort to several other methods.

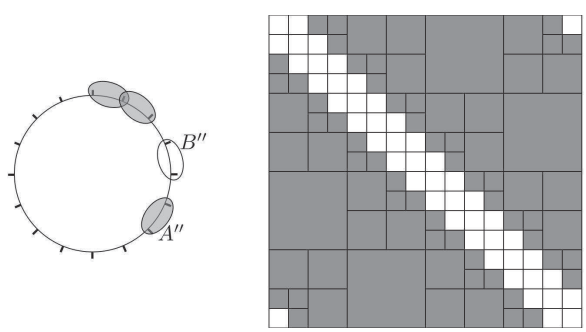

Fig. 31. At the fourth level.

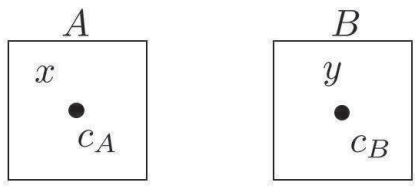

Fig. 32. Taylor expansion approach for constructing the low rank representations between two separated parts A and B.

Taylor expansion

Suppose that c_A and c_B are to be the center of segments A and B respectively (see Fig. 32). From the truncated Taylor expansion, we have

$$G(x,y) = \sum_{|\alpha|<p} \frac{1}{\alpha!} \partial_x^\alpha G(c_A,y)(x-c_A)^\alpha + O(\varepsilon) \tag{25}$$

where α is the multi-index and $p = O(\log(1/\varepsilon))$. In this derivation, we used the facts that

$$\partial_\alpha G(c_A,y) \approx \frac{1}{|y-c_A|^{|\alpha|}} \quad \text{and} \quad \frac{|x-c_A|}{|y-c_A|} \le \sqrt{2}/3.$$

This factorization provides us with a compressed form of rank $O(p^d)$.

There are two disadvantages of this approach. First, for complicated kernels, $\partial_x^\alpha G(x,y)$ is not easy to obtain. Even for the fundamental solution of the Laplace equation, this is far from trivial for large α. Second, Taylor expansion does not exploit the special structure of the kernel. Therefore, the resulting expansion has $O(p^d)$ terms where $d = 2,3$ is the dimension of the problem. This is quite wasteful comparing to the $O(p^{d-1})$ coefficients used in the FMM.

Tensor-product interpolation

In this approach, we pick a Cartesian grid to cover one of the domain (say A). The Cartesian grid is a tensor product of d one dimensional grids, each of which contains pth order Chebyshev points on an closed interval where $p = O(\log(1/\varepsilon))$. Suppose that $\{a_i\}$ are these grid points. Since $G(x,y)$ is smooth for $x \in A$ and $y \in B$ when A and B are well-separated, we have

$$G(x,y) = \sum_i G(a_i,y) L_i^A(x) + O(\varepsilon) \tag{26}$$

where $\{L_i^A(x)\}$ are d-dimensional Lagrange interpolants of the grid $\{a_i\}$ over A. Similarly, we pick a grid to cover B instead of A. Let $\{b_j\}$ be the grid points. For the same reason, we have

$$G(x,y) = \sum_j G(x,b_j) L_j^B(y) + O(\varepsilon) \tag{27}$$

where $\{L_j^B(y)\}$ are d-dimensional Lagrange interpolants of the grid $\{b_j\}$ over B. One can also choose to cover both A and B with Cartesian grids. In this case, we have

$$G(x,y) = \sum_i \sum_j L_i^A(x) G(a_i, b_j) L_j^B(y) + O(\varepsilon). \tag{28}$$

This tensor-product interpolation approach (illustrated in Fig. 33) is quite general since it only utilizes the kernel evaluation. However, similar to the Taylor expansion approach, it uses $O(p^d)$ terms, which is more than necessary.

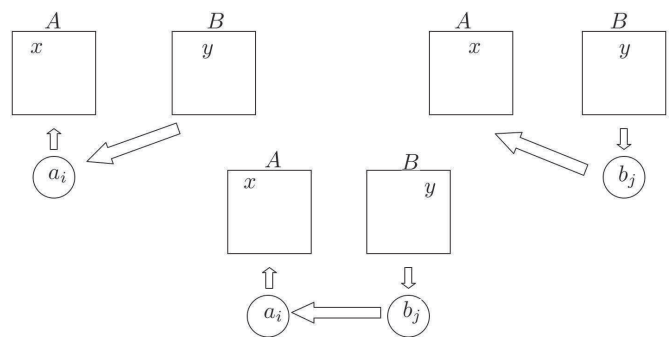

Fig. 33. Tensor-product interpolation approach for constructing the low rank representations between two separated parts A and B.

Pseudo-skeleton or cross approximation

Suppose $\{x_i, i \in I\}$ and $\{y_j, j \in J\}$ to be the point sets in A and B respectively. In our setting, they are subsets of $\{p_i\}$. We will use $\boldsymbol{G}_{I,J}$ to denote the subblock of $\boldsymbol{G}$ that corresponds to the interaction from B to A. Since the matrix $\boldsymbol{G}_{I,J}$ is approximately low rank, there exist a few columns of $\boldsymbol{G}_{I,J}$ which span its column space. Similarly, there exist a few rows of G which span its row space as well. The idea behind pseudo-skeleton approximation (or cross approximation) is to find these columns and rows and use them in the compressed representation. Suppose these columns correspond to the points $\{y_j, j \in L \subset J\}$ while these rows correspond to the points $\{x_i, i \in K \subset I\}$. The pseudo-skeleton approximation [14] is a factorization of the matrix $\boldsymbol{G}_{I,J}$ in the following form:

$$\boldsymbol{G}_{I,J} = \boldsymbol{G}_{I,L} \boldsymbol{M} \boldsymbol{G}_{K,J} + O(\varepsilon) \tag{29}$$

where $\boldsymbol{G}_{I,L}$ is the submatrix of $\boldsymbol{G}_{I,J}$ that contains the columns of points $\{y_j, j \in L\}$ while $\boldsymbol{G}_{K,J}$ is the submatrix of $\boldsymbol{G}_{I,J}$ that contains the rows of points $\{x_i, i \in K\}$ (see Fig. 34).

Several methods have been proposed to construct such a pseudo-skeleton approximation for $\boldsymbol{G}_{I,J}$. Approaches for selecting the sets $\{x_i, i \in K\}$ and $\{y_j, j \in L\}$ include greedy methods, adaptive methods, and random sampling techniques (see [6] for details). The middle matrix $\boldsymbol{M}$ is often computed using least square techniques.

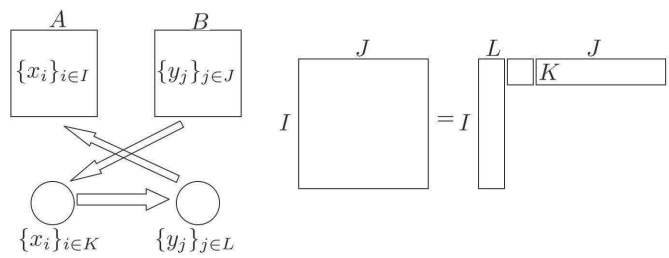

Fig. 34. Pseudo-skeleton approach for constructing the low rank representations between two separated parts A and B.

5.2 Hierarchical matrix arithmetics

Since the hierarchical matrix framework is an algebraic approach, it is possible to define a matrix arithmetics (addition, multiplication and inversion) for the hierarchical matrices. Here, we discuss these operations briefly.

Addition

Given two hierarchical matrices A and B with the same hierarchical structure, we seek a hierarchical matrix C such that $C \approx A + B$. Since both A and B have the same structure, we can perform the addition block by block. Suppose that P, Q and R are the blocks of A, B and C at the same location. There are two cases to consider. In the first case, P and Q are both dense blocks. Then R is simply the sum of P and Q. In the second case, both P and Q are stored in the compressed form. Let us further assume that $P = P_1 P_2^t$ and $Q = Q_1 Q_2^t$ where P_1, P_2, Q_1 and Q_2 are tall matrices. Then we have

$$R = \begin{pmatrix} P_1 & Q_1 \end{pmatrix} \begin{pmatrix} P_2 & Q_2 \end{pmatrix}^t .$$

The matrices $\begin{pmatrix} P_1 & Q_1 \end{pmatrix}$ and $\begin{pmatrix} P_2 & Q_2 \end{pmatrix}^t$ are further compressed using the pivoted QR factorization and the SVD.

Multiplication

Given two hierarchical matrices A and B with the same hierarchical structure, we seek a hierarchical matrix C such that $C \approx AB$. The multiplication algorithm for hierarchical matrices is similar to the one for block matrices. The basic step is to multiply two block matrices P and Q. There are four different cases to consider. In the first case, both P and Q are stored in the low rank compressed form, say $P = P_1 P_2^t$ and $Q = Q_1 Q_2^t$. Then

$$PQ = P_1 (P_2^t Q_1) Q_2^t.$$

In the second case, P is in the low rank form $P = P_1 P_2^t$ while Q is still in a hierarchical form. Without loss of generality, we assume

$$Q = \begin{pmatrix} Q_1 & Q_2 \\ Q_3 & Q_4 \end{pmatrix}.$$

By splitting P_2^t into

$$P_2^t = \begin{pmatrix} P_{2,1}^t & P_{2,2}^t \end{pmatrix},$$

we can compute PQ as

$$PQ = P_1 \left(\begin{pmatrix} P_{2,1}^t & P_{2,2}^t \end{pmatrix} \begin{pmatrix} Q_1 & Q_2 \\ Q_3 & Q_4 \end{pmatrix} \right).$$

where the multiplication in the parentheses is carried out recursively.

In the third case, P is in the hierarchical form while Q is in the low rank form $Q = Q_1 Q_2^t$. Let us assume that

$$P = \begin{pmatrix} P_1 & P_2 \\ P_3 & P_4 \end{pmatrix}.$$

By splitting Q_1 into

$$Q_1 = \begin{pmatrix} Q_{1,1} \\ Q_{1,2} \end{pmatrix},$$

we can compute PQ as

$$PQ = \left(\begin{pmatrix} P_1 & P_2 \\ P_3 & P_4 \end{pmatrix} \begin{pmatrix} Q_{1,1} \\ Q_{1,2} \end{pmatrix} \right) Q_2^t.$$

where the multiplication in the parentheses is carried out recursively.

In the last case, both P and Q is in the hierarchical form. We then resort to the block matrix multiplication algorithm and reduce its multiplication to the first three cases.

Inversion

Given a hierarchical matrix A, we want to compute a hierarchical matrix C such that $C \approx A^{-1}$. The solution is simply to apply the block matrix version of the LU factorization. Regular matrix addition and multiplication operations are now replaced with the ones of the hierarchical matrices described above.

6 Wavelet Based Methods

In this section, we consider yet another approach to the same problem discussed in the previous sections. Given a set of charges $\{f_j, 1 \le i \le N\}$ located at points $\{p_i, 1 \le i \le N\}$ (see the following figure) and the Green's function $G(x,y)$ of the Laplace equation, we want to compute at each p_i

$$u_i = \sum_{j=1}^{N} G(p_i, p_j) f_j.$$

Our discussion in this section focuses on the 2D case. Suppose that the boundary ∂D is parameterized by a periodic function $g(s)$ for $s \in [0,1]$. The matrix $\boldsymbol{G}$ with entries $G(p_i, p_j)$ can be viewed as an adequately sampled image of the continuous periodic 2D function $G(g(s), g(t))$. This image is smooth except at the diagonal where $s = t$ (see Fig. 35).

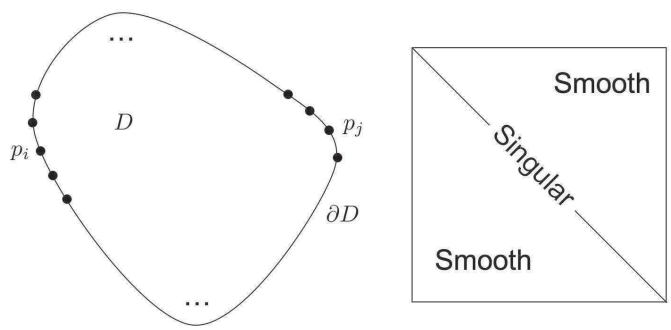

Fig. 35. Left: distribution of quadrature points $\{p_i\}$ on the boundary of the domain D. Right: a qualitative description of the 2D function $G(g(s), g(t))$.

Our plan is to find the best way to compress this matrix and then see whether the compression can help our computational problem. The presentation of this section follows [3].

6.1 Wavelet compression

One good way to compress such an image is to use 2D wavelets. Let us start our discussion with 1D wavelets on a unit periodic interval. The following discussion about the wavelets are quite brief and we refer the readers to [12, 21] for detailed exposition. Suppose that j is the index of the level and k is the spatial index. The scaling functions of the wavelet analysis are scaled and shifted copies of a mother scaling function $\varphi(x)$:

$$\{\varphi_{j,k}(x) := 2^{-j/2}\varphi(2^{-j}x - k)\}_{-\infty < j \le 0, 0 \le k < 2^{-j}}$$

Similarly, the wavelet functions are scaled and shifted versions of a mother wavelet function $\psi(x)$:

$$\{\psi_{j,k}(x) := 2^{-j/2}\psi(2^{-j}x - k)\}_{-\infty < j \le 0, 0 \le k < 2^{-j}}.$$

Let us assume that our wavelets are orthogonal and compactly supported. Therefore, a wavelet or a scaling function on the jth scale has a support of size $O(2^j)$. Due to the orthogonality condition, the scaling function at the 0th level $\varphi_{0,0}$ and the wavelets

$\{\psi_{j,k}\}_{-\infty<j\le 0, 0\le k<2^{-j}}$ form an orthogonal basis of $L^2([0,1])$. We further assume that our wavelets have M vanishing moments, i.e.,

$$\int \psi(x)x^m dx = 0, \quad m = 0, 1, \cdots, M-1$$

The 2D wavelets are built from the 1D wavelets using the tensor-product construction. The 2D wavelet orthobasis contains the following functions

$$\begin{aligned}
&\varphi_{0,(0,0)}(s,t) := \varphi_{0,0}(s)\varphi_{0,0}(t),\\
&\{\psi^1_{j,(k^1,k^2)}(s,t) := \varphi_{j,k^1}(s)\psi_{j,k^2}(t)\}_{-\infty<j<0, 0\le k^1,k^2<2^{-j}},\\
&\{\psi^2_{j,(k^1,k^2)}(s,t) := \psi_{j,k^1}(s)\varphi_{j,k^2}(t)\}_{-\infty<j<0, 0\le k^1,k^2<2^{-j}},\\
&\{\psi^3_{j,(k^1,k^2)}(s,t) := \psi_{j,k^1}(s)\psi_{j,k^2}(t)\}_{-\infty<j<0, 0\le k^1,k^2<2^{-j}}.
\end{aligned}$$

We now commit the *wavelet crime*: instead of studying how the discrete image $\boldsymbol{G}$ is compressed by the discrete 2D wavelet transform, we study wavelet coefficients of the continuous function $G(g(s), g(t))$ associated with the first N^2 elements of the orthobasis:

$$\begin{aligned}
&\varphi_{0,(0,0)},\\
&\{\psi^1_{j,(k^1,k^2)}\}_{-\log_2 N+1\le j\le 0, 0\le k<2^{-j}},\\
&\{\psi^2_{j,(k^1,k^2)}\}_{-\log_2 N+1\le j\le 0, 0\le k<2^{-j}},\\
&\{\psi^3_{j,(k^1,k^2)}\}_{-\log_2 N+1\le j\le 0, 0\le k<2^{-j}}
\end{aligned}$$

Since the singularity is only on the diagonal, we only need to focus on its neighborhood. Near the diagonal, the kernel $G(g(s), g(t)) = \ln|g(s) - g(t)|$ has the same singularity behavior as $\ln|s-t|$. Therefore, we can simply take the 2D function $G(g(s), g(t))$ to be $\ln|s-t|$.

Let us first estimate the coefficients of the first kind of wavelets $\{\psi^1_{j,(k^1,k^2)}\}$. Suppose $\psi^1_{j,(k^1,k^2)} = \varphi_{j,k^1}(s)\psi_{j,k^2}(t)$ and the two components $\varphi_{j,k^1}(\cdot)$ and $\psi_{j,k^2}(\cdot)$ have non-overlapping supports. Since our wavelets have M vanishing moments, we get

$$\begin{aligned}
&\iint \psi^1_{j,(k^1,k^2)}(s,t)\ln|s-t|dsdt\\
=&\int\int \varphi_{j,k^1}(s)\psi_{j,k^2}(t)\ln|s-t|dsdt\\
\le&\int |\varphi_{j,k^1}(s)|\left(\int \psi_{j,k^2}(t)\ln|s-t|dt\right)ds\\
\le&\int |\varphi_{j,k^1}(s)|\left(\max_{s\in supp(\varphi_{j,k^1}), t\in supp(\psi_{j,k^2})}\frac{2^{jM}}{|s-t|^M}\cdot\int \psi_{j,k^2}(t)dt\right)ds\\
\le& C\cdot\max_{s\in supp(\varphi_{j,k^1}), t\in supp(\psi_{j,k^2})}\frac{2^{j(M+1)}}{|s-t|^M}\\
\le& C\cdot\max_{s\in supp(\varphi_{j,k^1}), t\in supp(\psi_{j,k^2})}\frac{2^{jM}}{|s-t|^M}
\end{aligned}$$

for some constant C. Here we use the fact that $j \leq 0$. For a given accuracy ε, we set B to be $(1/\varepsilon)^{1/M}$. Suppose that the support of $\varphi_{j,k^1}(s)$ and $\psi_{j,k^2}(t)$ are separated so that $\min_{s\in supp(\varphi_{j,k^1}),t\in supp(\psi_{j,k^2})} |s-t| \geq B\cdot 2^j$, then

$$\iint \psi^1_{j,(k^1,k^2)}(s,t)\ln(|s-t|)dsdt = O(B^{-M}) = O(\varepsilon),$$

which is negligible.

Now let us count the number of non-negligible coefficients. For a fix level j and a fixed index k^2, the number of indices k^1 such that

$$\min_{s\in supp(\varphi_{j,k^1}),t\in supp(\psi_{j,k^2})} |s-t| \leq B\cdot 2^j$$

is $O(B)$. Therefore, for a fixed j and a fixed k^2, there are at most $O(B)$ wavelets $\{\psi^1_{j,(k^1,k^2)}(s,t)\}$ whose inner products with the kernel are greater than ε. The same argument works for the other two kinds of wavelets $\{\psi^2_{j,(k^1,k^2)}(s,t)\}$ and $\{\psi^3_{j,(k^1,k^2)}(s,t)\}$ because they all contain 1D wavelets (either in the variable s or in t). As a result, there are $O(3\cdot 2^{-j}\cdot B)$ non-negligible coefficients on each level j. Summing this over all $\log_2 N$ levels, we have in total

$$\sum_{j=-\log_2 N+1}^{0} O(3\cdot 2^{-j}\cdot B) = O(B\cdot N) = O(N)$$

non-negligible coefficients.

In order to compute the $O(N)$ non-negligible coefficients, we first notice that each wavelet or scaling function can be represented as the sum of a small number of scaling functions of the next level. Therefore, all we need to compute is the inner product of a function with a scaling function. Let us consider the 1D case to illustrate the idea.

$$\begin{aligned}\int f(x)\varphi_{j,k}(x)dx &= 2^{-j/2}\int f(x)\varphi(2^{-j}x-k+1)dx \\ &= 2^{-j/2}\int f(x+2^j(k-1))\varphi(2^{-j}x)dx\end{aligned}$$

Let us assume that, for some τ_M, our scaling functions satisfy

$$\int \varphi(x+\tau_M)x^m dx = 0, \quad m = 1,\cdots,M-1 \quad \text{and} \quad \int \varphi(x)dx = 1.$$

Scaling functions with these properties have been constructed in [3]. With the help of these properties, we have

$$2^{-j/2}\int f(x+2^j(k-1))\varphi(2^{-j}x)dx \approx 2^{j/2}f(2^j(k-1+\tau_M)) + O(2^{j(M+1/2)})$$

where the last equation uses only one point quadrature of f.

To summarize, we now have the following approximation

$$G(g(s),g(t)) = \sum_{i=1}^{O(N)} c_i \eta_i(s,t) + O(\varepsilon) \tag{30}$$

where $\eta_i(s,t) = \eta_i^1(s)\eta_i^2(t)$ is a 2D wavelet with non-negligible coefficient c_i for each i. This approximation is called the *non-standard form* of Beylkin, Coifman and Rokhlin. If we plot the supports of the wavelets with non-negligible coefficients, the figure looks very much like the one of the hierarchical matrices (see Fig. 36).

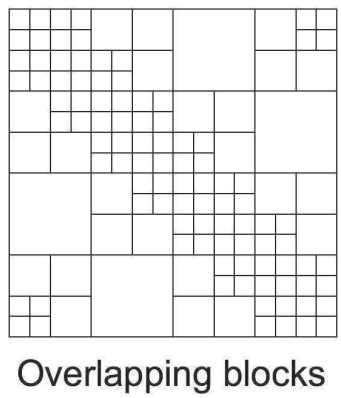

Fig. 36. The supports of the non-negligible terms in the non-standard form.

6.2 Fast matrix vector multiplication

We have so far focused on the compression the kernel matrix with the 2D wavelet basis. Let us discuss why this gives us a fast matrix vector multiplication algorithm. Using the non-standard form of the kernel matrix, we have

$$\begin{aligned}\int G(g(s),g(t))f(t)dt &\approx \sum_{i=1}^{O(N)} \int c_i \eta_i(s,t) f(t)dt \\ &= \sum_{i=1}^{O(N)} \int c_i \eta_i^1(s)\eta_i^2(t) f(t)dt \\ &= \sum_{i=1}^{O(N)} \eta_i^1(s) \cdot \left(c_i \int \eta_i^2(t) f(t)dt \right)\end{aligned}$$

where $\{\eta_i^1(s)\}$ and $\{\eta_i^1(t)\}$ are either wavelets or scaling functions in 1D. We recall that a fast wavelet transform produces in its intermediate steps the inner products of

the input function with all the scaling functions and wavelets. Therefore, the terms $\{\int \eta_i^2(t)f(t)dt\}$ for all i can be computed using a single fast wavelet transform by keeping all intermediate results.

Based on this simple observation, the wavelet based fast matrix multiplication algorithm has the following steps:

- Compute $\{\alpha_i = \int \eta_i^2(t)f(t)dt\}$ using a fast wavelet transform by keeping all intermediate results. The complexity of this step is $O(N)$.
- Compute $\{\beta_i = c_i\alpha_i\}$. This step takes only $O(N)$ operations.
- Synthesize $\sum_i \eta_i^1(s)\beta_i$ using an *extended* fast wavelet transform. This transform is *extended* in the sense that it includes not only the wavelet coefficients but also the scaling function coefficients since some of $\{\eta_i^1(s)\}$ are scaling functions. The complexity of this step is again $O(N)$.

As a result, the total complexity is $O(N)$, which is the same as the FMM. Before we end this section, let us summarize the main ideas behind the wavelet based method

- View the interaction matrix as an image. Since the singularity is only along the diagonal, a good compression with $O(N)$ non-negligible coefficients can be achieved using 2D wavelets.
- The tensor-product construction of the 2D wavelets allows one to use 1D fast wavelet transform to compute matrix vector multiplication in optimal time $O(N)$.

7 High Frequency FMM for the Helmholtz Kernel

In the rest two sections of this article, we discuss the computation of the oscillatory kernel of the Helmholtz equation in the 3D case.

$$-\Delta u - k^2 u = 0 \quad \text{in } \mathbb{R}^d \setminus \bar{D}.$$

Let us first rescale the geometry so that k is equal to 1. The equation then becomes

$$-\Delta u - u = 0 \quad \text{in } \mathbb{R}^d \setminus \bar{D}.$$

We face the following problem in each step of the iterative solver. Given a set of charges $\{f_i, 1 \le i \le N\}$ located at points $\{p_i, 1 \le i \le N\}$ and the Green's function

$$G(x,y) = h_0(|x-y|) = \frac{\exp(\mathrm{i}|x-y|)}{\mathrm{i}|x-y|}$$

of the Helmholtz kernel (up to a constant factor), we want to compute at each p_i

$$u_i = \sum_j G(p_i, p_j) f_j \tag{31}$$

(see Fig. 37).

In this section, we present the high frequency FMM (HF-FMM) by Rokhlin et al. [25, 26, 8] that calculates all $\{u_i\}$ in $O(N \log N)$ time. Suppose that the size of

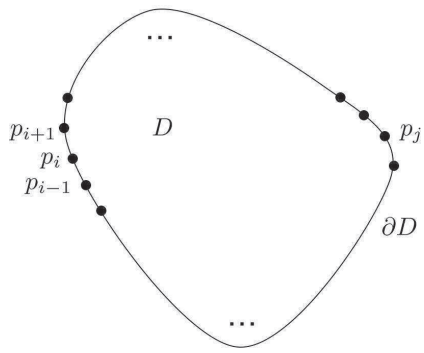

Fig. 37. Distribution of quadrature points $\{p_i\}$ on the boundary of the domain D.

the object is K wavelengths. Since one usually uses a constant number of points per wavelength in most of the scattering applications, $N = O(K^2)$.

For two points $x = (r, \theta, \varphi)$ and $x' = (r', \theta', \varphi')$ in spherical coordinates, we have the following important identity:

$$G(x,x') = h_0(|x-x'|) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} Y_n^{-m}(\theta', \varphi') j_n(r') Y_n^m(\theta, \varphi) h_n(r)$$

when $r > r'$.

Far field representation

Suppose that a set of charges $\{f_j\}$ are located at $\{y_j = (r_j, \theta_j, \varphi_j)\}$ inside a box centered at the origin. Let us consider the potential generated by $\{y_j\}$ at a point $x = (r, \theta, \varphi)$ in the far field (see Fig. 38). Using the identity just mentioned, we have

$$u(x) = \sum_j G(x, y_j) f_j = \sum_{n=0}^{p} \sum_{m=-n}^{n} \left(\sum_j f_j Y_n^{-m}(\theta_j, \varphi_j) j_n(r_j) \right) Y_n^m(\theta, \varphi) h_n(r) + \cdots$$

where p controls the number of terms to keep in the expansion and we will come back to it later. The far field representation is defined to be the coefficients $\{\alpha_n^m\}$ given by

$$\alpha_n^m = \sum_j f_j Y_n^{-m}(\theta_j, \varphi_j) j_n(r_j). \tag{32}$$

This representation is also called the *h-expansion* (see [8]).

Local field representation

Suppose that a set of charges $\{f_j\}$ are located at $\{y_j = (r_j, \theta_j, \varphi_j)\}$ in the far field of a box. Let us consider the potential generated by $\{y_j\}$ at a point $x = (r, \theta, \varphi)$ inside

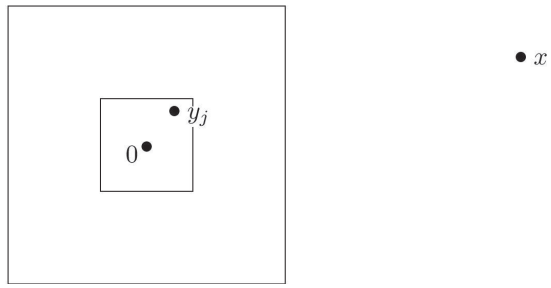

Fig. 38. Far field representation.

the box (see Fig. 39). We assume again that the center of the box is at the origin. From the identity given above, we have

$$u(x) = \sum_j G(x, y_j) f_j = \sum_{n=0}^{p} \sum_{m=-n}^{n} \left(\sum_j f_j Y_n^m(\theta_j, \varphi_j) h_n(r_j) \right) Y_n^{-m}(\theta, \varphi) j_n(r) + \cdots .$$

The local field representation is defined to be $\{\beta_n^m\}$ given by

$$\beta_n^m = \sum_j f_j Y_n^m(\theta_j, \varphi_j) h_n(r_j). \tag{33}$$

This representation is called the *j-expansion*.

The first question we need to address is what is the value of p, i.e., how many terms to keep in these two expansions for a prescribed accuracy ε. For a box with radius R, the n-th term of the h-expansion

$$u(x) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \left(\sum_j f_j Y_n^{-m}(\theta_j, \varphi_j) j_n(r_j) \right) Y_n^m(\theta, \varphi) h_n(r)$$

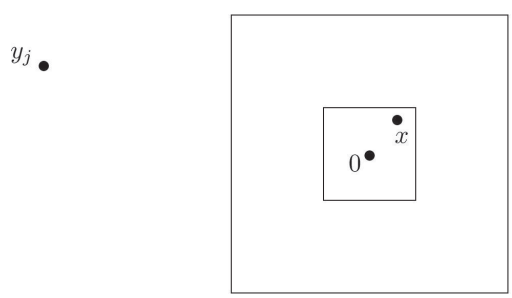

Fig. 39. Local field representation.

behaves like $h_n(3R)j_n(\sqrt{3}R)$. This product only decays when $n \geq 3R$. In order to get an accurate expansion, we are forced to choose $p = O(R)$ and keep all terms for $n = 0, 1, \cdots, p, -n \leq m \leq n$. Therefore, the number of coefficients in h-expansion is $O(R^2)$. The same is also true for the j-expansion. Let us recall that the point set $\{p_i\}$ is distributed on the boundary surface ∂D. It is not difficult to see that there are $O(R^2)$ points in a box with radius R as well. This means that, from the information theoretical point of view, there is no compression at all when one transforms the charges $\{f_j\}$ to the h-expansion coefficients $\{\alpha_n^m\}$.

When the radius of the box R is $O(1)$, the h-expansion and the j-expansion both have complexity $O(1)$. Therefore, it is still reasonable to use them as the far field and local field representations. The far-to-far, far-to-local, and local-to-local translations are very similar to the case of the 3D Laplace kernel:

- Far-to-far and local-to-local translations. The "point and shoot" approach is used. The complexity is cut down from $O(p^4)$ to $O(p^3)$ if the number of terms in the h-expansion is $O(p^2)$.
- Far-to-local translation. The plane wave (exponential) expansion is used to diagonalize the far-to-local translation. The translation between the h-expansion (or the j-expansion) and the plane wave expansion takes $O(p^3)$ operations, while each far-to-local translation in the plane wave expansion uses only $O(p^2)$ steps.

For large boxes, for example when $R = O(K)$, the situation is drastically different. The number of terms in the h-expansion or the j-expansion is equal to $O(R^2) = O(K^2)$. As a result, the complexity of the three translations are $O(R^3) = O(K^3) = O(N^{3/2})$, which is already higher than the $O(N \log N)$ complexity that we aim for. The solution to this problem, the so-called high frequency fast multipole method (HF-FMM), is to represent the h expansion and j expansion in a form such that the far-to-far, far-to-local, and local-to-local translations are all diagonalized.

Far field signature

For the h-expansion, we transform its coefficients $\{\alpha_n^m\}$ to

$$f(\theta, \varphi) := \sum_{n=0}^{p} \sum_{m=-n}^{n} \alpha_n^m (-1)^{n+1} Y_n^m(\theta, \varphi).$$

This function is in fact the *far field signature* of the potential of

$$u(x) = \sum_{n=0}^{p} \sum_{m=-n}^{n} \alpha_n^m Y_n^m(\theta, \varphi) h_n(r)$$

where $x = (r, \theta, \varphi)$. Similar, for j-expansion, we transform its coefficients $\{\beta_n^m\}$ to

$$g(\theta, \varphi) := \sum_{n=0}^{p} \sum_{m=-n}^{n} \beta_n^m (-1)^{n+1} Y_n^m(\theta, \varphi).$$

This function, which is also called the far field signature, can be viewed as a source distribution on a unit sphere which reproduces the potential inside the box if one

pushes the radius of the sphere to infinity and rescales the source distribution appropriately. All three translations are diagonalized in the far field signatures $f(\theta,\varphi)$ and $g(\theta,\varphi)$. We refer to [8, 10, 24] for the formulas of these translations.

In the HF-FMM algorithm, the octree is divided into the low and high frequency regimes. In a typical case, the low frequency regime contains all the boxes with radius < 1, while the boxes in the high frequency regime have radius ≥ 1. The h-expansion and the j-expansion serve as the far field and local field representations in the low frequency regime while the far field signatures $f(\theta,\varphi)$ and $g(\theta,\varphi)$ are the representations in the high frequency regime. A switch of the representations appears at the boxes with radius ≈ 1.

We would like to comment that the far field signatures $f(\theta,\varphi)$ and $g(\theta,\varphi)$ cannot be used for the low frequency box with radius $r < 1$. The reason is that the far-to-local translation of the far field signatures involves extremely large numbers when the box is too small. Therefore, given a fixed precision of calculation and a prescribed accuracy ε, one can only use the far field signatures on sufficiently large boxes in order to avoid numerical instability. The overall structure of the HF-FMM algorithm is shown in Fig. 40.

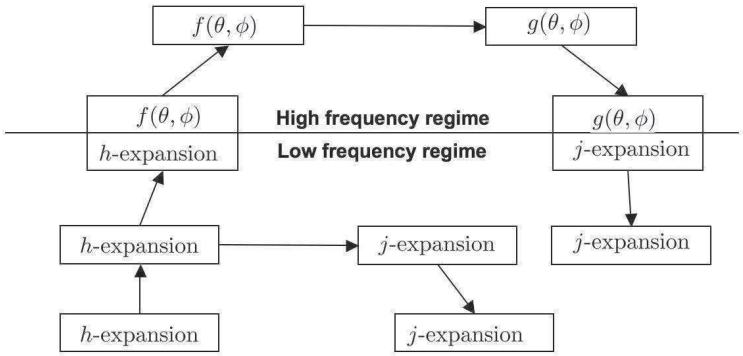

Fig. 40. The overall structure of the HF-FMM algorithm.

Most of the computation of HF-FMM is devoted to the high frequency regime, while the computation in the low frequency regime is similar to the one of the 3D Laplace kernel. Since the point set $\{p_i\}$ is sampled from the two-dimensional boundary ∂D, there are $O((K/r)^2)$ boxes with a fixed radius r. For each of them, the computation involves the far-to-far, far-to-local and local-to-local translations. Since all these translations are diagonalized, each of them has complexity $O(r^2)$. Therefore, the number of operations spent on the boxes with radius r is $O((K/r)^2)\cdot O(r^2) = O(K^2)$. Summing this over all $\log K$ level, we conclude that the complexity of the HF-FMM is

$$O(K^2 \log K) = O(N \log N).$$

8 Multidirectional Method

Through our discussion of the fast algorithms for the Laplace equation, we see that the interaction between a domain B and its far field has a low separation rank which is almost independent of the size of B. This low rank property has played a fundamental role in the fast multipole method, its kernel independent variant, and the hierarchical matrix framework.

As we have seen from the previous section, the situation is quite opposite for the Helmholtz equation. Suppose that B is a domain such that its radius is much larger than the wavelength. The interaction between B and its far field (see Fig. 41) through the Helmholtz kernel

$$G(x,y) = h_0(|x-y|) = \frac{\exp(\mathrm{i}|x-y|)}{\mathrm{i}|x-y|}$$

is not low rank anymore. In fact, the rank is proportional to the square of the radius of B. A natural question to ask is that whether it is possible to recover the low rank

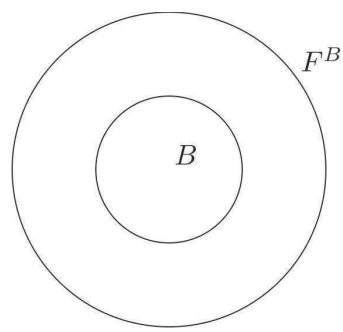

Fig. 41. The interaction between B and its far field F^B is not low rank for the Helmholtz kernel.

property in the setting of the Helmholtz kernel. The answer is positive and it is the motivation behind a multidirectional algorithm developed recently in [13].

8.1 Analysis

Directional parabolic separation

Let us start by considering the geometric configuration in Fig. 42. Suppose B is a domain with radius r. The wedge A, which has an opening angle $O(1/r)$, is at a distance r^2 or greater from B. Whenever two sets A and B follow this geometric configuration, we say that they satisfy the *directional parabolic separation condition*.

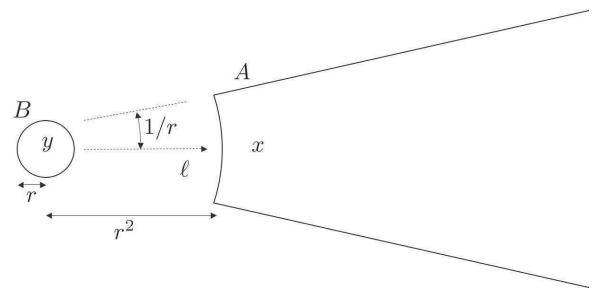

Fig. 42. Directional parabolic separated condition.

Theorem 5. *Suppose that B and A satisfy the directional parabolic separation condition. Then there exist an integer $T(\varepsilon)$ and two sets of functions $\{\alpha_i(x), 1 \le i \le T(\varepsilon)\}$ and $\{\beta_i(y), 1 \le i \le T(\varepsilon)\}$ such that, for any $x \in A$ and $y \in B$*

$$\left| \frac{\exp(\mathrm{i}|x-y|)}{\mathrm{i}|x-y|} - \sum_{i=1}^{T(\varepsilon)} \alpha_i(x)\beta_i(y) \right| < \varepsilon$$

where the number of terms $T(\varepsilon)$ of the expansion is independent of the radius of B.

The main idea behind this theorem is quite simple and it is not difficult to see why it can work. In the wedge A, the radiation generated by the points in B looks almost like a plane wave since the opening angle of the wedge A is inversely proportional to the radius of B. After one factors out the plane wave (which itself has a rank-1 separated representation), the rest of the interaction is smooth and hence has an approximate low rank separated representation.

The construction of $\{\alpha_i(x)\}$ and $\{\beta_i(y)\}$ is similar to the pseudo-skeleton approach discussed in the hierarchical matrix framework. In practice, the following randomized procedure works quite well.

- Randomly sample the set B to find positions $\{b_q\}$ such that the functions $\{G(x,b_q)\}_q$ span the space of the functions $\{G(x,y)\}_y$ within a prescribed accuracy ε.
- Randomly sample the set A to find positions $\{a_p\}$ such that the functions $\{G(a_p,y)\}_p$ span the space of the functions $\{G(x,y)\}_x$ within a prescribed accuracy ε.
- Find the matrix $D = (d_{qp})$ such that

$$\left| \frac{e^{\mathrm{i}|x-y|}}{\mathrm{i}|x-y|} - \sum_q \frac{e^{\mathrm{i}|x-b_q|}}{\mathrm{i}|x-b_q|} \sum_p d_{qp} \frac{e^{\mathrm{i}|a_p-y|}}{\mathrm{i}|a_p-y|} \right| = O(\varepsilon).$$

The first two steps use the pivoted QR factorizations, while the last step can be viewed as a least square problem.

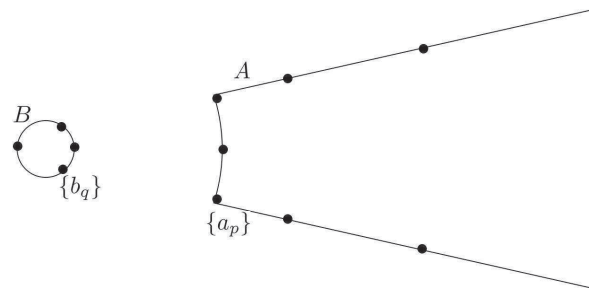

Fig. 43. Directional equivalent charges.

Directional equivalent charges

Suppose that B and A satisfy the directional parabolic separation condition. Let $\{y_j\}$ be a set of points in B. We consider the potential at $x \in A$ generated by the charges $\{f_j\}$ at $\{y_j\}$ (see Fig. 43).

Using the low rank representation generated above, we have

$$\left| \sum_i \frac{e^{\mathrm{i}|x-y_i|}}{\mathrm{i}|x-y_i|} f_i - \sum_q \frac{e^{\mathrm{i}|x-b_q|}}{\mathrm{i}|x-b_q|} \sum_p d_{qp} \sum_i \frac{e^{\mathrm{i}|a_p-y_i|}}{\mathrm{i}|a_p-y_i|} f_i \right| = O(\varepsilon).$$

This equation suggests that, by placing charges

$$\left\{ \sum_p d_{qp} \sum_i \frac{e^{\mathrm{i}|a_p-y_i|}}{\mathrm{i}|a_p-y_i|} f_i \right\}$$

at the points $\{b_q\}$, one can reproduce the potential at x accurately. We call these charges the *directional equivalent charges* of B in the direction of A. The above formula also provides a way to compute the directional equivalent charges from the source charges $\{f_j\}$:

- Evaluate the potentials at $\{a_p\}$ generated by $\{f_j\}$.
- Multiply the potentials with the matrix $D = (d_{qp})$ to obtain the directional equivalent charges.

Directional check potentials

Now let us reverse the roles of B and A (see Fig. 44). Suppose that $\{y_j\}$ are a set of points in the B. We consider the potential at $x \in A$ generated by the charges $\{f_j\}$ at $\{y_j\}$. Using the low rank representation of the kernel, we have

$$\left| \sum_i \frac{e^{\mathrm{i}|x-y_i|}}{\mathrm{i}|x-y_i|} f_i - \sum_q \frac{e^{\mathrm{i}|x-b_q|}}{\mathrm{i}|x-b_q|} \sum_p d_{qp} \sum_i \frac{e^{\mathrm{i}|a_p-y_i|}}{\mathrm{i}|a_p-y_i|} f_i \right| = O(\varepsilon).$$

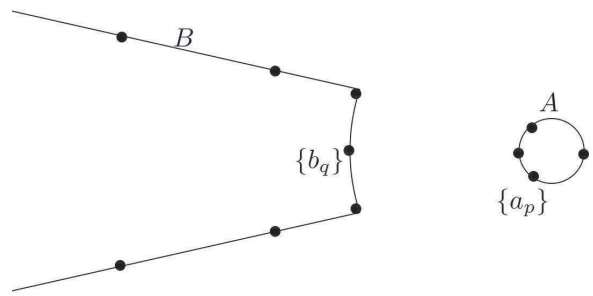

Fig. 44. Directional check potentials.

This equation shows that from the potentials

$$\left\{ \sum_i \frac{e^{\mathrm{i}|a_p - y_i|}}{\mathrm{i}|a_p - y_i|} f_i \right\}$$

at $\{a_p\}$ we can reconstruct the potential at any point $x \in A$ efficiently and accurately. The steps are:

- Multiply these potentials with the matrix $D = (d_{qp})$.
- Use the result as the charges at $\{b_q\}$ to compute the potential at x.

We call these potentials the *directional check potentials* of A in the direction of B.

8.2 Algorithmic description

Geometric part

Similar to the HF-FMM, the octree is divided into the low frequency regime (where the width of the box is < 1) and the high frequency regime (where the width of the box is ≥ 1). However, the definition of the far field region is much more complicated:

- For a box B with width $r < 1$ in the low frequency regime, the far field F^B contains all the well-separated boxes.
- For a box B with width $r \geq 1$ in the high frequency regime, the far field F^B contains the boxes which are at least r^2 away. The interaction list of B contains the boxes which are in the far field of B but not in the far field of the parent of B. All these boxes belong to a shell with radius from r^2 to $4r^2$. The far field is further partitioned into $O(r^2)$ wedges $\{W^{B,\ell}\}$ indexed by $\{\ell\}$, each with an opening angle of size $O(1/r)$ (see Fig. 45).

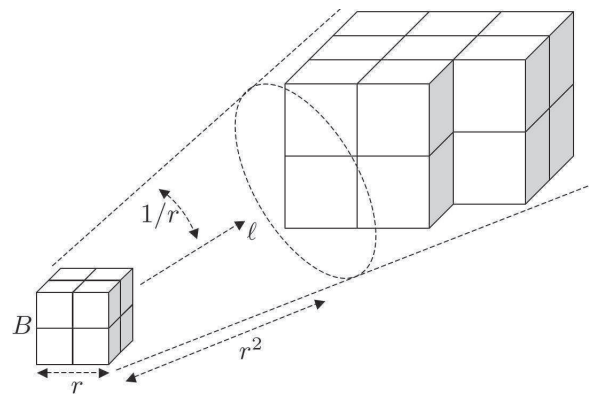

Fig. 45. The far field of B is partitioned into multiple wedges, each with an opening angle of size $O(1/r)$.

Far field and local field representations

For a box B with width $r < 1$ in the low frequency regime, the far field representation is the (non-directional) equivalent charges $\{f_k^{B,F}\}_k$ of the kernel independent FMM. From the previous discussion, we know that its complexity is $O(1)$.

For a box B with width $r \geq 1$ in the high frequency regime, the far field representation consists of the directional equivalent charges $\{f_k^{B,F,\ell}\}_k$ of all $O(r^2)$ wedges $\{W^{B,\ell}\}$. In order to compute the potential at a point x in the far field, we need to use the charges $\{f_k^{B,F,\ell}\}_k$ associated with the wedge $W^{B,\ell}$ that x belongs to. As we use $O(1)$ directional equivalent charges for each direction, the complexity of the far field representation is $O(r^2)$.

For a box A with width $r < 1$ in the low frequency regime, the local field representation is the (non-directional) check potentials $\{u_k^{A,L}\}_k$ of the kernel independent FMM. Its complexity is $O(1)$.

For a box A with width $r \geq 1$ in the high frequency regime, the local field representation consists of the directional check potentials $\{u_k^{A,L,\ell}\}_k$ of all $O(r^2)$ wedges $\{W^{A,\ell}\}$. For a point x in A, in order to compute the potential at x generated by the source charges in wedge $W^{A,\ell}$, we need to use the check potentials $\{u_k^{A,L,\ell}\}_k$. Since the directional check potentials for each direction contain $O(1)$ coefficients, the complexity of the local field representation is $O(r^2)$.

Far-to-far, far-to-local, and local-to-local translations

The translations in the low frequency regime are exactly the same as the ones of the kernel independent FMM. Therefore, we only discuss these translations in the high frequency regime.

The far-to-local translation is quite simple. Consider two boxes A and B in each other's interaction list. Suppose A is in $W^{B,\ell}$ and B is in $W^{A,\ell'}$. The far-to-local translation from B to A simply evaluates $\{u_k^{A,L,\ell'}\}_k$ using $\{f_k^{B,F,\ell}\}_k$.

For the far-to-far translation, we construct the directional equivalent charges of a parent box B from its child box B'. Let us consider the wedges $\{W^{B,\ell}\}$ one by one. An important observation, which is clear from the following figure, is that $W^{B,\ell}$ is contained in a wedge $W^{B',\ell'}$ of its child B'. Therefore, to construct $\{f_k^{B,F,\ell}\}_k$ at B, we can simply regard $\{f_k^{B',F,\ell'}\}_k$ as the source charges (see Fig. 46).

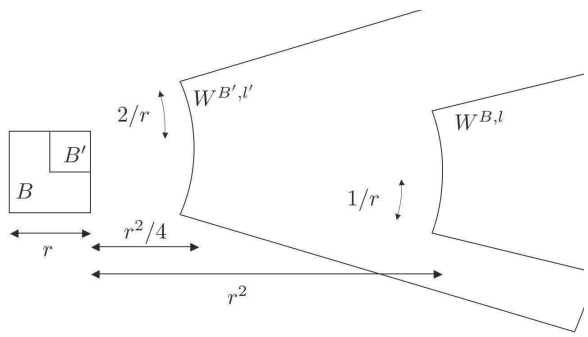

Fig. 46. Far-to-local translation between B and B'.

As a result, the steps of a far-to-local translation in the high frequency regime are:

- Use the directional equivalent charges $\{f_k^{B',F,\ell'}\}_k$ of B' as the source charges to compute the potentials at locations $\{a_p\}$ of the box B.
- Multiplication with the matrix (d_{qp}) to obtain $\{f_k^{B,F,\ell}\}_k$.

The local-to-local translation is implemented in a similar way. The main components of this algorithm is illustrated in Fig. 47.

Let us now discuss the computational complexity of this multidirectional algorithm. For a box of width r, most of the computation is devoted to the three translations.

- There are $O(r^2)$ far-to-far translations, one for each wedge. Since each far-to-far translation takes $O(1)$ operations, the complexity is $O(r^2)$.
- There are $O(r^2)$ local-to-local translations, again one for each wedge. Since each local-to-local translation takes also $O(1)$ operation, the complexity is again $O(r^2)$.
- Let us count the number of far-to-local translations for a box B. All the boxes in B's interaction list belong to a shell with radius between r^2 and $4r^2$. It is clear that there are $O(r^2)$ boxes in this shell since the points are sampled from the

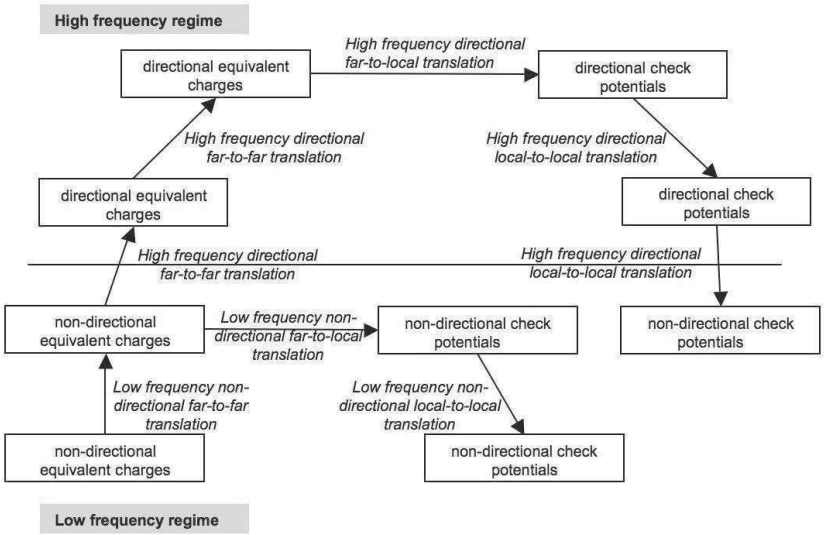

Fig. 47. The overall structure of the multidirectional algorithm.

surface boundary ∂D. Since each far-to-local translation takes $O(1)$ operations, the complexity is also $O(r^2)$.

For a given size r, there are $O(K^2/r^2)$ boxes of this size. Therefore, the number of steps spent on each level is $O(K^2/r^2) \cdot O(r^2) = O(K^2)$. Finally, summing over all $O(\log K)$ levels, we conclude that the complexity of this multidirectional algorithm is

$$O(K^2 \log K) = O(N \log N),$$

which is the same as the complexity of the HF-FMM algorithm.

9 Concluding Remarks

This paper discussed several fast algorithms for boundary integral equations. In the case of non-oscillatory kernels, we reviewed the fast multipole method (FMM) and its kernel independent variant, the hierarchical matrix framework, and the wavelet-based method. In each of these methods, we exploit the fact that the interaction between two well-separated regions is approximately low rank. For the oscillatory kernels, we discussed the high frequency fast multipole method (HF-FMM) and the recently proposed multidirectional algorithm. The HF-FMM used the far field signature to diagonalize the well-separated interaction, while the multidirectional algorithm decomposes the interaction in a directional way using the so-called directional parabolic separation condition.

Our choice of the methods is quite personal. Many other efficient algorithms were left out, such as the panel-clustering method [19], the FFT-based methods [4, 5, 7, 23], the local Fourier basis method [2], and the direct solver method [22]. Furthermore, we omitted the important field of time-domain integral equations, which often provide efficient ways to solve linear parabolic and hyperbolic PDEs. We point the readers to [10, 17] for fast algorithms for these boundary integral equations.

Acknowledgement. The author thanks the organizers of the 2007 Summer School on Multiscale Modeling and Simulation in Science for their hospitality and the students at this same school for their comments and suggestions on the manuscript. The author particularly acknowledges Stefan Engblom and Per Lötstedt for reading an early version of this article and correcting numerous mistakes and misprints. The author is supported by an Alfred P. Sloan Research Fellowship and a startup grant from the University of Texas at Austin.

10 Exercises

Exercise 1. Solve the Laplace equation

$$-\Delta u = 0 \quad \text{in } D$$

$$u = f \quad \text{on } \partial D$$

on the domain $D = \{(x_1, x_2) : x_1^2 + 4x_2^2 < 1\}$ using the second kind integral equation

$$f(z) = \frac{1}{2}\varphi(z) - \int_{\partial D} \frac{\partial G(z,y)}{\partial n(y)} \varphi(y) ds(y).$$

Let us parameterize the boundary ∂D as

$$x_1 = \cos(\theta) \quad x_2 = 0.5\sin(\theta), \quad \theta \in [0, 2\pi].$$

You can use the trapezoidal rule to numerically approximate the integral $\int_{\partial D} \frac{\partial G(z,y)}{\partial n(y)} \varphi(y) dy$. For this problem, $\lim_{y \to z} \frac{\partial G(z,y)}{\partial n(y)}$ exists and can be expressed in terms of the curvature of ∂D at z.

- Solve for $\varphi(z)$ when the boundary condition is $f(\theta) = \cos(4\theta)$ in the angular parameterization. Let the number of quadrature points N be 128.
- Please plot the solution $u(x)$ for x in $\{(x_1, x_2) : x_1^2 + 4x_2^2 < 0.9^2\}$.

Exercise 2. Solve the Helmholtz equation

$$-\Delta u - k^2 u = 0 \quad \text{in } \mathbb{R}^d \setminus \bar{D}$$

$$u(x) = -u^{inc}(x) \quad \text{for } x \in \partial D$$

$$\lim_{r \to \infty} r^{\frac{d-1}{2}} \left(\frac{\partial u}{\partial r} - \mathrm{i}ku \right) = 0$$

on the domain $\mathbb{R}^2 \setminus \bar{D}$ where $D = \{(x_1, x_2) : x_1^2 + 4x_2^2 < 1\}$ using the second kind integral equation

$$-u^{inc}(z) = \frac{1}{2}\varphi(z) + \int_{\partial D} \left(\frac{\partial G(z,y)}{\partial n(y)} \right) \varphi(y) ds(y).$$

where η is set to be zero. Let us use the same parameterization for ∂D as the previous problem and again the trapezoidal rule to discretize the integral. The following formula will be useful for the computation of $\frac{\partial G(z,y)}{\partial n(y)}$

$$\frac{d}{dr} H_0^n(r) = \frac{n H_n^1(r)}{r} - H_{n+1}^1(r).$$

The limit $\lim_{y \to z} \frac{\partial G(z,y)}{\partial n(y)}$ exists as well in this case and is equal to the one of the previous example.

- Choose $k = 64$ and $N = 8k$. Given $u^{inc}(x) = \exp(\mathrm{i}k(x \cdot d))$ with $d = (1,0)$, please solve for $\varphi(z)$.
- Pleases plot the scattering field $u(x)$ for x in $\{(x_1, x_2) : x_1^2 + 4x_2^2 > 1.1^2\}$.

Exercise 3. Let us consider the wavelet based method. The boundary is a circle parameterized by $g(s) = (\cos(2\pi s), \sin(2\pi s))$ for $s \in [0,1]$. Take the kernel to be the Green's function of the Laplace equation:

$$\ln |g(s) - g(t)|.$$

- Please discretize the kernel with N points. For the diagonal, simply put 0. This gives you an $N \times N$ image.
- Compress this image with 2D Daubechies wavelets.
- Compare, for different values of N and ε, how many wavelet coefficients are greater than ε.

References

1. C. R. Anderson. An implementation of the fast multipole method without multipoles. *SIAM J. Sci. Statist. Comput.*, 13(4):923–947, 1992.
2. A. Averbuch, E. Braverman, R. Coifman, M. Israeli, and A. Sidi. Efficient computation of oscillatory integrals via adaptive multiscale local Fourier bases. *Appl. Comput. Harmon. Anal.*, 9(1):19–53, 2000.
3. G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms. I. *Comm. Pure Appl. Math.*, 44(2):141–183, 1991.
4. E. Bleszynski, M. Bleszynski, and T. Jaroszewicz. AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems. *Radio Science*, 31:1225–1252, 1996.
5. N. Bojarski. K-space formulation of the electromagnetic scattering problems. Technical report, Air Force Avionic Lab. Technical Report AFAL-TR-71-75, 1971.

6. S. Börm, L. Grasedyck, and W. Hackbusch. Hierarchical matrices. Technical Report 21, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, 2003.
7. O. P. Bruno and L. A. Kunyansky. A fast, high-order algorithm for the solution of surface scattering problems: basic implementation, tests, and applications. *J. Comput. Phys.*, 169(1):80–110, 2001.
8. H. Cheng, W. Y. Crutchfield, Z. Gimbutas, L. F. Greengard, J. F. Ethridge, J. Huang, V. Rokhlin, N. Yarvin, and J. Zhao. A wideband fast multipole method for the Helmholtz equation in three dimensions. *J. Comput. Phys.*, 216(1):300–325, 2006.
9. H. Cheng, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm in three dimensions. *J. Comput. Phys.*, 155(2):468–498, 1999.
10. W. Chew, E. Michielssen, J. M. Song, and J. M. Jin, editors. *Fast and efficient algorithms in computational electromagnetics*. Artech House, Inc., Norwood, MA, USA, 2001.
11. D. Colton and R. Kress. *Inverse acoustic and electromagnetic scattering theory*, volume 93 of *Applied Mathematical Sciences*. Springer-Verlag, Berlin, second edition, 1998.
12. I. Daubechies. *Ten lectures on wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.
13. B. Engquist and L. Ying. Fast directional multilevel algorithms for oscillatory kernels. *SIAM J. Sci. Comput.*, 29(4):1710–1737 (electronic), 2007.
14. S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra Appl.*, 261:1–21, 1997.
15. L. Greengard. *The rapid evaluation of potential fields in particle systems*. ACM Distinguished Dissertations. MIT Press, Cambridge, MA, 1988.
16. L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348, 1987.
17. L. Greengard and J. Strain. A fast algorithm for the evaluation of heat potentials. *Comm. Pure Appl. Math.*, 43(8):949–963, 1990.
18. W. Hackbusch. *Integral equations*, volume 120 of *International Series of Numerical Mathematics*. Birkhäuser Verlag, Basel, 1995. Theory and numerical treatment, Translated and revised by the author from the 1989 German original.
19. W. Hackbusch and Z. P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numer. Math.*, 54(4):463–491, 1989.
20. R. Kress. *Linear integral equations*, volume 82 of *Applied Mathematical Sciences*. Springer-Verlag, New York, second edition, 1999.
21. S. Mallat. *A wavelet tour of signal processing*. Academic Press Inc., San Diego, CA, 1998.
22. P. G. Martinsson and V. Rokhlin. A fast direct solver for boundary integral equations in two dimensions. *J. Comput. Phys.*, 205(1):1–23, 2005.
23. K. Nabors, F. Korsmeyer, F. Leighton, and J. K. White. Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory. *SIAM Journal on Scientific and Statistical Computing*, 15:713–735, 1994.
24. N. Nishimura. Fast multipole accelerated boundary integral equation methods. *Applied Mechanics Reviews*, 55(4):299–324, 2002.
25. V. Rokhlin. Rapid solution of integral equations of scattering theory in two dimensions. *J. Comput. Phys.*, 86(2):414–439, 1990.
26. V. Rokhlin. Diagonal forms of translation operators for the Helmholtz equation in three dimensions. *Appl. Comput. Harmon. Anal.*, 1(1):82–93, 1993.
27. L. Ying, G. Biros, and D. Zorin. A kernel-independent adaptive fast multipole algorithm in two and three dimensions. *J. Comput. Phys.*, 196(2):591–626, 2004.