CHAPTER 1

Fast summation and multipole expansions

The Fast Multipole Method (FMM) was originally developed as an algorithm for rapidly evaluating all pairwise interactions between a given set of N electrically charged particles in two or three dimensions. A general problem of this type can be written as

$$u_i = \sum_{j=1}^{N} G(\boldsymbol{x}_i, \boldsymbol{x}_j)\, q_j, \qquad i = 1,\, 2,\, \ldots,\, N, \tag{1.1}$$

where $\{\boldsymbol{x}_i\}_{i=1}^N$ is a given set of point locations, where $\{q_i\}_{i=1}^N$ is a given set of corresponding sources, where $\{u_i\}_{i=1}^N$ is a set of potential values to be determined, and where $G(\boldsymbol{x}, \boldsymbol{y})$ is the kernel associated with the electrostatic problem. In two dimensions, we recall that the fundamental solution of the Laplace equation is $\phi(\boldsymbol{x}) = -(2\pi)^{-1} \log |\boldsymbol{x}|$. To keep formulas less cluttered, we will omit the scaling in front of the logarithm, and will use the kernel

$$G(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \log |\boldsymbol{x} - \boldsymbol{y}| & \boldsymbol{x} \neq \boldsymbol{y}, \\ 0 & \boldsymbol{x} = \boldsymbol{y}. \end{cases} \tag{1.2}$$

Directly evaluating all interactions in (1.1) would require $O(N^2)$ operations, but Greengard and Rokhlin [**11**] demonstrated that the task could to any finite precision $\epsilon > 0$ be solved in $O(N)$ operations, with a scaling constant that depends logarithmically on $1/\epsilon$ as $\epsilon \to 0$. The FMM has since then been generalized to work for a range of other interaction potentials and has today become one of the core algorithms in many areas high performance computing.

To introduce the concepts supporting fast summation techniques like the FMM, we will in Chapters 1 and 2 describe a bare-bones algorithm for solving the basic problem of evaluating all electro-static interactions between N charged particles in the plane that we described above, which is also the problem considered in the original paper [**11**] of Greengard and Rokhlin. We will then in Chapter 3 briefly discuss ways in which the FMM can be extended to problems in three dimensions, and to other interaction kernels such as those associated with the Helmholtz or Stokes equations. We will also briefly mention ways in which the FMM can be accelerated and improved from an algorithmic point of view.

1.1. Separation of variables

Let us in this section *temporarily* consider a simplified summation problem that illustrates the idea of a separation of variables, which is a foundational concept in the FMM. The simplification is that we for now suppose that the points where the sources are located and the points where we seek to evaluate the potentials are separated, as shown in Figure 1.1. We are given a set of electric sources $\{q_j\}_{j=1}^N$ at locations $\{\boldsymbol{y}_j\}_{j=1}^N$ in one box Ω_σ and seek the potentials these sources induce at some target locations $\{\boldsymbol{x}_i\}_{i=1}^M$ in a different box Ω_τ. In other words, we seek to evaluate the sum

$$u_i = \sum_{j=1}^{N} G(\boldsymbol{x}_i, \boldsymbol{y}_j)\, q_j, \qquad i = 1,\, 2,\, \ldots,\, M. \tag{1.3}$$

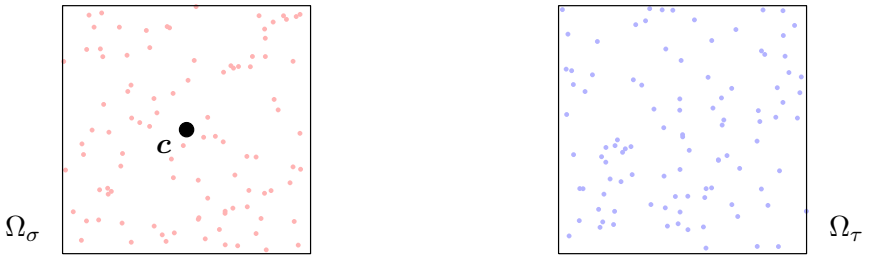


FIGURE 1.1. Geometry of the simplified potential evaluation problem discussed in Sections 1.1 and 1.2. The red points are sources and the blue points are targets. The point $\boldsymbol{c}$ is the center of the source box Ω_σ.

The key observation is now that the kernel function that represents electrostatic interactions in the plane, $G(\boldsymbol{x}, \boldsymbol{y}) = \log|\boldsymbol{x} - \boldsymbol{y}|$, is smooth whenever $\boldsymbol{x}$ and $\boldsymbol{y}$ are not close. This means that G can be approximated by a short sum of tensor products

$$G(\boldsymbol{x}, \boldsymbol{y}) \approx \sum_{p=0}^{P-1} B_p(\boldsymbol{x})\, C_p(\boldsymbol{y}), \qquad \text{when } \boldsymbol{x} \in \Omega_\tau,\ \boldsymbol{y} \in \Omega_\sigma, \tag{1.4}$$

where P is a small integer called the *interaction rank*. (Section 1.2 provides detailed formulas for B_p and C_p, and describes how to choose P to attain a requested accuracy.) Inserting (1.4) into (1.3), we find that

$$u_i \approx \sum_{j=1}^{N} \sum_{p=0}^{P-1} B_p(\boldsymbol{x})\, C_p(\boldsymbol{y})\, q_j = \sum_{p=0}^{P-1} B_p(\boldsymbol{x}) \left(\sum_{j=1}^{N} C_p(\boldsymbol{y})\, q_j \right). \tag{1.5}$$

In other words, we can evaluate an approximation to the sum (1.3) via the two steps

$$\hat{q}_p := \sum_{j \in I_\sigma} C_p(\boldsymbol{x}_j)\, q_j, \qquad p = 0,\, 1,\, 2,\, \ldots,\, P-1. \tag{1.6}$$

and

$$u_i \approx \sum_{p=0}^{P-1} B_p(\boldsymbol{x}_i)\, \hat{q}_p, \qquad i = 1,\, 2,\, \ldots,\, M. \tag{1.7}$$

While evaluating (1.3) directly requires MN operations, evaluating (1.6) and (1.7) requires only $P(M+N)$ operations. The power of this observation stems from the fact that high accuracy is achieved even for small P when the regions Ω_σ and Ω_τ are moderately well separated, cf. Section 2.6.

Using matrix notation, the approximation (1.4) implies that the $M \times N$ matrix $\mathbf{A}$ with entries $\mathbf{A}(i,j) = G(\boldsymbol{x}_i, \boldsymbol{y}_j)$ admits an approximate rank-P factorization $\mathbf{A} \approx \mathbf{BC}$. Then clearly the matrix-vector product $\mathbf{Aq}$ can efficiently be evaluated via $\mathbf{Aq} \approx \mathbf{B}\,(\mathbf{Cq})$.

In the summation problem that we are actually interested in, defined by (1.1), the sets of target locations and source locations coincide. In this case, no one relation like (1.4) can hold for all combinations of target and source points. Instead, our plan is to cut the domain up into pieces that we organize in a hierarchical tree. Then we use approximations such as (1.4) to evaluate interactions between distant pieces, and direct evaluation for points that are close. From a linear algebraic perspective, one could alternatively say that we will efficiently evaluate the matrix-vector product $\mathbf{u} = \mathbf{Aq}$ by exploiting rank-deficiencies in certain off-diagonal blocks of $\mathbf{A}$.

1.2. Multipole expansions

In Section 1.1, we refrained from specifying the exact expressions for the functions B_p and C_p in the approximation formula (1.4). This was done to emphasize that it is the *form* of the approximation that matters — each term in the sum is a product of one factor that depends only on $\boldsymbol{x}$ and one factor that depends only on $\boldsymbol{y}$. In this section, let us describe in some additional detail a classical technique for building an approximation to the kernel function $G(\boldsymbol{x}, \boldsymbol{y}) = \log|\boldsymbol{x} - \boldsymbol{y}|$, based on so called *multipole expansions.* These can be viewed as efficient representations of harmonic potentials. To be precise, suppose that a box shaped domain Ω_σ holds a set of N sources with charges $\{q_j\}_{j=1}^N$ at locations $\{\boldsymbol{y}_j\}_{j=1}^N \subset \Omega_\sigma$. At a target point $\boldsymbol{x}$, the electric potential $u(\boldsymbol{x})$ caused by these charges is given by

$$u(\boldsymbol{x}) = \sum_{j=1}^{N} G(\boldsymbol{x}, \boldsymbol{y}_j)\, q_j. \tag{1.8}$$

A very simple and intuitively natural approximation to the potential $u(\boldsymbol{x})$ is obtained by evaluating the sum $\hat{q}_0$ of all the charges

$$\hat{q}_0 = \sum_{j=1}^{N} q_j, \tag{1.9}$$

and then approximating the exact potential u by the potential caused by a single point charge of strength $\hat{q}_0$ placed at the center $\boldsymbol{c}$ of the box,

$$u(\boldsymbol{x}) \approx \hat{q}_0 \log|\boldsymbol{x} - \boldsymbol{c}|. \tag{1.10}$$

The approximation error in (1.10) tends to zero as $|\boldsymbol{x} - \boldsymbol{c}| \to \infty$, but at a quite slow rate that would typically be $O(|\boldsymbol{x} - \boldsymbol{c}|^{-1})$. The approximation can be improved by adding a dipole charge at $\boldsymbol{c}$ so that

$$u(\boldsymbol{x}) \approx \hat{q}_0 \, \log|\boldsymbol{x} - \boldsymbol{c}| - \mathbf{d} \cdot \frac{\boldsymbol{x} - \boldsymbol{c}}{|\boldsymbol{x} - \boldsymbol{c}|^2}. \tag{1.11}$$

The optimal dipole charge vector $\mathbf{d}$ is given by

$$\mathbf{d} = \sum_{j=1}^{N} q_j \, \big(\boldsymbol{y}_j - \boldsymbol{c}\big).$$

The error in (1.11) would typically decay as $O(|\boldsymbol{x} - \boldsymbol{c}|^{-2})$ as $|\boldsymbol{x} - \boldsymbol{c}| \to \infty$.

The approximations (1.10) and (1.11) form the first two terms in the so called *multipole expansion* of the harmonic function u. The full expansion is conveniently expressed using polar coordinates

$$\boldsymbol{x} - \boldsymbol{c} = r\,(\cos\theta,\, \sin\theta), \qquad \boldsymbol{y} - \boldsymbol{c} = r'\,(\cos\theta',\, \sin\theta'). \tag{1.12}$$

Then

$$u(\boldsymbol{x}) = \hat{q}_0 \, \log r + \sum_{p=1}^{\infty} \left(\hat{q}_{2p-1} \, \frac{\cos(p\,\theta)}{r^p} + \hat{q}_{2p} \frac{\sin(p\,\theta)}{r^p} \right), \tag{1.13}$$

where $\hat{q}_0$ is defined by (1.9), and where, for $p \geq 1$,

$$\hat{q}_{2p-1} = -\sum_{j=1}^{N} q_j \, (r')^p \frac{1}{p} \cos(p\,\theta'), \tag{1.14}$$

$$\hat{q}_{2p} = -\sum_{j=1}^{N} q_j \, (r')^p \frac{1}{p} \sin(p\,\theta'). \tag{1.15}$$

The sum in (1.13) converges exponentially fast whenever $|\boldsymbol{x} - \boldsymbol{c}| > \max_j |\boldsymbol{y}_j - \boldsymbol{c}|$. To be precise, we have:

THEOREM 1.1. *Let u be the function defined by (1.8), and let for a positive integer P the P-term multipole expansion be defined by*

$$u_P(\boldsymbol{x}) = \hat{q}_0 \, \log r + \sum_{p=1}^{P} \left(\hat{q}_{2p-1} \, \frac{\cos(p\,\theta)}{r^p} + \hat{q}_{2p} \frac{\sin(p\,\theta)}{r^p} \right), \tag{1.16}$$

where the multipole coefficients $\{\hat{q}_j\}_{j=0}^{2P}$ are defined via (1.9), (1.14), and (1.15). Set

$$\beta = \frac{\max_j |\boldsymbol{y}_j - \boldsymbol{c}|}{|\boldsymbol{x} - \boldsymbol{c}|}.$$

If $\beta < 1$, then

$$|u(\boldsymbol{x}) - u_P(\boldsymbol{x})| \leq \frac{\beta^{P+1}}{(P+1)(1-\beta)} \sum_{j=1}^{N} |q_j|.$$

To prove the theorem, we will switch to complex valued representations of the kernel function G. To this end, we identify a point $\boldsymbol{x} = (x_1, x_2)$ in $\mathbb{R}^2$ with a complex number $x_1 + ix_2$ as usual. Then we interpret G as a complex valued kernel

$$G(\boldsymbol{x}, \boldsymbol{y}) = \log(\boldsymbol{x} - \boldsymbol{y}),$$

and view the sum (1.8) as a complex valued sum. The real valued potentials are easily recovered from the complex valued ones by simply taking the real part of the end result since $\mathrm{Re}(\log \boldsymbol{z}) = \log |\boldsymbol{z}|$ for any non-zero complex number $\boldsymbol{z}$. Now let $\boldsymbol{x}$ and $\boldsymbol{y}$ be a target point and a source point, respectively, and let $\boldsymbol{c}$ still denote the center of Ω_σ. For some number $\beta < 1$, we assume that

$$|\boldsymbol{y} - \boldsymbol{c}| < \beta |\boldsymbol{x} - \boldsymbol{c}|.$$

Then recalling the formula $\log(1 - \boldsymbol{z}) = -\sum_{p=1}^{\infty} \frac{\boldsymbol{z}^p}{p}$, which is valid for any complex number $\boldsymbol{z}$ such that $|\boldsymbol{z}| < 1$, we immediately find the expression

$$\begin{aligned} G(\boldsymbol{x}, \boldsymbol{y}) = \log\big(\boldsymbol{x} - \boldsymbol{y}\big) &= \log\big((\boldsymbol{x} - \boldsymbol{c}) - (\boldsymbol{y} - \boldsymbol{c})\big) \\ &= \log\big(\boldsymbol{x} - \boldsymbol{c}\big) + \log\left(1 - \frac{\boldsymbol{y} - \boldsymbol{c}}{\boldsymbol{x} - \boldsymbol{c}}\right) = \log\big(\boldsymbol{x} - \boldsymbol{c}\big) - \sum_{p=1}^{\infty} \frac{1}{p} \left(\frac{\boldsymbol{y} - \boldsymbol{c}}{\boldsymbol{x} - \boldsymbol{c}}\right)^p. \end{aligned} \tag{1.17}$$

Using polar coordinates,

$$\boldsymbol{x} - \boldsymbol{c} = re^{i\theta}, \qquad \text{and} \qquad \boldsymbol{y} - \boldsymbol{c} = r'e^{i\theta'}, \tag{1.18}$$

we see that

$$\left(\frac{\boldsymbol{y} - \boldsymbol{c}}{\boldsymbol{x} - \boldsymbol{c}}\right)^p = \left(\frac{r'e^{i\theta'}}{re^{i\theta}}\right)^p = \left(\frac{r'}{r}\right)^p e^{ip(\theta' - \theta)}. \tag{1.19}$$

To recover the real-valued multipole expansion (1.13), let us insert (1.19) into (1.17), whence

$$\log |\boldsymbol{x} - \boldsymbol{y}| = \mathrm{Re}(G(\boldsymbol{x}, \boldsymbol{y})) = \log r - \sum_{p=1}^{\infty} \frac{1}{p} \left(\frac{r'}{r}\right)^p \cos(p(\theta' - \theta)).$$

We then obtain (1.13) directly from the formula

$$\cos(p(\theta' - \theta)) = \cos(p\theta') \cos(p\theta) + \sin(p\theta') \sin(p\theta).$$

The proof of Theorem 1.1 now follows from a bound of the terms that we neglect in the tail:

$$\left| \sum_{p=P+1}^{\infty} \frac{1}{p} \frac{(\boldsymbol{y} - \boldsymbol{c})^p}{(\boldsymbol{x} - \boldsymbol{c})^p} \right| \leq \frac{1}{P+1} \sum_{p=P+1}^{\infty} \left(\frac{r'}{r}\right)^p \leq \frac{1}{P+1} \sum_{p=P+1}^{\infty} \beta^p = \frac{1}{P+1} \frac{\beta^{P+1}}{1-\beta}.$$

1.3. The general problem, and well-separated points

After the introduction to multipole expansions in Sections 1.1 and 1.2, we will in the remainder of this chapter discuss the full potential evaluation problem, which we state as follows: Let $\{\boldsymbol{x}_i\}_{i=1}^N$ denote the locations of a set of electric charges in a box Ω, and let $\{q_i\}_{i=1}^N$ denote their source strengths, cf. Figure 1.3(a). Our task is then to evaluate the potentials

$$u_i = \sum_{j=1}^{N} G(\boldsymbol{x}_i, \boldsymbol{x}_j)\, q_j, \qquad i = 1,\, 2,\, \ldots,\, N. \tag{1.20}$$

In the remainder of this chapter, we will use the complex-valued kernel that was introduced in Section 1.2. For future reference, let us formulate (1.20) as a matrix-vector multiplication. We introduce a vector $\mathbf{q} \in \mathbb{C}^N$ and a matrix $\mathbf{A} \in \mathbb{C}^{N\times N}$ via

$$\mathbf{q}(i) = q_i, \qquad \text{and} \qquad \mathbf{A}(i,j) = G(\boldsymbol{x}_i,\, \boldsymbol{x}_j) \qquad i,j = 1,\, 2,\, 3,\, \ldots,\, N.$$

The summation problem (1.20) is then equivalent to evaluating the matrix-vector product

$$\mathbf{u} = \mathbf{A}\mathbf{q}. \tag{1.21}$$

Since we now have the more interesting situation where the source and the target points coincide, we must exercise some care about when to use multipole expansions, and when to evaluate sums directly. When using the FMM, we will tessellate the computational box Ω into smaller boxes, and will use expansions only for interactions between boxes that are sufficiently well separated. This concept is made precise by the following definition:

DEFINITION 1. Let Ω be a square with center $\boldsymbol{c} = (c_1,\, c_2)$ and side length $2a$. Then we say that a point $\boldsymbol{x} = (x_1,\, x_2) \in \mathbb{R}^2$ is *well-separated* from Ω if

$$\max(|x_1 - c_1|,\, |x_2 - c_2|) \geq 3a.$$

In other words, $\boldsymbol{x}$ is well-separated if and only if it lies outside a square of side-length $6a$ that is concentric with Ω, as illustrated in Figure 1.2(a).

Now suppose that a set of source points are located inside a box with sidelength $2a$, and that we use a truncated multipole expansion to evaluate potentials at some points that are well-separated from Ω, as shown in Figure 1.2(b). Then for any source point $\boldsymbol{y}_j$, we have

$$|\boldsymbol{y}_j - \boldsymbol{c}| \leq \sqrt{2}a,$$

and for any target point $\boldsymbol{x}$ we have

$$|\boldsymbol{x} - \boldsymbol{c}| \geq 3a.$$

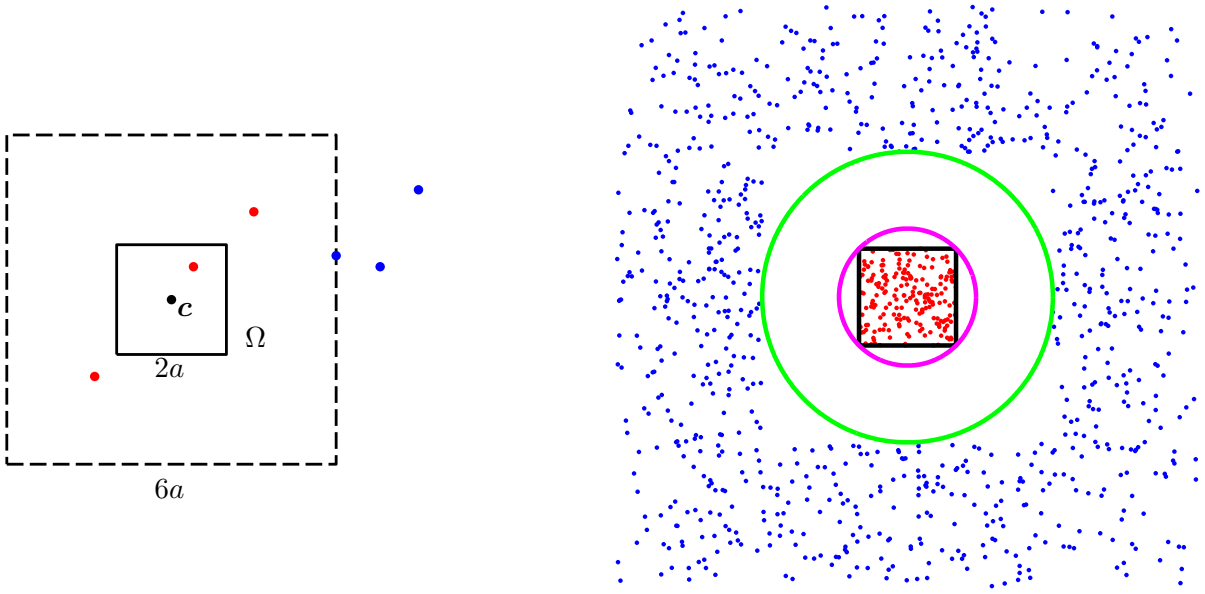
Theorem 1.1 then provides an assured rate of convergence of the multipole expansion, since

$$\frac{\max_j |\boldsymbol{y}_j - \boldsymbol{c}|}{|\boldsymbol{x} - \boldsymbol{c}|} \leq \frac{\sqrt{2}a}{3a} = \frac{\sqrt{2}}{3} =: \eta.$$

Given a requested tolerance ε, we now simply need to pick a P that is large enough that

$$\frac{\eta^P}{P(1-\eta)} \leq \varepsilon, \qquad \text{where } \eta = \sqrt{2}/3.$$

Roughly speaking, for purposes of convergence of the outgoing expansion, the choice $P \approx \frac{\log(\varepsilon)}{\log(\sqrt{2}/3)}$ would result in relative errors bounded by ε. We will in Section 1.4 see that there is a different expansion that converges slightly slower, so we will in practice have to choose $P \approx \frac{\log(\varepsilon)}{\log(\sqrt{2}/(4-\sqrt{2}))}$.



(a) (b)

FIGURE 1.2. (a) Any point on or outside of the dashed square is *well-separated* from Ω, cf. Definition 1. Blue points are well-separated from Ω. Red points are not. (b) A set of source points (red) in a box Ω. All target points (blue) are well-separated from Ω. When Ω has side-length $2a$, the magenta circle has radius $\sqrt{2}a$ and the green circle has radius $3a$.

1.4. Multipole expansions and translation operators

We start by considering a subproblem of (1.20) corresponding to the interaction between two disjoint subsets Ω_σ and Ω_τ, as illustrated in Figure 1.3(b). Specifically, we seek to evaluate the potential at all points in Ω_τ (the "target points") caused by sources in Ω_σ. To formalize, let I_σ and I_τ be index sets pointing to the locations inside each box so that, e.g.,

$$i \in I_\sigma \qquad \Leftrightarrow \qquad \boldsymbol{x}_i \in \Omega_\sigma.$$

Our task is then to evaluate the sums

$$v_i = \sum_{j \in I_\sigma} G(\boldsymbol{x}_i, \boldsymbol{x}_j)\, q_j, \qquad i \in I_\tau. \tag{1.22}$$

In matrix notation, (1.22) is equivalent to the matrix vector-product

$$\mathbf{v}^\tau = \mathbf{A}(I_\tau,\, I_\sigma)\, \mathbf{q}(I_\sigma). \tag{1.23}$$

Going forwards, it will be convenient to use complex valued multipole expansions. Following in the footsteps of the derivation in Section 1.2, let us derive a complex valued separation of variables for the kernel in (1.22). To this end, let $\boldsymbol{c}_\sigma$ and $\boldsymbol{c}_\tau$ denote the centers of Ω_σ and Ω_τ, respectively. Then, for $\boldsymbol{y} \in \Omega_\sigma$ and $\boldsymbol{x} \in \Omega_\tau$, the kernel G admits the separation of variables

$$G(\boldsymbol{x}, \boldsymbol{y}) = \log\big((\boldsymbol{x} - \boldsymbol{c}_\sigma) - (\boldsymbol{y} - \boldsymbol{c}_\sigma)\big) = \log\big(\boldsymbol{x} - \boldsymbol{c}_\sigma\big) - \sum_{p=1}^{\infty} \frac{1}{p} \frac{(\boldsymbol{y} - \boldsymbol{c}_\sigma)^p}{(\boldsymbol{x} - \boldsymbol{c}_\sigma)^p}.$$

The last expression in (1.17) is of the form (1.4) with $C_p(\boldsymbol{y}) = -\frac{1}{p}(\boldsymbol{y} - \boldsymbol{c}_\sigma)^p$ and $B_p(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{c}_\sigma)^{-p}$. We recall from Theorem 1.1 that when the sum is truncated after $P - 1$ terms, the error incurred is bounded

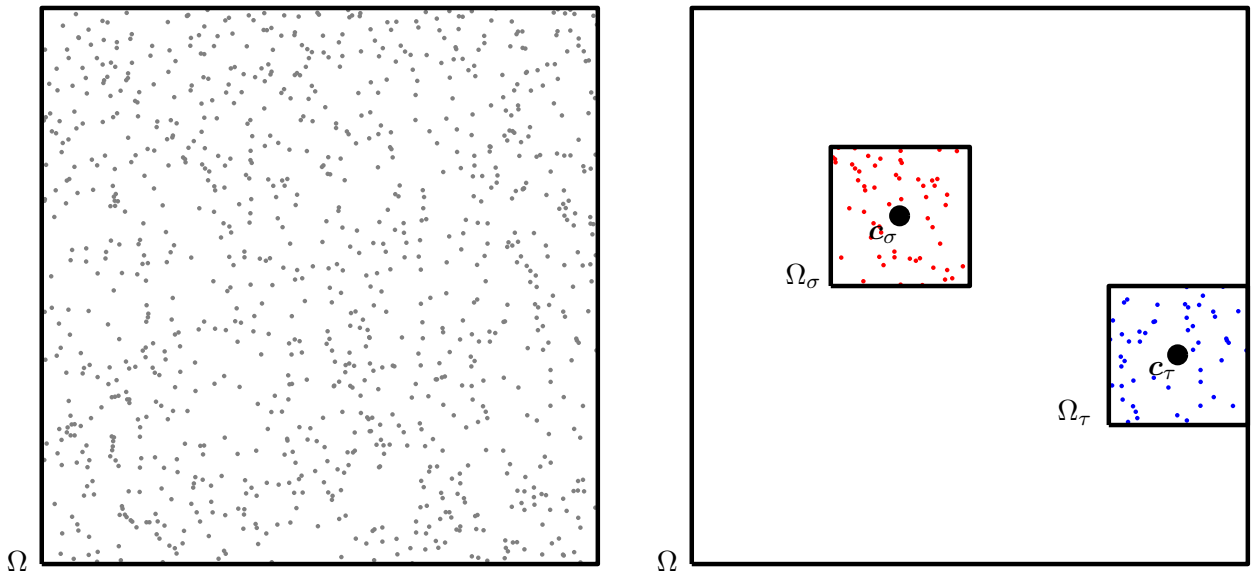


FIGURE 1.3. (a) Geometry of the full N-body problem. The domain Ω is drawn in black and the points $\boldsymbol{x}_i$ are gray. (b) The geometry described in Section 1.4. The box Ω_σ contains source locations (red) and Ω_τ contains target locations (blue).

roughly by η^P, for $\eta = \sqrt{2}/3$. The complex valued *outgoing expansion* of Ω_σ of σ, is now defined as the vector $\hat{\mathbf{q}}^\sigma = \{\hat{q}_p^\sigma\}_{p=0}^{P-1}$ where

$$\begin{cases} \hat{q}_0^\sigma = \displaystyle\sum_{j \in I_\sigma} q_j \\ \hat{q}_p^\sigma = \displaystyle\sum_{j \in I_\sigma} -\frac{1}{p} (\boldsymbol{x}_j - \boldsymbol{c}_\sigma)^p \, q_j, \qquad p = 1,\, 2,\, 3,\, \ldots,\, P-1. \end{cases} \tag{1.24}$$

The vector $\hat{\mathbf{q}}^\sigma$ is a compact representation of the sources in Ω_σ. It contains all information needed to evaluate the field $v(\boldsymbol{x}) = \sum_{j \in I_\sigma} G(\boldsymbol{x}, \boldsymbol{x}_j)\, q_j$ when $\boldsymbol{x}$ is a point "far away" from Ω_σ.

It turns out to be convenient to also define an *incoming expansion* for Ω_τ. The basic idea here is that for $\boldsymbol{x} \in \Omega_\tau$, the potential

$$v(\boldsymbol{x}) = \sum_{j \in I_\sigma} G(\boldsymbol{x}, \boldsymbol{x}_j)\, q_j = \log(\boldsymbol{x} - \boldsymbol{c}_\sigma)\, \hat{q}_0^\sigma + \sum_{p=1}^{\infty} \frac{1}{(\boldsymbol{x} - \boldsymbol{c}_\sigma)^p}\, \hat{q}_p^\sigma \tag{1.25}$$

is a harmonic function on Ω_τ. In consequence, it has a convergent expansion

$$v(\boldsymbol{x}) = \sum_{p=0}^{\infty} (\boldsymbol{x} - \boldsymbol{c}_\tau)^p \, \hat{v}_p^\tau.$$

The complex numbers $\{\hat{v}_p^\tau\}_{p=0}^\infty$ can be obtained from the numbers $\{\hat{q}_p\}_{p=0}^\infty$ via the formulas

$$\begin{cases} \hat{v}_0^\tau = \hat{q}_0^\sigma \log(\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma) + \displaystyle\sum_{p=1}^{\infty} \hat{q}_p^\sigma \, (-1)^p \, \frac{1}{(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^p}, \\ \hat{v}_r^\tau = -\hat{q}_0^\sigma \dfrac{1}{r(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^r} + \displaystyle\sum_{p=1}^{\infty} \hat{q}_p^\sigma (-1)^p \binom{r+p-1}{p-1} \frac{1}{(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{r+p}}. \end{cases} \tag{1.26}$$

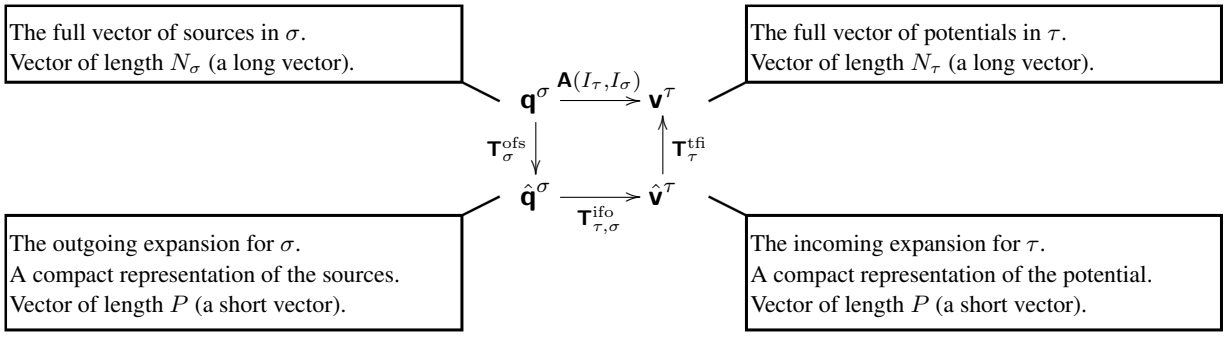


FIGURE 1.4. The outgoing and incoming expansions introduced in Section 1.4 are compact representations of the sources and potentials in the source and target boxes, respectively. The diagram commutes to high precision since $\mathbf{A}(I_\tau, I_\sigma) \approx \mathbf{T}_\tau^{\mathrm{tfi}} \mathbf{T}_{\tau,\sigma}^{\mathrm{ifo}} \mathbf{T}_\sigma^{\mathrm{ofs}}$, cf. (1.27).

We provide a derivation of these formulas in Remark 1.2. Upon truncation of the expansion coefficients, we get the vector $\hat{\mathbf{v}}^\tau = \{\hat{v}_p^\tau\}_{p=0}^{P-1}$, which we refer to as the *incoming expansion* for Ω_τ generated by the sources in Ω_σ. It is a compact (approximate) representation of the harmonic field v defined by (1.25).

The linear maps introduced in this section can advantageously be represented via matrices that we refer to as *translation operators*. Let N_σ and N_τ denote the number of points in Ω_σ and Ω_τ, respectively. The map (1.24) can then upon truncation be written

$$\hat{\mathbf{q}}^\sigma = \mathbf{T}_\sigma^{\mathrm{ofs}} \, \mathbf{q}(I_\sigma),$$

where $\mathbf{T}_\sigma^{\mathrm{ofs}}$ is a $P \times N_\sigma$ matrix called the *outgoing-from-sources* translation operator with the entries implied by (1.24). Analogously, (1.26) can upon truncation be written $\hat{\mathbf{v}}^\tau = \mathbf{T}_{\tau,\sigma}^{\mathrm{ifo}} \, \hat{\mathbf{q}}^\sigma$, where $\mathbf{T}_{\tau,\sigma}^{\mathrm{ifo}}$ is the *incoming-from-outgoing* translation operator. Finally, the *targets-from-incoming* translation operator is the matrix $\mathbf{T}_\tau^{\mathrm{tfi}}$ such that $\mathbf{v}^\tau = \mathbf{T}_\tau^{\mathrm{tfi}} \hat{\mathbf{v}}^\tau$, where $\mathbf{v}^\tau$ is an approximation to the field v defined by (1.25), in other words $\mathbf{T}_\tau^{\mathrm{tfi}}(i,p) = (\boldsymbol{x}_i - \boldsymbol{c}_\tau)^{p-1}$. These three translation operators are factors in an approximate rank-P factorization

$$\begin{array}{lccccc} \mathbf{A}(I_\tau, I_\sigma) & \approx & \mathbf{T}_\tau^{\mathrm{tfi}} & \mathbf{T}_{\tau,\sigma}^{\mathrm{ifo}} & \mathbf{T}_\sigma^{\mathrm{ofs}}. \\ N_\tau \times N_\sigma & & N_\tau \times P & P \times P & P \times N_\sigma \end{array} \tag{1.27}$$

A diagram illustrating the factorization (1.27) is shown in Figure 1.4.

The introduction of the incoming expansion unfortunately reduces the rate of convergence of the expansions that we use. Now both the local expansion and the multipole expansions must converge in discs of radius $\sqrt{2}a$, which reduces the rate of convergence of the error ε to

$$\varepsilon \sim \eta_{2\mathrm{D}}^P, \qquad \text{where } \eta_{2\mathrm{D}} = \frac{\sqrt{2}}{4 - \sqrt{2}}.$$

REMARK 1.1. *The terms "outgoing expansion" and "incoming expansion" are slightly unconventional. The corresponding objects were in the original papers called "multipole expansion" and "local expansion," and these terms continue to be commonly used; even in summation schemes where the expansions have nothing to do with multipoles. Correspondingly, what we call the "incoming-from-outgoing" translation operator is often called the "multipole-to-local" or "M2L" operator.*

REMARK 1.2. *In this remark, we demonstrate how to obtain a formula such as (1.26). Let $\boldsymbol{c}_\sigma$ and $\boldsymbol{c}_\tau$ be two given points in the complex plane. Let v be a function with a MacLaurin expansion about $\boldsymbol{c}_\sigma$*

$$v(\boldsymbol{x}) = \log(\boldsymbol{x} - \boldsymbol{c}_\sigma) \, \hat{q}_0^\sigma + \sum_{p=1}^{\infty} \frac{1}{(\boldsymbol{x} - \boldsymbol{c}_\sigma)^p} \, \hat{q}_p^\sigma.$$

If the series converges at $\boldsymbol{c}_\tau$, then its Maclaurin expansion $v(\boldsymbol{x}) = \sum_{r=0}^{\infty} \hat{v}_r^\tau (\boldsymbol{x} - \boldsymbol{c}_\tau)^r$, has coefficients

$$\begin{cases} \hat{v}_0^\tau = \hat{q}_0^\sigma \log(\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma) + \sum_{p=1}^{\infty} \hat{q}_p^\sigma \, (-1)^p \, \dfrac{1}{(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^p}, \\ \hat{v}_r^\tau = -\hat{q}_0^\sigma \dfrac{1}{r(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^r} + \sum_{p=1}^{\infty} \hat{q}_p^\sigma (-1)^p \dbinom{r+p-1}{p-1} \dfrac{1}{(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{r+p}}. \end{cases} \tag{1.28}$$

To derive this expression, we first use the formula $\log(1-t) = -\sum_{r=1}^{\infty} t^r/r$, valid for $|t| < 1$:

$$\begin{aligned} \log(\boldsymbol{x} - \boldsymbol{c}_\sigma) &= \log\big((\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma) - (\boldsymbol{c}_\tau - \boldsymbol{x})\big) = \log(\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma) + \log\left(1 - \frac{\boldsymbol{c}_\tau - \boldsymbol{x}}{\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma}\right) \\ &= \log(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau) - \sum_{r=1}^{\infty} \frac{1}{r} \left(\frac{\boldsymbol{c}_\tau - \boldsymbol{x}}{\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma}\right)^r = \log(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau) - \sum_{r=1}^{\infty} \frac{1}{r} \left(\frac{1}{\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau}\right)^r (\boldsymbol{x} - \boldsymbol{c}_\tau)^r. \end{aligned}$$

Next let us use the formula $(1-t)^{-p} = \sum_{r=0}^{\infty} \binom{p+r-1}{r-1} t^r$, which is also valid for $|t| < 1$, to get

$$\begin{aligned} (\boldsymbol{x} - \boldsymbol{c}_\sigma)^{-p} &= \big((\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma) - (\boldsymbol{c}_\tau - \boldsymbol{x})\big)^{-p} = (\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma)^{-p} \left(1 - \frac{\boldsymbol{x} - \boldsymbol{c}_\tau}{\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau}\right)^{-p} \\ &= (\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma)^{-p} \sum_{r=0}^{\infty} \binom{p+r-1}{r-1} \left(\frac{\boldsymbol{x} - \boldsymbol{c}_\tau}{\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau}\right)^r = \sum_{r=0}^{\infty} (-1)^p \binom{p+r-1}{r-1} \frac{1}{(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{r+p}} (\boldsymbol{x} - \boldsymbol{c}_\tau)^r. \end{aligned}$$

1.5. A single-level method

Having introduced the concepts of incoming and outgoing expansions in Section 1.4, let us next demonstrate how to evaluate our basic summation problem (1.1). Before we tackle the full multilevel hierarchical scheme that forms the actual FMM, let us ease into things with a single-level method that does not achieve $O(N)$ complexity but that will help familiarize us with the core concepts and with the notation used.

Sub-divide the box Ω into a grid of m equisized smaller boxes $\{\Omega_\tau\}_{\tau=1}^m$ as shown in Figure 1.5(a). As in Section 1.4, we let for each box τ the index vector I_τ list the points inside Ω_τ and let $\boldsymbol{c}_\tau$ denote the center of τ. The vector $\hat{\mathbf{q}}^\tau$ denotes the *outgoing expansion* of τ, as defined by (1.24).

For a box τ, let $\mathcal{L}_\tau^{\text{nei}}$ denote the list of *neighbor boxes*; these are the boxes that directly touch τ; there will be between 3 and 8 of them, depending on where τ is located in the grid. The remaining boxes are collected in the list of *far-field boxes* $\mathcal{L}_\tau^{\text{far}}$. Figure 1.5(b) illustrates the lists.

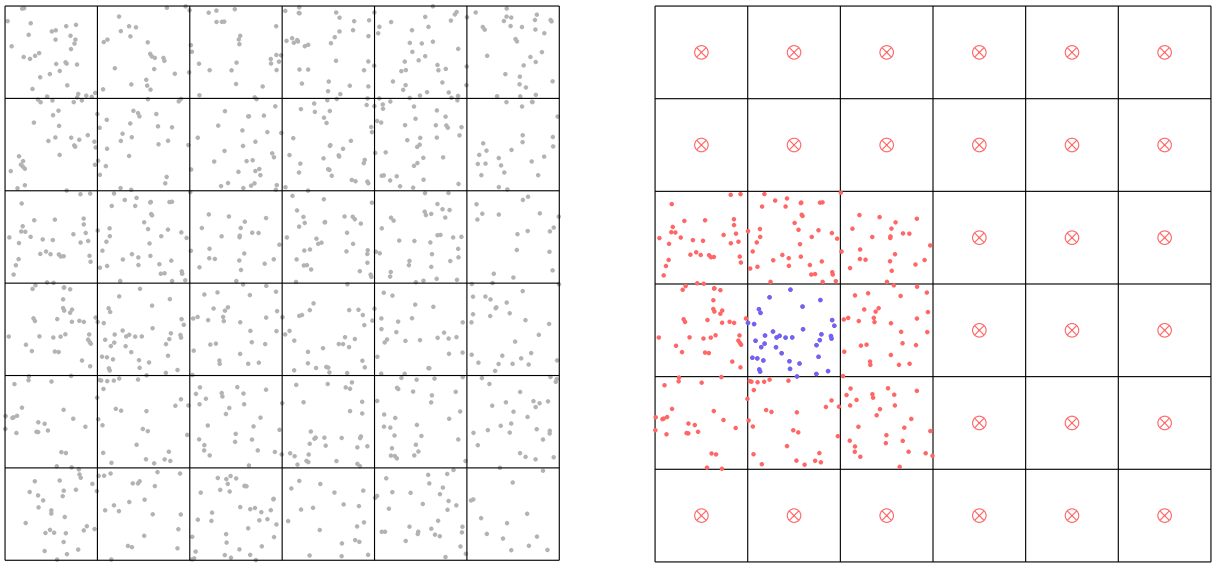
The sum (1.1) can now be approximated via three steps:

(1) *Compute the outgoing expansions:* Loop over all boxes τ. For each box, compute its outgoing expansion $\hat{\mathbf{q}}^\tau$ via the *outgoing-from-sources* translation operator:

$$\hat{\mathbf{q}}^\tau = \mathbf{T}_\tau^{\text{ofs}} \, \mathbf{q}(I_\tau).$$

(2) *Convert outgoing expansions to incoming expansions:* Loop over all boxes τ. For each box, construct a vector $\hat{\mathbf{u}}^\tau$ called the *incoming expansion*. It represents the contribution to the potential in τ from sources in all boxes in the far-field of τ and is given by

$$\hat{\mathbf{u}}^\tau = \sum_{\sigma \in \mathcal{L}_\tau^{\text{far}}} \mathbf{T}_{\tau,\sigma}^{\text{ifo}} \, \hat{\mathbf{q}}^\sigma.$$



(a) (b)

FIGURE 1.5. (a) A tessellation of Ω into $\sqrt{m} \times \sqrt{m}$ smaller boxes, cf. Section 1.5. (b) Evaluation of the potential in a box τ. The target points in τ are marked with blue dots, the source points in the neighbor boxes in $\mathcal{L}_\tau^{\text{nei}}$ are marked with red dots, and the centers of the outgoing expansions in the far-field boxes $\mathcal{L}_\tau^{\text{far}}$ are marked $\otimes$.

(3) *Compute near interactions:* Loop over all boxes τ. Expand the incoming expansion and add the contributions from its neighbors via direct summation:

$$\mathbf{u}(I_\tau) = \mathbf{T}_\tau^{\text{tfi}}\, \hat{\mathbf{u}}^\tau + \mathbf{A}(I_\tau, I_\tau)\, \mathbf{q}(I_\tau) + \sum_{\sigma \in \mathcal{L}_\tau^{\text{nei}}} \mathbf{A}(I_\tau, I_\sigma)\, \mathbf{q}(I_\sigma).$$

The asymptotic complexity of Step 1 is $O(NP)$ since each of the N points contributes to P multipole coefficients. The cost of steps two and three depend on how the number m of boxes is picked. If m is large, then Step 2 becomes expensive since there are many boxes that all need to talk to each other. On the other hand, the larger m is, the cheaper Step 3 becomes since the number of particles in one box decreases as the number of boxes increases. To be precise, if T_i denotes the cost of Step i, then it is easy to see that

$$T_1 \sim NP, \qquad T_2 \sim m^2 P^2, \qquad \text{and} \qquad T_3 \sim NP + \frac{N^2}{m}. \tag{1.29}$$

The total cost $T = T_1 + T_2 + T_3$ is minimized for $m \sim N^{2/3}$, which leads to an overall complexity of $O(N^{4/3})$.

REMARK 1.3 (Rank-structured matrix interpretation). *The single level scheme we describe in this section can readily be interpreted as a particularly simple form of a rank-structured matrix. If we create the index vector $I = [I_1, I_2, \ldots, I_m]$, then the matrix $\mathbf{A}(I, I)$ will be an $m \times m$ block matrix, as shown in Figure 1.6. On each block row, there will be at most 9 blocks that correspond to interactions that cannot be compressed, and must be represented as full dense matrices. The remaining $O(m)$ blocks on each row represent low rank interactions that can be accounted for via the $P \times P$ translation operators that map outgoing to incoming expansions. Since there are m block rows, and each row holds $O(1)$ matrices of size roughly $N/m \times N/m$, and $O(m)$ matrices of size $P \times P$, we find a total cost of*

$$T \sim m \times \left(\left(\frac{N}{m} \right)^2 + mP^2 \right) \sim \frac{N^2}{m} + m^2 P^2.$$

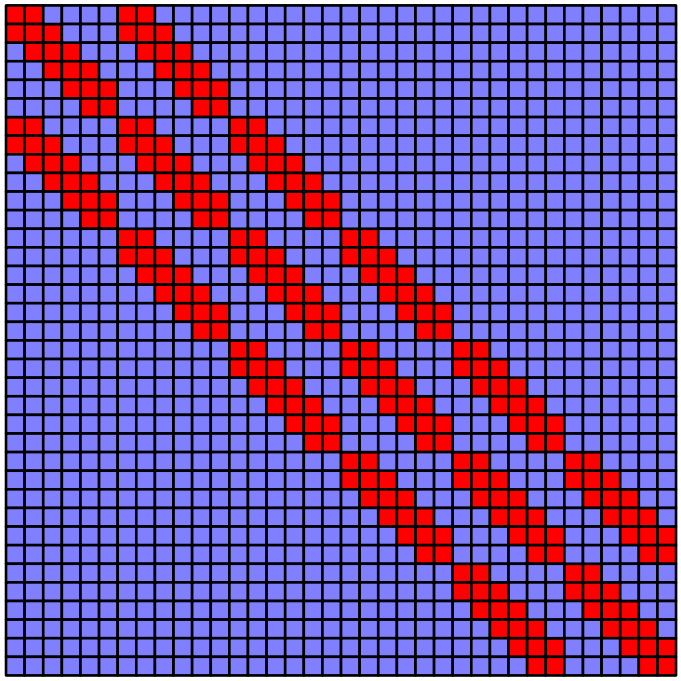


FIGURE 1.6. The $N \times N$ matrix $\mathbf{A}(I, I)$ described in Remark 1.3, which is made up of $m \times m$ blocks. The figure shows the matrix associated with the tessellation in Figure 1.5(a) when the blocks are ordered column by column. Red block are stored as dense matrices of size roughly $N/m \times N/m$. Blue blocks are all of (numerical) rank P, and can be stored by simply storing the corresponding $P \times P$ translation operators.

This again is minimized for $m \sim N^{2/3}$, which yields $T \sim N^{4/3}$.

Exercises

(1) At the end of Section 1.5, we claim that the overall complexity of the single-level scheme is $O(N^{4/3})$ if the number of boxes m is chosen to satisfy $m \sim N^{2/3}$. Work out the details that establish the formulas (1.29), and then prove the claim that the optimal choice is $m \sim N^{2/3}$. (In this exercise, we treat P as a fixed number.)

CHAPTER 2

The Fast Multipole Method (FMM)

In this chapter, we continue the discussion from Chapter 1 on how to evaluate the sum

$$u_i = \sum_{j=1}^{N} G(\boldsymbol{x}_i, \boldsymbol{x}_j)\, q_j, \qquad i = 1,\, 2,\, \ldots,\, N, \tag{2.1}$$

where $\{\boldsymbol{x}_i\}_{i=1}^N$ is a given set of point locations in a square Ω, where $\{q_i\}_{i=1}^N$ is a given set of corresponding sources, where $\{u_i\}_{i=1}^N$ is a set of potential values to be determined, and where $G(\boldsymbol{x}, \boldsymbol{y}) = \log|\boldsymbol{x} - \boldsymbol{y}|$. To evaluate (2.1 in linear complexity, the FMM uses a multi-level technique in which the computational domain Ω is split into a tree of boxes, cf. Figure 2.1. The FMM works well for adaptive and non-uniform trees, but for notational simplicity, we will in this chapter restrict attention to the case of a uniform tree in which every level is fully populated. The FMM then evaluates the sum (1.1) in two passes over the tree, one going upwards (from smaller boxes to larger) and one going downwards:

The upwards pass: In the upwards pass, the outgoing expansion is computed for every box. For a leaf box τ, the straight-forward approach described in Section 1.4 is used. For a box τ that has children, the outgoing expansion is computed not directly from the sources located in the box, but from the outgoing expansions of its children, which are already available.

The downwards pass: In the downwards pass, the incoming expansion is computed for every box. This is done by converting the outgoing expansions constructed in the upwards pass to incoming expansions via the formula (1.26). The trick is to organize the computation so that each conversion happens at its appropriate length-scale. This hierarchical process computes the incoming expansions for every box in $O(N)$ operations.

Once the upwards and downwards passes have been completed, the incoming expansion is known for all leaf boxes. All that remains at that point is to expand the incoming expansion into potentials and adding the contributions from sources in the near-field via direct computations.

In order to describe the details of how to execute the upwards and downwards passes, we need to introduce some language for describing the hierarchical tree of boxes (Section 2.1), and some additional translation operators in Sections 2.3 and 2.4.

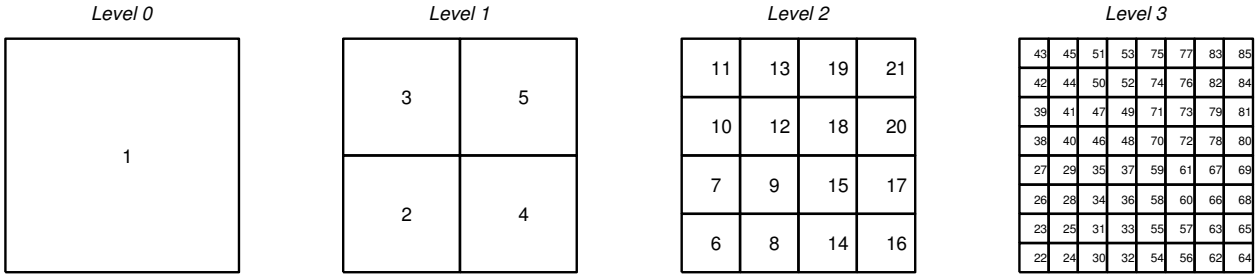
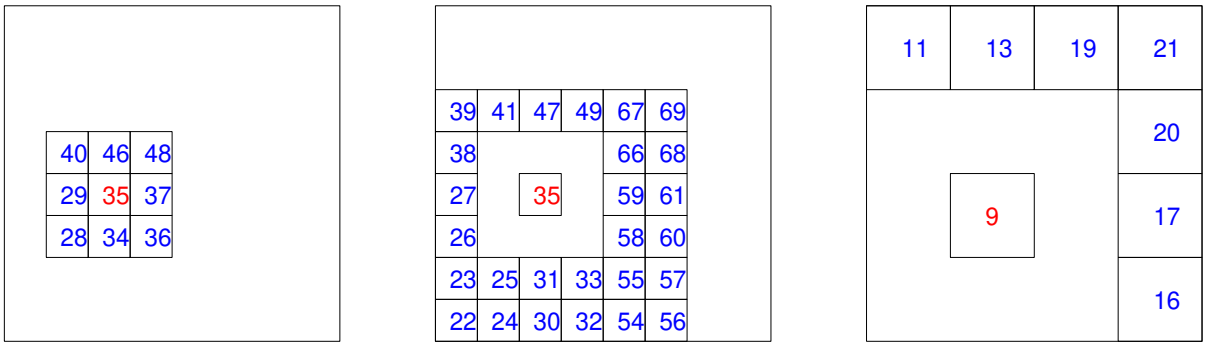


FIGURE 2.1. A tree of boxes on Ω with $L = 3$ levels. The enumeration of boxes shown is simply one of many possible ones.



Neighbor list of $\tau = 35$ *Interaction list of* $\tau = 35$ *Interaction list of* $\tau = 9$

FIGURE 2.2. Illustration of some index vectors called "lists" that were introduced in Section 2.1. For instance, the left-most figure illustrates that $\mathcal{L}_{35}^{\text{nei}} = \{28, 29, 34, 36, 37, 40, 46, 48\}$. (Boxes are numbered as in Figure 2.1.)

2.1. A tree of boxes

Split the square Ω into 4^L equisized smaller boxes, where the integer L is chosen to be large enough that each box holds only a small number of points. (The optimal number of points to keep in a box depends on many factors, but having $\sim 10^2$ points per box is often reasonable.) These 4^L equisized small boxes form the *leaf boxes* of the tree. We merge the leaves by sets of four to form 4^{L-1} boxes of twice the side-length, and then continue merging by sets of four until we recover the original box Ω, which we call the *root*. The set consisting of all boxes of the same size forms what we call a *level*. We label the levels using the integers $\ell = 0, 1, 2, \ldots, L$, with $\ell = 0$ denoting the root, and $\ell = L$ denoting the leaves, cf. Figure 2.1.

Given a box τ in the hierarchical tree, we next define some index lists, cf. Figure 2.2:

- The parent of τ is the box on the next coarser level that contains τ.
- The children of τ is the set $\mathcal{L}_\tau^{\text{child}}$ of boxes whose parent is τ.
- The neighbors of τ is the set $\mathcal{L}_\tau^{\text{nei}}$ of boxes on the same level that directly touch τ.
- The interaction list of τ is the set $\mathcal{L}_\tau^{\text{int}}$ of all boxes σ such that (1) σ and τ are on the same level, (2) σ and τ do not touch, and (3) the parents of σ and τ do touch.

When non-uniform trees are considered, additional lists will be required, as we will discuss in Section 3.1.

2.2. Outgoing and incoming expansions

With each box in the tree, we associate two vectors that we call the *outgoing expansion* and the *incoming expansion*, respectively. The outgoing expansion is a compact representation of the field generated by all the sources that are contained in the box. The incoming expansion is a compact representation of the potential in the box that is generated by all sources that are well-separated from the box. These vectors are of length P, where P is a tuning parameter that trades computational cost for accuracy. For a fixed P, we define for a box Ω_τ in the tree these expansions as follows:

The outgoing expansion $\hat{\mathbf{q}}_\tau \in \mathbb{C}^P$*:* Let I_τ denote a list of all the points $\boldsymbol{x}_j$ such that $\boldsymbol{x}_j \in \Omega_\tau$. Then the entries of the outgoing expansion are the multipole coefficients defined via the formula (1.24), which we

repeat for completeness:

$$\begin{cases} \hat{q}_0^\tau = \displaystyle\sum_{j \in I_\tau} q_j \\ \hat{q}_p^\tau = \displaystyle\sum_{j \in I_\tau} -\frac{1}{p} \big(\boldsymbol{x}_j - \boldsymbol{c}_\tau\big)^p \, q_j, \qquad p = 1,\, 2,\, 3,\, \ldots,\, P-1, \end{cases} \tag{2.2}$$

The incoming expansion $\hat{\mathbf{q}}_\tau \in \mathbb{C}^P$*:* Let J_τ denote a list of all points $\boldsymbol{x}_j$ that are well-separated from Ω_τ. The incoming potential u_τ is then defined as the field generated by these sources:

$$u_\tau(\boldsymbol{x}) = \sum_{j \in J_\tau} q_j \log |\boldsymbol{x} - \boldsymbol{x}_j|.$$

This field is analytic around the center $\boldsymbol{c}_\tau$ of Ω_τ, and the incoming expansion vector $\hat{\mathbf{u}}_\tau$ holds the first P coefficients in the Taylor series for u_τ around $\boldsymbol{c}_\tau$.

2.3. Computing all outgoing expansions

As our first task, we will for each box τ in the tree build its outgoing expansion $\hat{\mathbf{q}}_\tau$, as defined by (2.2). We start by processing all leaf boxes, and simply use formula (2.2) directly, which we recall is encoded in the matrix operation

$$\hat{\mathbf{q}}_\tau = \mathbf{T}_\tau^{\text{ofs}} \mathbf{q}(I_\tau).$$

Next let us consider a box τ on level $L-1$, meaning that it is formed by four leaf boxes. We could now of course simply use (2.2) again. But this would be wasteful, since the outgoing expansions for the four leaves already provide us with economic representations of all the sources inside τ. It turns out to be straightforward to compute the outgoing representation for τ from the outgoing representations from its children via simple linear transformations. To be precise, there exist matrices $\mathbf{T}_{\tau,\sigma}^{\text{ofo}}$, known as *outgoing-from-outgoing translation operators* such that

$$\hat{\mathbf{q}}_\tau = \sum_{\sigma \in \mathcal{L}_\tau^{\text{child}}} \mathbf{T}_{\tau,\sigma}^{\text{ofo}} \hat{\mathbf{q}}_\tau.$$

This means that after we have processed the leaf boxes, we can compute the outgoing expansion for every other box by four matrix-vector multiplications involving matrices of size $P \times P$. Additional details on how to build the matrices are given in Theorem 2.1 below, and the whole process is illustrated in Figure 2.3.

THEOREM 2.1. *Let $\boldsymbol{c}_\tau$ and $\boldsymbol{c}_\sigma$ denote two points in the complex plane. Let $\{\boldsymbol{y}_j\}_{j=1}^N$ denote a number of source locations with associated charges $\{q_j\}_{j=1}^N$. Define for $p = 0,\, 1,\, 2,\, \ldots,\, P-1$ the "outgoing expansions" $\hat{\mathbf{q}}_\tau$ and $\hat{\mathbf{q}}_\sigma$ via*

$$\hat{\mathbf{q}}_\tau(p) = \sum_{j=1}^N q_j \, (\boldsymbol{y}_j - \boldsymbol{c}_\tau)^p, \qquad \textit{and} \qquad \hat{\mathbf{q}}_\sigma(p) = \sum_{j=1}^N q_j \, (\boldsymbol{y}_j - \boldsymbol{c}_\sigma)^p.$$

Let $\mathbf{T}_{\tau,\sigma}^{\text{ofo}}$ denote the $P \times P$ lower triangular matrix defined by

$$\mathbf{T}_{\tau,\sigma}^{\text{ofo}}(r,s) = \begin{cases} \binom{r}{s} \, (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{r-s}, & \textit{when } s \le r, \\ 0 & \textit{when } s > r. \end{cases}$$

Then

$$\hat{\mathbf{q}}_\tau = \mathbf{T}_{\tau,\sigma}^{\text{ofo}} \, \hat{\mathbf{q}}_\sigma. \tag{2.3}$$

THEOREM 2.2. *Bello.*

Consider a set four leaf boxes $\sigma_1, \sigma_2, \sigma_3$, and σ_4, whose union forms a parent box $\Omega_\tau = \Omega_{\sigma_1} \cup \Omega_{\sigma_2} \cup \Omega_{\sigma_3} \cup \Omega_{\sigma_4}$. We seek to build the outgoing potentials for the four leaf boxes and for the parent box.

We first process each of the four leaf boxes. For each box, we compute the outgoing expansion directly from the point sources, using the formula $\hat{\mathbf{q}}_{\sigma_j} = \mathbf{T}^{\text{ofs}}_{\sigma_j} \mathbf{q}(I_{\sigma_j})$.

In order to compute the outgoing expansion for the parent box τ, we simply shift the centers for the expansions of the children (cf. Theorem 2.1) and add them up: $\hat{\mathbf{q}}_\tau = \sum_{\sigma \in \mathcal{L}^{\text{child}}_\tau} \mathbf{T}^{\text{ofo}}_{\tau,\sigma} \hat{\mathbf{q}}_\sigma$.

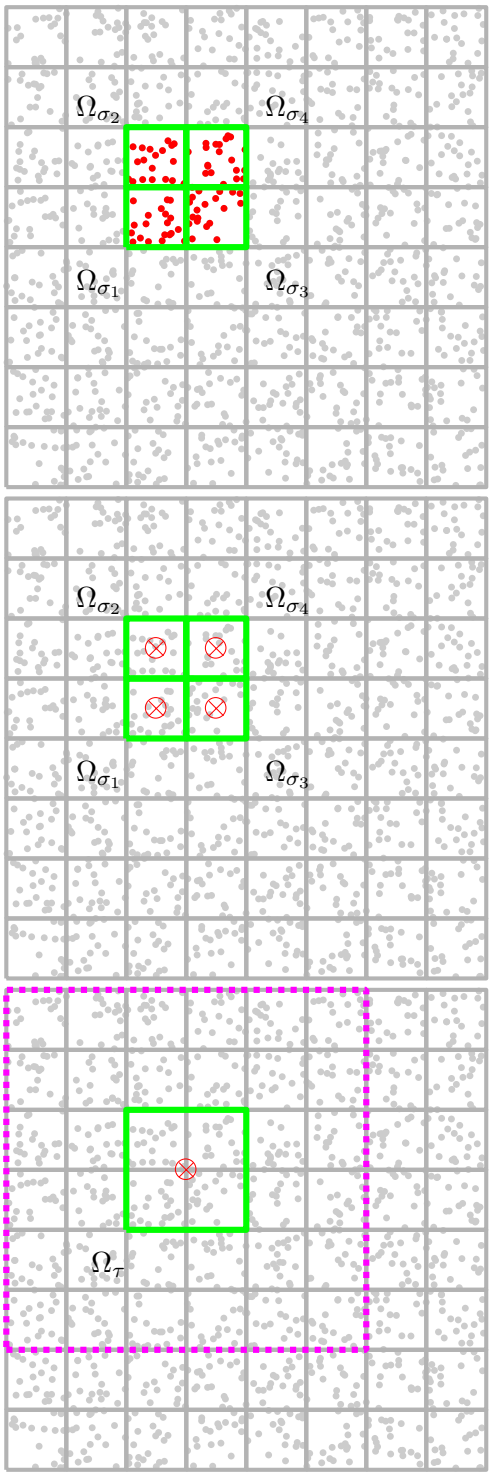


FIGURE 2.3. Illustration of how the outgoing expansions are computed. Observe that the outgoing expansions of the parent box Ω_τ should be valid outside the magenta line. Any point outside this line is well-separated from each child box.

PROOF. The equation (2.3) follows directly from the binomial identity:

$$\hat{\mathbf{q}}_\tau(r) = \sum_{j=1}^{N} q_j \left(\boldsymbol{y}_j - \boldsymbol{c}_\tau\right)^r = \sum_{j=1}^{N} q_j \left[\left(\boldsymbol{y}_j - \boldsymbol{c}_\sigma\right) + \left(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau\right)\right]^r$$

$$= \sum_{j=1}^{N} q_j \sum_{s=0}^{r} \binom{r}{s} \left(\boldsymbol{y}_j - \boldsymbol{c}_\sigma\right)^s \left(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau\right)^{r-s} = \sum_{s=0}^{r} \binom{r}{s} (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{r-s} \underbrace{\sum_{j=1}^{N} q_j (\boldsymbol{y}_j - \boldsymbol{c}_\sigma)^s}_{=\hat{\mathbf{q}}_\sigma(s)}.$$

□

2.4. Computing all incoming expansions

The incoming expansions for all boxes are built in a top-down sweep through the tree of boxes, going from larger to smaller.

All boxes at levels $\ell = 0$ and $\ell = 1$ have zero incoming expansions, so the first non-trivial ones arise on level $\ell = 2$. Let τ be a box on level 2, as shown in Figure 2.4(a). We see that all sources that contribute to the incoming expansion for τ belong to boxes in the interaction list of τ. We can therefore easily build the incoming expansion for τ by exploiting the outgoing expansions for all of these boxes, so

$$\hat{\mathbf{u}}_\tau = \sum_{\sigma \in \mathcal{L}_\tau} \mathbf{T}^{\text{ifo}}_{\tau,\sigma} \hat{\mathbf{q}}_\sigma. \tag{2.4}$$

Next let us consider a box σ on level $\ell = 3$, as shown in Figure 2.4(c). Let u_σ denote the incoming field on σ. We split this field into two components

$$u_\sigma = u_\sigma^{(1)} + u_\sigma^{(2)},$$

where $u_\sigma^{(1)}$ is the field caused by sources that are well-separated *from the parent τ of σ*, while $u_\sigma^{(2)}$ is the field generated by the sources that are well-separated from σ, but are *not* well-separated from the parent τ. Now observe that $u_\sigma^{(1)}$ is well-represented inside Ω_σ by the incoming expansion for the parent τ. All we need to do is to shift the expansion center, which corresponds to a linear map $\hat{\mathbf{u}}_\sigma = \mathbf{T}^{\text{ifi}}_{\sigma,\tau}$ whose coefficient matrix we refer to as the *incoming-from-incoming* translation operator. (A precise formulation is given in Theorem 2.1.) The field $u_\sigma^{(2)}$ can be generated by adding up the contributions from the outgoing expansions of boxes in the interaction list of σ. Consequently

$$\hat{\mathbf{u}}_\sigma = \mathbf{T}^{\text{ifi}}_{\sigma,\tau} \hat{\mathbf{u}}_\tau + \sum_{\nu \in \mathcal{L}_\sigma} \mathbf{T}^{\text{ifo}}_{\sigma,\nu} \hat{\mathbf{q}}_\nu. \tag{2.5}$$

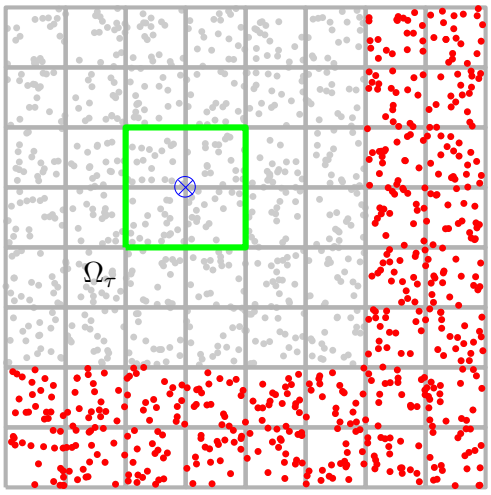
As we proceed down the tree, we use the formula (2.5) to build the incoming expansions for all boxes. (Observe that the formula (2.5) also works for the boxes at level $\ell = 2$ since for any such box, its parent has zero incoming field.)

THEOREM 2.1. *Let $\mathbf{c}_\sigma$ and $\mathbf{c}_\tau$ be two points in the complex plane. Let u be a function that is analytic at $\mathbf{c}_\sigma$ and at $\mathbf{c}_\tau$, and suppose that each point lies within the radius of analyticity of the other. We let $\{\hat{u}_p^\sigma\}_{p=0}^\infty$ and $\{\hat{u}_p^\tau\}_{p=0}^\infty$ denote the expansion coefficients for u about the two points, so that locally*

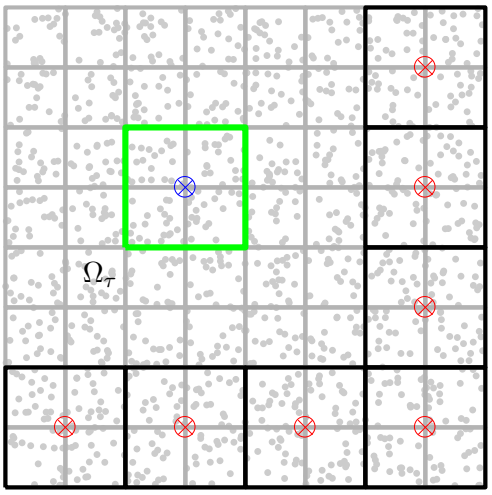
$$u(\boldsymbol{x}) = \sum_{p=0}^{\infty} \hat{u}_p^\sigma \left(\boldsymbol{x} - \boldsymbol{c}_\sigma\right)^p, \qquad \textit{and} \qquad u(\boldsymbol{x}) = \sum_{p=0}^{\infty} \hat{u}_p^\tau \left(\boldsymbol{x} - \boldsymbol{c}_\tau\right)^p.$$

Then

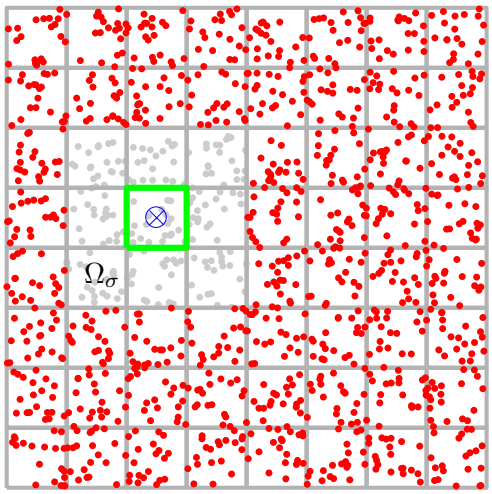
$$\hat{u}_r^\sigma = \sum_{p=r}^{\infty} \binom{p}{r} (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{p-r} \hat{u}_p^\tau.$$



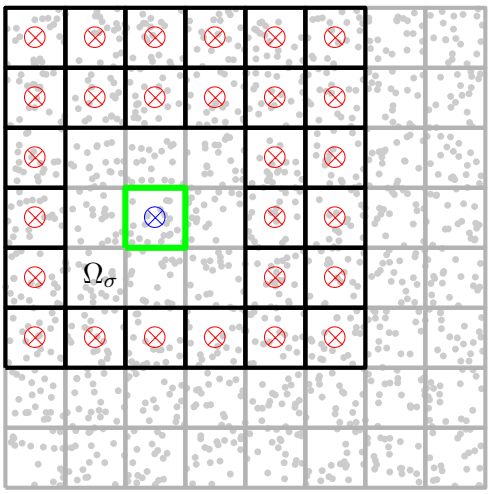
(a) We seek to compute the incoming expansion $\hat{\mathbf{u}}_\tau$ for box Ω_τ on level $\ell = 2$. Ω_τ is marked in green. $\hat{\mathbf{u}}_\tau$ is the potential caused by the sources marked in red.



(b) Instead of evaluating $\hat{\mathbf{u}}_\tau$ directly from the sources, we use the outgoing expansions for the boxes in the interaction list, $\hat{\mathbf{u}}_\tau = \sum_{\sigma \in \mathcal{L}_\tau} \mathbf{T}^{\text{ifo}}_{\tau,\sigma} \hat{\mathbf{q}}_\sigma$.



(c) Now let us build the incoming expansion for a box Ω_σ (green) that is a child of Ω_τ. $\hat{\mathbf{u}}_\sigma$ is the potential caused by the sources marked in red.



(d) Contributions to $\hat{\mathbf{u}}_\sigma$ come in two types: (1) Sources that are well-separated from the parent τ are communicated by shifting $\hat{\mathbf{u}}_\tau$. (2) Sources that are in the interaction list of σ send their contributions via their outgoing expansions. In all: $\hat{\mathbf{u}}_\sigma = \mathbf{T}^{\text{ifi}}_{\sigma,\tau} \hat{\mathbf{u}}_\tau + \sum_{\nu \in \mathcal{L}_\sigma} \mathbf{T}^{\text{ifo}}_{\sigma,\nu} \hat{\mathbf{q}}_\nu$.

FIGURE 2.4. Illustration of how the incoming expansions are computed.

PROOF. A simple application of the binomial identity yields

$$u(\boldsymbol{x}) = \sum_{p=0}^{\infty} \hat{u}_p^\tau \, (\boldsymbol{x} - \boldsymbol{c}_\tau)^p = \sum_{p=0}^{\infty} \hat{u}_p^\tau \left[(\boldsymbol{x} - \boldsymbol{c}_\sigma) + (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau) \right]^p$$

$$= \sum_{p=0}^{\infty} \hat{u}_p^\tau \sum_{r=0}^{p} \binom{p}{r} (\boldsymbol{x} - \boldsymbol{c}_\sigma)^r \, (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{p-r} = \sum_{r=0}^{\infty} \underbrace{\left\{ \sum_{p=r}^{\infty} \binom{p}{r} (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{p-r} \, \hat{u}_p^\tau \right\}}_{=\hat{u}_r^\sigma} (\boldsymbol{x} - \boldsymbol{c}_\sigma)^r .$$

□

In the context of the fast multipole method, we observe that the incoming field is always represented as a *polynomial of degree* $P-1$. Since it is not a general harmonic function, all sums in Theorem 2.1 are in fact finite. We see that the incoming-from-incoming translation operator corresponds to the upper triangular matrix with entries

$$\mathbf{T}^{\text{ifi}}_{\sigma,\tau}(r,p) = \begin{cases} \binom{p}{r}(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{p-r} & \text{for } r \leq p, \\ 0 & \text{for } r > p. \end{cases}$$

Observe that the transfer of an incoming expansion is an exact operation (up to round-off errors).

2.5. The classical Fast Multipole Method

We now have all tools required to describe the classical FMM in detail.

Given a set of sources $\{q_i\}_{i=1}^N$ with associated locations $\{\boldsymbol{x}_i\}_{i=1}^N$, the first step is to find a minimal square Ω that holds all points. Next, subdivide Ω into a hierarchy of successively smaller boxes as described in Section 2.1. Then fix an integer P that determines the accuracy, remembering that a larger P gives higher accuracy, but also higher cost. As we saw in Section 1.4, to attain a requested accuracy ε, we need to pick P roughly as $P \approx \log(\varepsilon)/\log(\sqrt{2}/3)$. (We will return to the question of errors for a more detailed discussion in Section 2.6.)

Once we have constructed the tree and picked an expansion length P, the actual computations can start. We first execute an upwards sweep where all outgoing expansions are computed, using the techniques described in Section 2.3. Then perform a downwards sweep where all incoming expansions are computed, using the techniques described in Section 2.4. After these sweeps have completed, the final step is to loop over all leaf boxes in order to build the actual potentials. Since we at this point know the incoming expansion for each leaf box, this is simply a matter of expanding the incoming field to actual potentials, and then adding the contributions from sources in the box itself and its immediate neighbors via a brute force calculation. For a leaf box τ, we would have

$$\mathbf{u}(I_\tau) = \mathbf{A}^{\text{tfi}}_\tau\, \hat{\mathbf{u}}_\tau + \mathbf{A}(I_\tau, I_\tau)\, \mathbf{q}(I_\tau) + \sum_{\sigma \in \mathcal{L}^{\text{nei}}_\tau} \mathbf{A}(I_\tau, I_\sigma)\, \mathbf{q}(I_\sigma).$$

The classical FMM we have just described is summarized in Figure 2.5.

Observe that the translation operators $\mathbf{T}^{\text{ofo}}_{\tau,\sigma}$, $\mathbf{T}^{\text{ifo}}_{\tau,\sigma}$, and $\mathbf{T}^{\text{ifi}}_{\tau,\sigma}$ can all be pre-computed since they depend only on P and on the vectors $\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma$. The tree structure of the boxes ensures that only a small number of values of $\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma$ are encountered.

Since we for now restrict attention to uniform trees, it is relatively straight-forward to estimate the asymptotic complexity of the classical FMM. In what follows, let m denote the typical number of nodes in a leaf box.

Let us consider the cost of applying each of the translation operators in turn, starting with the outgoing-from-sources operators that build the outgoing expansion for each leaf box. For each leaf box, the matrix $\mathbf{T}^{\text{ofs}}_\tau$ is of size $P \times m$, so the cost to build and apply it is Pm. Since there are roughly N/m leaf boxes, we find an overall cost of

$$T_{\text{ofs}} \sim \frac{N}{m} \times Pm = PN.$$

(One could alternately argue that $T_{\text{ofs}} \sim PN$ since each particle contributes to precisely P expansion coefficients.) Next we consider the outgoing-from-outgoing operators that arise in the upwards pass. These are each of size $P \times P$, and each box in the tree involves precisely one communication to its parent box.

Upwards sweep: Compute the outgoing expansion of each box in a pass over all boxes, going from smaller boxes to larger. For a leaf box, compute the expansion directly from the sources in the box; for a parent box, use the outgoing expansions of its children, cf. Section 2.3:

loop over levels ℓ, from the finest to the coarsest
  loop over all boxes τ on level ℓ
    if (τ is a leaf)
      $\hat{\mathbf{q}}_\tau = \mathbf{T}^{\text{ofs}}_\tau \, \mathbf{q}(I_\tau)$
    else
      $\hat{\mathbf{q}}_\tau = \sum_{\sigma \in \mathcal{L}^{\text{child}}_\tau} \mathbf{T}^{\text{ofo}}_{\tau,\sigma} \, \hat{\mathbf{q}}_\sigma$
    end if
  end loop
end loop

Downwards sweep: Compute the incoming expansion for every box in a pass over all boxes, going from larger boxes to smaller. For each box, combine the incoming expansion of its parent with the contributions from the outgoing expansions of all boxes in its interaction list, cf. Section 2.4:

Set $\hat{\mathbf{u}}_\tau = \mathbf{0}$ for every box τ on level 1.
loop over levels ℓ, from level $\ell = 2$ to the finest
  loop over all boxes τ on level ℓ
    Let ν denote the parent of τ.
    $\hat{\mathbf{u}}_\tau = \mathbf{T}^{\text{ifi}}_{\tau,\nu} \, \hat{\mathbf{u}}_\nu + \sum_{\sigma \in \mathcal{L}_\tau} \mathbf{T}^{\text{ifo}}_{\tau,\sigma} \, \hat{\mathbf{q}}_\sigma$.
  end loop
end loop

Compute the potential on every leaf by expanding its incoming potential (via the targets-from-incoming operator) and then adding the contributions from its near field via direct evaluation, cf. Section 2.5:

loop over every leaf boxes τ
  $\mathbf{u}(I_\tau) = \mathbf{A}^{\text{tfi}}_\tau \, \hat{\mathbf{u}}_\tau + \mathbf{A}(I_\tau, I_\tau) \, \mathbf{q}(I_\tau) + \sum_{\sigma \in \mathcal{L}^{\text{nei}}_\tau} \mathbf{A}(I_\tau, I_\sigma) \, \mathbf{q}(I_\sigma)$
end loop

FIGURE 2.5. The Classical Fast Multipole Method.

On level ℓ, there are roughly $\frac{N}{m} 4^{L-\ell}$ boxes, so the cost to apply all the outgoing-from-outgoing expansion operators is roughly

$$T_{\text{ofo}} \sim P^2 \sum_{\ell=3}^{L} \frac{N}{m} 4^{L-\ell} = P^2 \frac{N}{m} \left(1 + \frac{1}{4} + \frac{1}{16} + \cdots + \frac{1}{4^{L-3}} \right) \leq P^2 \frac{N}{m} \frac{1}{1 - 1/4}.$$

The incoming-to-incoming operators all involve exactly the same amount of work, so $T_{\text{ifi}} = T_{\text{ofo}}$. In the downwards sweep, we also apply all the incoming-from-outgoing operators. While there can be up to 27 of these, the fact that the number is bounded by a fixed constant means that the cost to process any box remains $O(P^2)$, so

$$T_{\text{ifo}} \sim T_{\text{ifi}} \sim T_{\text{ofo}} \sim P^2 N/m.$$

The cost to expand the incoming expansions is the same as the cost to build the outgoing expansions, so

$$T_{\text{tfi}} \sim T_{\text{ofs}} \sim PN.$$

The cost to evaluate all near field interactions is on average $9m$ per particle, so we find that

$$T_{\text{near}} \sim N \times 9m \sim Nm.$$

Adding up all the costs we obtain an estimate for the total cost of

$$T_{\mathrm{FMM}} = T_{\mathrm{ofs}} + T_{\mathrm{ofo}} + T_{\mathrm{ifo}} + T_{\mathrm{ifi}} + T_{\mathrm{tfi}} + T_{\mathrm{near}} \sim PN + P^2N/m + P^2N/m + P^2N/m + PN + Nm.$$

Finally, we observe that the number m of nodes in a leaf box is a number that we are free to pick as we see fit. By choosing $m \sim P$, we obtain the estimate

$$T_{\mathrm{FMM}} = PN + P^2N/P + NP \sim PN \sim \{\text{Use that } P \sim \log(1/\varepsilon)\} \sim \log(1/\varepsilon)\, N.$$

REMARK 2.1. *In this section, we restricted attention to the case of relatively uniform particle distributions for which a a fully populated tree (as described in Section 2.1) is appropriate. When the particle distribution is non-uniform, locally adaptive trees perform much better. The basic FMM can readily be adapted to operate on non-uniform trees. The key modification required is that some outgoing expansions need to be broadcast directly to target points, and some incoming expansions must receive direct contributions from source points in certain boxes* [**3**]. *We return to this question in Section 3.1.*

2.6. Error analysis

The potentials computed by the FMM are not exact since all expansions have been truncated to P terms. An analysis of how such errors could propagate through the transformations across all levels is technically complicated, and should seek to estimate both the worst case error, and the statistically expected error [**6, 27**]. As it happens, the global error is in most cases similar to the (worst case) local truncation error, which means that it scales roughly as η^P, where $\eta = \sqrt{2}/(4-\sqrt{2}) = 0.5469\cdots$ [**22**]. As a rough estimate, we see that in order to achieve a given tolerance ε, we need to pick

$$P \approx \log(\varepsilon)/\log(\eta).$$

As P increases, the asymptotic complexity of the 2D FMM is $O(PN)$, provided that each leaf box holds $O(P)$ sources. In consequence, the overall complexity can be said to scale as $\log(1/\varepsilon)\, N$ as $\varepsilon \rightarrow 0$ and $N \rightarrow \infty$.

2.7. The Barnes-Hut algorithm

Before we close this chapter, let us describe a method called "Barnes-Hut" [**1**] that was introduced at around the same time as the FMM. The Barnes-Hut method has complexity $O(N \log N)$, which is worse than the $O(N)$ of the FMM, but has the appeal that it attains near linear complexity without introducing any machinery other than classical multipole expansions, which are already familiar to many scientists and engineers. In other words, it does not need any "incoming expansions" or new translation opreators. Let us stress that the material in this section is included mostly to provide historical context; we will not build on these ideas in the remainder of the book.

In the Barnes-Hut method, we introduce a hierarchical tree of boxes, as described in Section 2.1, and then execute an upwards pass where we build all the outgoing expansions, as described in Section 2.3. The key observation is now that once we have all outgoing expansions, we can compute any potential using only $O(\log N)$ work. For an informal description of how this works, let us consider a leaf box τ, as shown in Figure 2.6. What work do we need to do in order to evaluate the potential at the blue target points in τ? We first need to evaluate interactions with the near boxes directly, without using any expansions. There are only a small number of near neighbors, however, so this is inexpensive. For the far-field interactions, we tessellate the far field into as coarse boxes as possible while maintaining the requirement that the target points (in blue in Figure 2.6) should be well-separated from any box in the tessellation. Then we use the outgoing expansions to evaluate the potentials. We will show that there are at most $O(\log N)$ boxes in this tessellation, which leads us to an overall complexity of $\sim N \times \log N$.

To make the description precise, let us for a leaf box τ introduce the list $\mathcal{L}_\tau^{\text{anc}}$ of its "ancestors"; this list contains τ itself, and also its parent, grand-parent, great grand-parent, etc. Then the tessellation of the far-field that we seek is the union of the boxes in the interaction lists for all of the ancestors of τ, so that

$$\Omega_\tau^{\text{far}} = \bigcup_{\sigma \in \mathcal{L}_\tau^{\text{anc}}} \bigcup_{\nu \in \mathcal{L}_\sigma} \Omega_\nu.$$

The formula for evaluating all potentials in τ therefore takes the form

$$\mathbf{u}(I_\tau) = \mathbf{A}(I_\tau, I_\tau)\mathbf{q}(I_\tau) + \sum_{\sigma \in \mathcal{L}_\tau^{\text{near}}} \mathbf{A}(I_\tau, I_\sigma)\mathbf{q}(I_\sigma) + \sum_{\sigma \in \mathcal{L}_\tau^{\text{anc}}} \sum_{\nu \in \mathcal{L}_\sigma} \mathbf{T}_{\tau,\nu}^{\text{tfo}} \hat{\mathbf{q}}_\nu, \tag{2.6}$$

where the first two terms represent near field interactions and the last term represents far-field interactions.

In order to estimate the asymptotic complexity of the simplified scheme, let us again suppose that we have a uniform tree where each leaf box holds approximately m nodes. Then the cost to evaluate the potentials in one leaf box is

$$T_{\text{leaf}} \approx m^2 + 8m^2 + L \times 27 \times mP, \tag{2.7}$$

where each term is an estimate of the cost to evaluate the corresponding term in (2.6), since the near-field matrices are each of size $m \times m$ and the matrices $\mathbf{T}_{\tau,\nu}^{\text{tfo}}$ are each of size $m \times P$. There are roughly N/m leaf boxes, so

$$T_{\text{BarnesHut}} \sim \frac{N}{m}\left(m^2 + mLP\right) \sim \{\text{Use that } L \sim \log(N/m)\} \sim Nm + PN\log(N/m).$$

Using simply $m = O(1)$, we find an overall complexity of $O(PN \log N)$.

We summarize the Barnes-Hut scheme in Figure 2.7. The formulation given in the figure does not directly invoke the list of ancestors. Instead, we use an algebraically equivalent formulation where instead of focusing on what information each box needs to *gather* in order to evaluate the potentials in a box, we instead focus on what information each box needs to *broadcast.*

REMARK 2.2. *In the original Barnes-Hut scheme, there were no outgoing-from-outgoing translation operators. Instead, each outgoing expansion was computed from scratch from the sources inside the box. In this version, the "upwards sweep" in Figure 2.7 could be replaced by a very simple sweep over all boxes:*

loop over all boxes τ
  $\hat{\mathbf{q}}_\tau = \mathbf{T}_\tau^{\text{ofs}}\, \mathbf{q}(I_\tau)$
end loop

This formulation has the advantage that each loop can be traversed in an entirely arbitrary order. However, since the outgoing-from-outgoing translations are exact up to floating point precision, one may as well use these to save some time. (The overall execution time does not change much either way, since the dominant time is the sweep over all interaction lists.)

REMARK 2.3. *In Remark 1.4, we illustrate how the various translation operators in the FMM can be viewed as legs in a commutative diagram. In the Barnes-Hut method, we remove one leg from this diagram, and end up with the simpler one*

$$\begin{array}{ccc} \mathbf{q}_\tau & \xrightarrow{\mathbf{A}(I_\sigma, I_\tau)} & \mathbf{u}_\sigma \\ {\scriptstyle \mathbf{T}_\tau^{\text{ofs}}}\downarrow & \nearrow_{\mathbf{T}_{\sigma,\tau}^{\text{tfo}}} & \\ \hat{\mathbf{q}}_\tau & & \end{array}$$

An equivalent formulation would be as an approximate matrix factorization

$$\begin{array}{ccccc} \mathbf{A}(I_\sigma, I_\tau) & \approx & \mathbf{T}_{\sigma,\tau}^{\text{tfo}} & \mathbf{T}_\tau^{\text{ofs}}. \\ n_\sigma \times n_\tau & & n_\sigma \times P & P \times n_\tau \end{array} \tag{2.8}$$

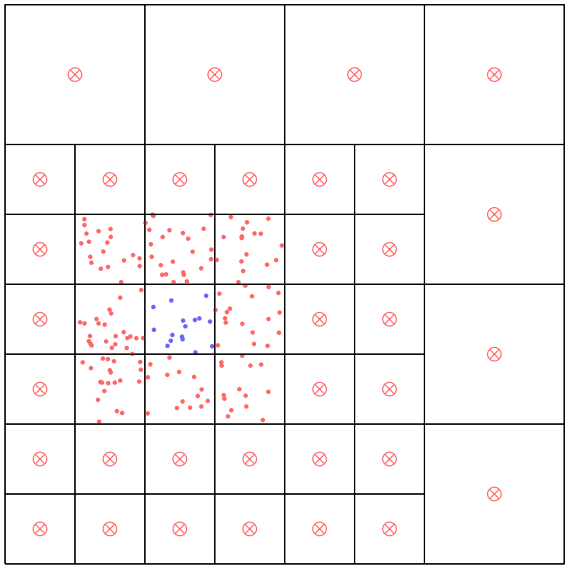


FIGURE 2.6. Illustration of how the potentials in a box τ (marked with blue dots) are evaluated in the Barnes-Hut scheme described in Section 2.7. Contributions to the potential caused by sources in τ itself (blue dots), or in its immediate neighbors (red dots) are computed via direct evaluation. The contributions from more distant sources are computed via the outgoing expansions centered on the $\otimes$ marks in the figure.

Upwards sweep: Compute the outgoing expansion of each box in a pass over all boxes, going from smaller boxes to larger. For a leaf box, compute the expansion directly from the sources in the box; for a parent box, use the outgoing expansions of its children, cf. Section 2.3:

loop over levels ℓ, from the finest to the coarsest
  loop over all boxes τ on level ℓ
    if (τ is a leaf)
      $\hat{\mathbf{q}}_\tau = \mathbf{T}_\tau^{\mathrm{ofs}}\, \mathbf{q}(I_\tau)$
    else
      $\hat{\mathbf{q}}_\tau = \sum_{\sigma \in \mathcal{L}_\tau^{\mathrm{child}}} \mathbf{T}_{\tau,\sigma}^{\mathrm{ofo}}\, \hat{\mathbf{q}}_\sigma$
    end if
  end loop
end loop

Evaluate the far field potentials. Each box τ broadcasts its outgoing representation to all boxes in its interaction list:

$\mathbf{u} = 0$
loop over all boxes τ
  loop over all $\sigma \in \mathcal{L}_\tau^{\mathrm{int}}$
    $\mathbf{u}(I_\sigma) = \mathbf{u}(I_\sigma) + \mathbf{T}_{\sigma,\tau}^{\mathrm{tfo}}\, \hat{\mathbf{q}}_\tau$
  end loop
end loop

Evaluate the near field interactions:

loop over all leaf boxes τ
  $\mathbf{u}(I_\tau) = \mathbf{u}(I_\tau) + \mathbf{A}(I_\tau, I_\tau)\, \mathbf{q}(I_\tau) + \sum_{\sigma \in \mathcal{L}_\tau^{\mathrm{nei}}} \mathbf{A}(I_\tau, I_\sigma)\, \mathbf{q}(I_\sigma)$
end

FIGURE 2.7. The Barnes-Hut algorithm.

CHAPTER 3

Extensions and improvements on the basic FMM

In this chapter, we describe different ways that the basic FMM can be extended to more general situations, starting with the extension to non-uniform trees, and then continuing on to three spatial dimensions, and other interaction kernels. We also describe how the accuracy of the FMM may be analyzed, and different techniques for optimizing its performance. Throughout this chapter, we keep the discussion brief, and provide pointers to the extensive literature on the subject for an in-depth treatment.

3.1. The extension to non-uniform trees

Let us again consider the situation where we are given a set of source points $\{\boldsymbol{x}_j\}_{j=1}^N$ in a square Ω. When the points are not close to uniformly distributed in Ω, the basic FMM that we described in Chapter 2 can become inefficient. Luckily, it turns out to be relatively straight-forward to modify the FMM to allow it to use a non-uniform tree, which more effectively organizes the points [**3**]. To illustrate, let us consider a distribution of points such as the one shown in Figure 3.1. To build a tree that is tailored to the given set of locations, we start by picking a positive integer, say m, that represents the largest number of nodes that we are willing to have in a single leaf. We then split Ω into four equisized squares. Any box that ends up being empty is pruned from the tree. For the boxes that remain, we proceed recursively to subdivide any of them that hold more than m points. Figure 3.1 illustrates the process.

Once the adaptive tree has been built, we next build all outgoing expansions using exactly the same process as for the original FMM. The only modification required is that some boxes will have fewer than four children. The difference appears once we start to build the incoming expansions and the potentials in the all boxes. In a uniform tree, any box τ interacts only with other boxes that *are on the same level as itself* (in addition to its parent and children, of course). In an adaptive tree, things can get more complicated. To illustrate, let us consider how to evaluate the potential in box $\tau = 18$ on level 3 in Figure 3.1. In order to evaluate the potential in this box, we need to use full evaluation for all leaf boxes that directly touch it. We call these boxes its neighbor list, which in this case works out to $\mathcal{L}_{18}^{(\text{nei})} = \{19, 48, 49, 52\}$. Observe that the boxes in the neighbor list need no longer be located on the same level as τ itself. Next let us consider the interaction between box 18 and box 51. While box 51 is not well-separated from box 18, it *is* the case that box 18 is well-separated from box 51. This means that we can effectively evaluate the potential caused on box 18 by using the outgoing expansion for box 51. Figure 3.2 illustrates how the three boxes 50, 51, and 53 communicate their outgoing expansions to potentials in box 18. Analogously, these three boxes $\{50, 51, 53\}$ must gather their incoming expansions directly from the source points in box 18.

In order to account for the new types of interactions between boxes on different levels, we introduce two additional lists. For a box τ, we for historical reasons refer to these as List 3 and List 4, or $\mathcal{L}_\tau^{(3)}$ and $\mathcal{L}_\tau^{(4)}$. For a box τ, the list $\mathcal{L}_\tau^{(3)}$ keeps track of of the boxes that must communicate their outgoing expansions directly to target points in Ω_τ. List 4 is the dual of list 3, so $\mathcal{L}_\tau^{(4)}$ keeps track of all boxes that must communicate their sources directly to the incoming expansion for box τ. Precise definitions of these lists are given in Figure 3.3, and then the full adaptive FMM is summarized in Figure 3.4. For completeness, we also list in Figure 3.5 all the seven translation operators that arise in the FMM.

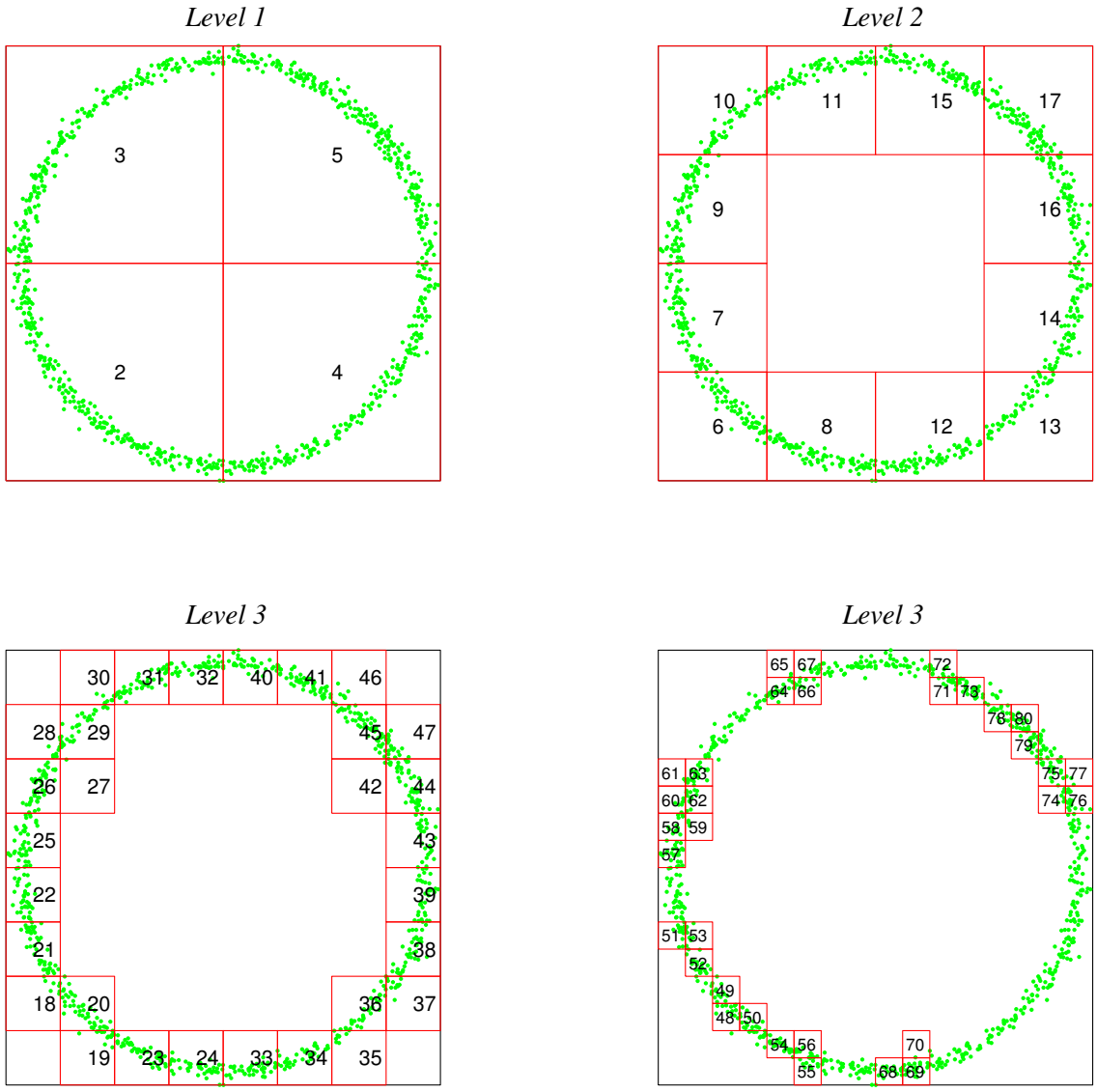
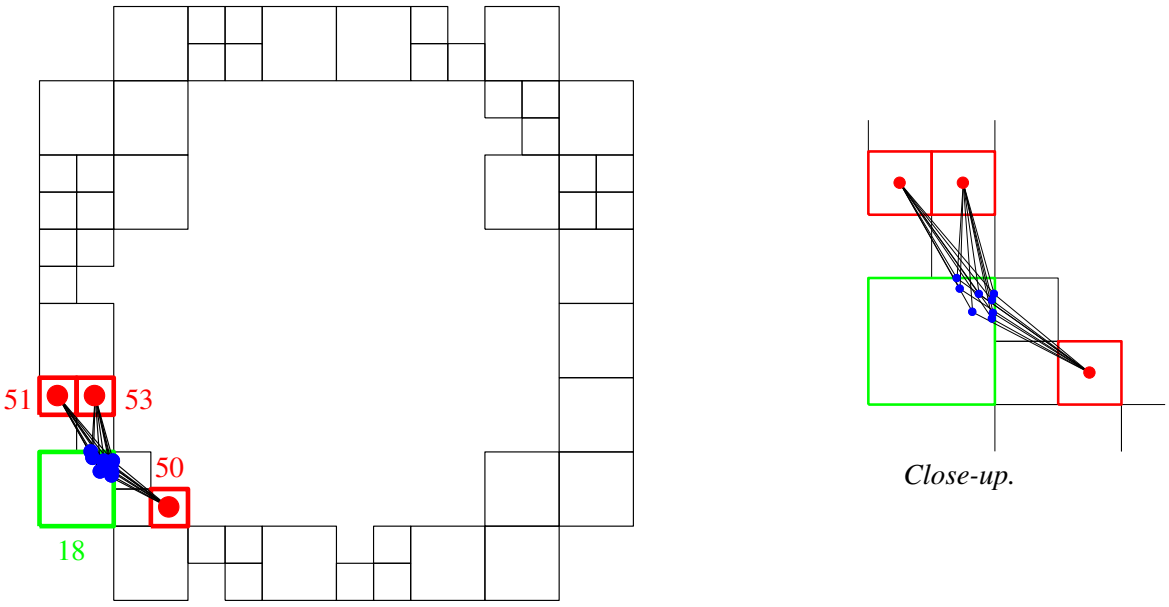


FIGURE 3.1. The green points represent a given distribution of source points. Since the points are non-uniformly distributed, an adaptive tree is used. Any box holding more than some preset maximal number of points is subdivided into 4 equisized boxes. Any of these that end up being empty are pruned from the tree.

REMARK 3.1 (FMM versus PPPM methods). *The ability to effectively work with non-uniform trees is an advantage of the FMM that is often decisive. In situations where the sources are distributed uniformly in a box, it is often possible to solve the potential evaluation problem by leveraging the exceptionally high speed of the FFT. In principle, the fastest versions of the FMM only work when the distribution of source points forms a uniform grid in the box, but techniques have been developed that work for general source distributions as well. The idea in these "PPPM" (particle-particle / particle-mesh) methods is to lay down an artificial uniform grid on the computational box, then create some "equivalent" charges at these new points, and use the grid based problem for evaluating long range interactions, while performing local corrections to accurately evaluate short range interactions* [**18**]. *These techniques are very popular, as they tend to be very fast and can be easier to implement than an FMM. However, PPPM methods suffer suffer in performance when the distribution of points is non-uniform, and can also be challenging to implement in situations when*



Close-up.

FIGURE 3.2. The box $\tau = 18$ is marked in green. The potentials at target points in this box are collected from the outgoing expansions of the boxes in $\mathcal{L}_{18}^{(3)} = \{50, 51, 53\}$. Analogously, sources in box 18 communicate their contributions directly to the incoming expansions for the boxes $\{50, 51, 53\}$.

$\mathcal{L}_\tau^{(\text{child})}$	The children of τ.
$\mathcal{L}_\tau^{(\text{parent})}$	The parent of τ.
$\mathcal{L}_\tau^{(\text{nei})}$	For a parent box τ, $\mathcal{L}_\tau^{(\text{nei})}$ is empty. For a leaf box τ, $\mathcal{L}_\tau^{(\text{nei})}$ is a list of the leaf boxes that directly border τ.
$\mathcal{L}_\tau^{(\text{int})}$	A box $\sigma \in \mathcal{L}_\tau^{(\text{int})}$ iff σ and τ are on the same level, σ and τ are well-separated, but the parents of σ and τ are not well-separated.
$\mathcal{L}_\tau^{(3)}$	For a parent box τ, $\mathcal{L}_\tau^{(3)}$ is empty. For a leaf box τ, a box $\sigma \in \mathcal{L}_\tau^{(3)}$ if and only if σ lives on a finer level than τ, τ is well-separated from σ, but τ is not well-separated from the parent of σ.
$\mathcal{L}_\tau^{(4)}$	The dual of $\mathcal{L}_\tau^{(3)}$. In other words, $\sigma \in \mathcal{L}_\tau^{(4)}$ if and only if $\tau \in \mathcal{L}_\sigma^{(3)}$.

FIGURE 3.3. A summary of the various "lists" that are used in the FMM. The last two, $\mathcal{L}_\tau^{(3)}$ and $\mathcal{L}_\tau^{(4)}$, are needed only for non-uniform trees.

non-periodic boundary conditions are required. (In contrast, the FMM can readily handle both periodic and non-periodic boundary conditions.)

Set $\hat{\mathbf{u}}_\tau = \mathbf{0}$ and $\hat{\mathbf{q}}_\tau = \mathbf{0}$ for all τ.

loop over all leaf nodes τ
  $\hat{\mathbf{q}}_\tau = \mathbf{T}_\tau^{(\mathrm{ofs})}\, \mathbf{q}(J_\tau)$
end loop

loop over levels $\ell = L,\ L-1,\ \ldots,\ 2$
  loop over all nodes τ on level ℓ
    $\hat{\mathbf{q}}_\tau = \sum_{\sigma \in \mathcal{L}_\tau^{(\mathrm{child})}} \mathbf{T}_{\tau,\sigma}^{(\mathrm{ofo})}\, \hat{\mathbf{q}}_\sigma$
  end loop
end loop

loop over all nodes τ
  $\hat{\mathbf{u}}_\tau = \hat{\mathbf{u}}_\tau + \sum_{\sigma \in \mathcal{L}_\tau^{(\mathrm{int})}} \mathbf{T}_{\tau,\sigma}^{(\mathrm{ifo})}\, \hat{\mathbf{q}}_\sigma.$
end loop

loop over all nodes τ
  $\hat{\mathbf{u}}_\tau = \hat{\mathbf{u}}_\tau + \sum_{\sigma \in \mathcal{L}_\tau^{(4)}} \mathbf{T}_{\tau,\sigma}^{(\mathrm{ifs})}\, \mathbf{q}(J_\sigma).$
end loop

loop over levels $\ell = 2,\ 3,\ 4,\ \ldots,\ L-1$
  loop over all nodes τ on level ℓ
    loop over all children σ of τ
      $\hat{\mathbf{u}}_\sigma = \hat{\mathbf{u}}_\sigma + \mathbf{T}_{\sigma,\tau}^{(\mathrm{ifi})}\, \hat{\mathbf{u}}_\tau.$
    end loop
  end loop
end loop

loop over all leaf nodes τ
  $\mathbf{u}(J_\tau) = \mathbf{T}_\tau^{(\mathrm{tfi})}\, \hat{\mathbf{u}}_\tau$
end loop

loop over all nodes τ
  $\mathbf{u}(J_\tau) = \mathbf{u}(J_\tau) + \sum_{\sigma \in \mathcal{L}_\tau^{(3)}} \mathbf{T}_{\tau,\sigma}^{(\mathrm{tfo})}\, \hat{\mathbf{q}}_\sigma.$
end loop

loop over all leaf nodes τ
  $\mathbf{u}(J_\tau) = \mathbf{u}(J_\tau) + \mathbf{A}(J_\tau, J_\tau)\, \mathbf{q}(J_\tau)$
  $\quad + \sum_{\sigma \in \mathcal{L}_\tau^{(\mathrm{nei})}} \mathbf{A}(J_\tau, J_\sigma)\, \mathbf{q}(J_\sigma)$
end loop

FIGURE 3.4. The adaptive FMM. The algorithm takes as input a set of points $\{\boldsymbol{x}_j\}_{j=1}^N$, a precision parameter P, and a number m that specifies the maximum number of nodes allowed in a leaf. Before the computations listed start, the FMM finds a bounding square Ω (if it is not specified by the user), builds the adaptive tree for the given set of points, and determines the lists specified in Figure 3.3.

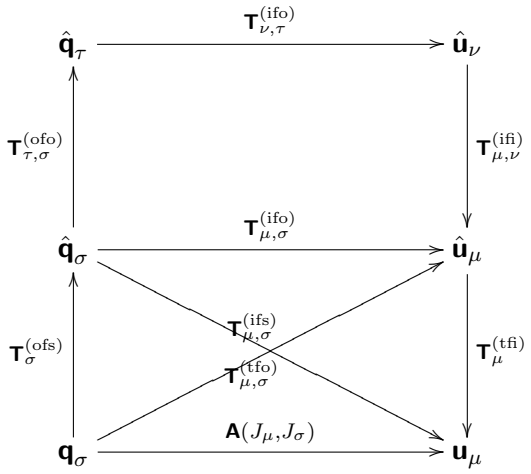


FIGURE 3.5. A commutative diagram that summarizes 7 different types of translation operators used in the adaptive FMM. The two operators $\mathbf{T}_{\mu,\sigma}^{(\mathrm{tfo})}$ and $\mathbf{T}_{\mu,\sigma}^{(\mathrm{ift})}$ which correspond to the two diagonal arrows are not required for uniform trees.

3.2. The extension from two to three dimensions

It is in some sense relatively straight-forward to modify the basic FMM for two dimensional problems that we have described to handle problems in three dimensions. The concept of multipole expansions has an immediate three dimensional analog, and the data structures can readily be changed from a quadtree of squares to an octree of cubes. The resulting algorithm has complexity $O(N)$, but a significant challenge arises in that the scaling constant turns out to depend on the requested accuracy ε not as $\log(1/\varepsilon)$ in two dimensions, but as $(\log(1/\varepsilon))^2$. In practice, this means that the algorithm becomes quite slow if a high accuracy is requested. It turns out to be possible to accelerate the scheme substantially, but this will require the introduction of some additional machinery. In this section, our intention is to give a high level description of how the generalization to 3D works; for details, we refer to the survey [**12**].

Let us first consider the changes in the data structures that are required. We are now given a set of points $\{\boldsymbol{x}_j\}_{j=1}^N$ in $\mathbb{R}^3$ that are contained in a cube Ω. (The generalization to a rectangular parallelepiped is straightforward.) We create a tree of boxes by splitting Ω into 8 equisized cubes, and then continuing subdividing the cubes recursively until only a handful of nodes remain in each leaf. The definitions of lists remain *exactly* the same as for the 2D case, as listed in Figure 3.3. We observe that while the generalization from a quadtree to an octree is in principle simple, things get more expensive from a practical point of view since all lists get longer. For a fully populated tree, a typical neighbor list has 26 boxes in 3D, as opposed to 8 in 2D. The interaction list in 3D holds no fewer than 189 boxes, as opposed to the 27 we had in 2D. (More generally, in $\mathbb{R}^d$, the interaction list holds $6^d - 3^d$ boxes, which is one reason the FMM scales poorly with dimension.)

As far as multipole expansions go, the generalization is conceptually simple. However, the notation required is more complicated, and the number of terms required to attain a given accuracy grows quite substantially. The interaction potential for Laplace problems (electrostatics, gravitation, etc.) is now

$$G(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{4\pi|\boldsymbol{x} - \boldsymbol{y}|}. \tag{3.1}$$

(The definition (3.1) is slightly inconsistent with our treatment of the 2D case since it includes the scaling factor 4π. We do this to conform with the standard literature on 3D multipole expansions.) We seek to separate the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ to find an approximate expansion of the form (1.4). This is a classical problem in mathematical physics, whose solution involve so called spherical harmonic functions. To define these, let us introduce polar coordinates (r, θ, φ) so that

$$\boldsymbol{x} = (x_1,\, x_2,\, x_3) = r(\sin\theta\cos\varphi,\, \sin\theta\sin\varphi,\, \cos\theta).$$

The "spherical harmonic function" $Y_n^m = Y_n^m(\theta, \varphi)$ is defined for any non-negative integer n, and for any integer $m \in \{-n,\, -n+1,\, \ldots,\, n-1,\, n\}$ via

$$Y_n^m(\theta, \varphi) = (-1)^m \sqrt{\frac{2n+1}{4\pi}\frac{(n-|m|)!}{(n+|m|)!}}\, P_n^{|m|}(\cos\theta) e^{im\varphi},$$

where $P_n^{|m|}$ are the so called "associated Legendre functions". The functions Y_n^m form an orthogonal basis (in L^2) on the unit sphere in $\mathbb{R}^3$, and together form the analog in 3D to the set $\left\{\frac{1}{\sqrt{2\pi}}\right\} \cup \left\{\frac{\cos(n\theta)}{\sqrt{\pi}},\, \frac{\sin(n\theta)}{\sqrt{\pi}}\right\}_{n=1}^{\infty}$, which forms an orthonormal basis for the unit circle in 2D. For each spherical harmonic function Y_n^m, there are two associated "solid harmonic functions" which are defined via

$$R_n^m(\boldsymbol{x}) = r^n Y_n^m(\theta, \varphi), \tag{3.2}$$

$$S_n^m(\boldsymbol{x}) = \frac{1}{r^{n+1}} Y_n^m(\theta, \varphi). \tag{3.3}$$

The "regular solid harmonic function" R_n^m solves the Laplace equation at all points in $\mathbb{R}^3$; it is zero at the origin, and grows as r^n as $r \to \infty$. The "irregular solid harmonic function" S_n^m solves the Laplace equation

at all points in $\mathbb{R}^3\backslash\{\mathbf{0}\}$; it is singular at the origin, and decays as r^{-n-1} as $r \to \infty$. Using the solid harmonic functions, the separation of variables of the fundamental solution takes the form

$$\frac{1}{4\pi|\boldsymbol{x}-\boldsymbol{y}|} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} R_n^{-m}(\boldsymbol{y})\, S_n^m(\boldsymbol{x}), \qquad \text{for } |\boldsymbol{x}| > |\boldsymbol{y}|. \tag{3.4}$$

In other words, if we are given a set of source points $\{\boldsymbol{y}_j\}_{j=1}^n$ in a cube Ω_σ with associated source strengths $\{q_j\}_{j=1}^n$ and a positive integer P, then the field

$$u(\boldsymbol{x}) = \sum_{j=1}^{n} \frac{1}{4\pi|\boldsymbol{x}-\boldsymbol{y}_j|}\, q_j$$

can be approximated by

$$u_P(\boldsymbol{x}) = \sum_{n=0}^{P} \sum_{m=-n}^{n} \hat{q}^{n,m}\, S_n^m(\boldsymbol{x}-\boldsymbol{c}_\sigma),$$

where the multipole coefficients are defined via

$$\hat{q}^{n,m} = \sum_{j=1}^{n} R_n^{-m}(\boldsymbol{y}_j - \boldsymbol{c}_\sigma)\, q_j, \qquad n \in \{0,1,\ldots,P\}, \qquad m \in \{-n,\ldots,n\},$$

and where $\boldsymbol{c}_\sigma$ is the center of Ω_σ. If $\boldsymbol{x}$ is well-separated from Ω_σ (with Definition 1 generalized to 3D in the obvious way), then as $P \to \infty$, we have

$$\big|u(\boldsymbol{x}) - u_P(\boldsymbol{x})\big| = O(\eta^P)$$

where $\eta = \sqrt{3}/3$, since the radius of the smallest sphere enclosing a cube or side length $2a$ is $\sqrt{3}a$. We refrain from going in to details about the exact form of the various translation operators required in the FMM, but in principle everything works exactly the same as in 2D, and we have analytic expressions for all entries.

Comparing the machinery of multipole expansions required for problems and two and three dimensions respectively, there are important differences. An obvious one is that there are for any given order P far more basis functions in the 3D case. In 2D, there were $1+2P$ functions of order at most P. In 3D, the number of multipoles of order precisely p is $1+2p$, leading to an overall $(P+1)^2$ functions of order P or less. This growth in the number of terms required leads to substantial increase in the time required to apply all translation operators. Let us consider the incoming-from-outgoing operator, which is often the most time consuming one. For a fully populated tree with N_{tree} parent nodes, we find that the costs in two and three dimensions are

$$\begin{aligned} &\text{Cost of applying all ifo (“M2L”) operators in 2D:} \quad N_{\text{parent}} \times 27 \times (2P+1)^2 \\ &\text{Cost of applying all ifo (“M2L”) operators in 3D:} \quad N_{\text{parent}} \times 189 \times (P+1)^4 \end{aligned}$$

To make things even worse, we in 3D typically have to pick a larger P to achieve any given tolerance than we did in 2D. To be precise, in 3D, the truncation errors scale as $\eta_{3\text{D}}^P$ where $\eta_{3\text{D}} = \sqrt{3}/(4-\sqrt{3}) = 0.76\cdots$ whereas in 2D we have $\eta_{2\text{D}} = \sqrt{2}/(4-\sqrt{2}) = 0.54\cdots$; cf. [**23**] for details.

A detailed analysis of the asymptotic complexity of the basic FMM in 3D is given in [**12**]. It is demonstrated that by choosing the maximal number of particles in a leaf as $O(P^2)$ (so that the number of boxes scales as N/p^2), the overall complexity becomes $O(NP^2)$. However, the scaling factor remains quite high, which coupled with the fact that errors decay only slowly as P increases makes this version of the 3D FMM of limited use in cases where high accuracy is required.

In order to accelerate the 3D FMM at high accuracies, Rokhlin and Greengard [**12**] developed a technique that reduces the cost of applying each incoming-from-outgoing (“M2L”) translation operator from $O(P^4)$ to $O(P^2)$. The idea is to split the interaction list into several smaller pieces, that each lie within a limited solid angle from the source box. One can then prove that after performing various local transformations at the

source and the target boxes, it is possible to transfer the expension via a *diagonal* translation operator. This new method is substantially faster the classical FMM, but also requires more effort to implement.

REMARK 3.2. *While switching to diagonal translation operators leads to a very fast FMM, the modifications required to implement this technique are non-trivial to code effectively. There is a much simpler method for reducing the cost of applying the incoming-from-outgoing ("M2L") translation operators that in practice is often "fast enough." To describe it, let us consider a source box Ω_σ that needs to communicate its outgoing expansion to the incoming expansion for a target box Ω_τ. The idea is then to carry out a local rotation of the coordinate system so that the axis corresponding to polar coordinate $\theta = 0$ goes through the line that connects the centers of Ω_σ and Ω_τ; then each Fourier mode in the azimuthal variable φ communicates independently of all the others. This rotation method reduces the cost of applying each ifo translation operator from $O(P^4)$ to $O(P^3)$. This is of course not as good as the $O(P^2)$ cost of diagonal translation operators, but it is much simpler to implement* [**12**, Sec. 6.1].

3.3. The extension from the Laplace to the Helmholtz equation

Let us next consider how to evaluate sums that involve an oscillatory kernel function such as the fundamental solution associated with the Helmholtz equation. The difficulty of this task depends crucially on the size of the computational domain in relation to the wave-length in the kernel function. If the domain is of modest size, say a dozen wave-lengths or less, then the modifications required to obtain a Helmholtz FMM given a Laplace FMM are conceptually straight-forward. However, as the problem size grows to include larger domains, fundamentally new ideas need to be introduced in order to attain linear complexity. Our objective in this section is merely to describe the main ideas at a high level, for details, see, e.g. [**4, 5, 26**].

Let us center the discussion on the problem of evaluating a sum associated with the Helmholtz equation

$$-\Delta u(\boldsymbol{x}) - \kappa^2 u(\boldsymbol{x}) = f(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^2, \tag{3.5}$$

coupled with a radiating boundary condition at infinity. In (3.5), the wave-number κ is assumed to be a positive real number. To be precise, we will given a set of sources $\{q_j\}_{j=1}^N$ and a set of points $\{\boldsymbol{x}_j\}_{j=1}^N$ in a square Ω in the plane discuss fast algorithms for evaluating the sums

$$u_i = \sum_{j \neq i} G(\boldsymbol{x}_i, \boldsymbol{x}_j) q_j, \qquad \text{for } i = 1,\, 2,\, 3,\, \ldots,\, N. \tag{3.6}$$

The kernel function is now the Hankel function of the first kind

$$G(\boldsymbol{x}, \boldsymbol{y}) = H_0^{(1)}(\kappa|\boldsymbol{x} - \boldsymbol{y}|) = J_0(\kappa|\boldsymbol{x} - \boldsymbol{y}|) + Y_0(\kappa|\boldsymbol{x} - \boldsymbol{y}|),$$

where J_0 and Y_0 are the zeroth other Bessel functions of the first and the second kind, respectively, cf. Figure 3.6. The fundamental solution of the Helmholtz equation (3.5) is the function $\phi(\boldsymbol{x}) = \frac{i}{4} G(\boldsymbol{x}, \boldsymbol{y})$. This means that up to a scaling by the factor $i/4$, the sum (3.6) represents the solution to (3.5) in the specific case where the body load f is a set of point sources located at the points $\{\boldsymbol{x}_j\}$, so that

$$f(\boldsymbol{x}) = \sum_{j=1}^{N} \delta(\boldsymbol{x} - \boldsymbol{x}_j) q_j.$$

More generally, a sum such as (3.6) may arise from applying a quadrature rule to discretize a compactly supported continuous function f. The perhaps most important application of fast algorithms for evaluating a sum such as (3.6) is in solving boundary integral equations associated with the Helmholtz equation in the plane using iterative solvers.

The foundation for the 2D Helmholtz FMM is an addition theorem for $H_0^{(1)}$ that takes the form

$$H_0^{(1)}(\kappa|\boldsymbol{x} - \boldsymbol{y}|) = \sum_{p=-\infty}^{\infty} \underbrace{\left(J_p(\kappa s) e^{ip\beta}\right)}_{=R_p(\boldsymbol{y})} \underbrace{\left(H_p^{(1)}(\kappa r) e^{-ip\alpha}\right)}_{=S_p(\boldsymbol{x})}, \qquad \text{when } |\boldsymbol{x}| > |\boldsymbol{y}|. \tag{3.7}$$

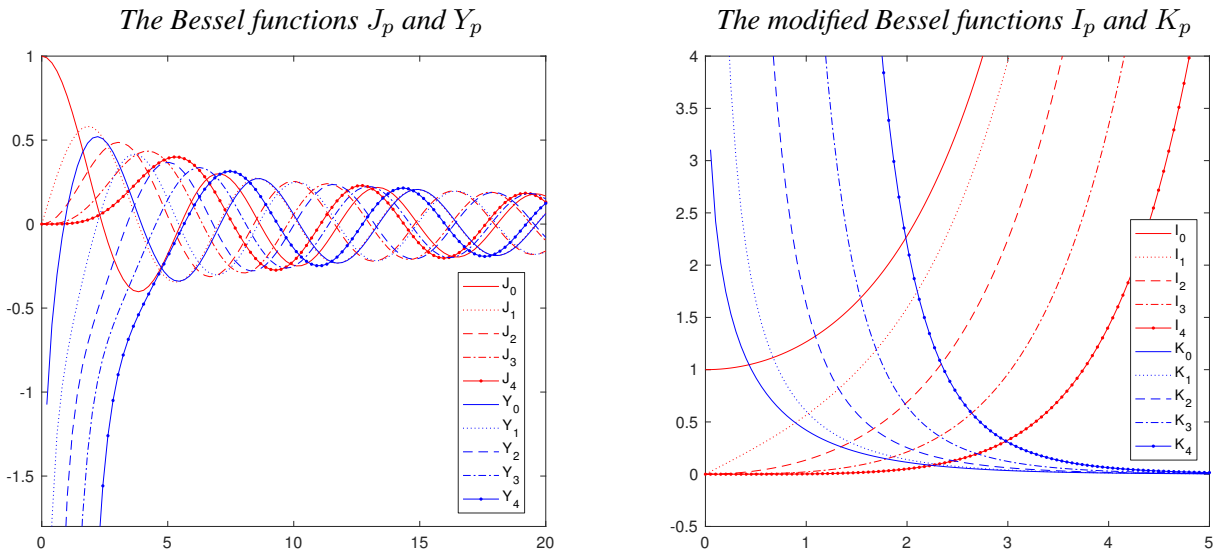


FIGURE 3.6. (a) Plots of the Bessel functions $\{J_p\}_{p=0}^4$ and $\{Y_p\}_{p=0}^4$. These arise in the FMM for the Helmholtz equation in 2D, cf. Section 3.3. (b) Plots of the modified Bessel functions $\{I_p\}_{p=0}^4$ and $\{K_p\}_{p=0}^4$, cf. Remark 3.5.

In (3.7), we used polar coordinates

$$\boldsymbol{x} = r\, e^{i\alpha} \qquad \boldsymbol{y} = s\, e^{i\beta},$$

to represent the target and the sources point respectively. The formula (3.7) is the Helmholtz analog of the formula for the fundamental solution associated with Laplace's equation

$$\log(\boldsymbol{x} - \boldsymbol{y}) = \log(\boldsymbol{x}) + \sum_{p=1}^{\infty} \left(-\frac{s^p\, e^{ip\beta}}{p} \right) \left(\frac{e^{-ip\alpha}}{r^p} \right), \qquad \text{when } |\boldsymbol{x}| > |\boldsymbol{y}|. \tag{3.8}$$

that we derived in Section 1.2. Given a positive integer P, we can truncate the sum (3.7) to obtain an approximate kernel

$$G_P(\boldsymbol{x}, \boldsymbol{y}) = \sum_{p=-P}^{P} \left(J_p(\kappa s) e^{ip\beta} \right) \left(H_p^{(1)}(\kappa r) e^{-ip\alpha} \right) \approx G(\boldsymbol{x}, \boldsymbol{y}). \tag{3.9}$$

For a domain that is of modest size in comparison to the wave-length of the kernel, an FMM can now be constructed in a manner very similar to the FMM for the Laplace equation. There exist "translation" theorems that directly provide formulas for how to shift the centers of expansions, and for converting an outgoing to an incoming expansion.

The crucial difference between the Laplace and the Helmholtz summation problems is that the number of terms P required to attain a certain precision in the approximation (3.9) depends on the size of the box holding the sources to be compressed. The larger the box is, the more terms will be needed. This means that in the FMM, the length of the incoming and the outgoing expansions will need to grow for larger boxes. For boxes that are of size $O(1)$ wave-lengths or smaller, a fixed P will work, just like in the Laplace case. But once the boxes get larger, we will need to use expansions of length $P \sim \kappa D$, where D is the diameter of the box, just to maintain a fixed accuracy. (In Section 4.5, we illustrate this effect through a numerical experiment.)

If the basic framework of the FMM were to be kept, the growth in interaction ranks will lead to a disastrous increase in computational complexity for domains that are large in relation to the wave-length.

However, Vladimir Rokhlin made the remarkable discovery [**25, 26**] that it is possible to execute the transfer of an outgoing expansion to an incoming expansion using *diagonal* translation operators. To make this work, substantial new machinery needs to be introduced, however. The interaction list needs to be split into different groups that correspond to different directions from a source box, additional local transformations in both the source and the target box are required, and then one needs to figure out when to switch from one representation to the other to maintain numerical stability [**4, 5**].

Let us summarize the key points of this section: For domains that are small in relation to the wave-length, it is relatively straight-forward to modify the Laplace FMM to obtain a Helmholtz FMM. For larger ("high-frequency") problems, fundamental re-engineering is required. These codes are complex to implement, but work very well and provide a tool for solving a class of problems for which very few competing methods are available.

REMARK 3.3 (Helmholtz in 3D). *In three dimension, the fundamental solution of the Helmholtz equation takes the form*

$$G(\boldsymbol{x}, \boldsymbol{y}) = \frac{e^{i\kappa|\boldsymbol{x}-\boldsymbol{y}|}}{4\pi|\boldsymbol{x}-\boldsymbol{y}|}.$$

This fundamental solution admits a summation theorem of similar form to the formula (3.4) for the Laplace equation. A key difference is that the function r^n in the formula (3.2) is replaced by the spherical Bessel function $j_n(\kappa r)$, and r^{-n-1} in formula (3.3) is replaced by the spherical Hankel function $h_n(\kappa r)$. In other words, $R_n^m(\boldsymbol{x}) = j_n(\kappa r)Y_n^m(\theta, \varphi)$ and $S_n^m(\boldsymbol{x}) = h_n(\kappa r)Y_n^m(\theta, \varphi)$. Serious obstacles must be overcome to implement a high efficiency FMM in this environment. All challenges that we described for 3D Laplace FMM arise, as well as the need for diagonal translation operators to combat the increase in rank that we described for the 2D Helmholtz case. In fact, in 3D, the interaction rank between two boxes depends on the area *(as measured in square wave-lengths) of the boxes in the plane perpendicular to the line between the box centers.*

REMARK 3.4 (Directional FMM). *Another approach to recovering linear complexity in the Helmholtz FMM is to use hierarchical trees with far more boxes. By tessellating the far field with more boxes, it becomes possible to ensure that whenever an outgoing expansion is transferred from a source box Ω_σ, all target boxes fit within a cone centered on Ω_σ whose opening angle is small enough to ensure that the interaction rank is bounded. This way pure rank-deficiency turns out to be sufficient to attain close to linear complexity in the summation scheme* [**7**].

REMARK 3.5 (The modified Helmholtz equation). *By flipping the sign in front of κ^2 term in the Helmholtz equation $-\Delta u - \kappa^2 u = f$, we obtain the "modified Helmholtz equation" or "Yukawa equation" $-\Delta u + \kappa^2 u = f$. This equation has the fundamental solution $G(\boldsymbol{x}, \boldsymbol{y}) = K_0(\kappa|\boldsymbol{x}-\boldsymbol{y}|)$ where K_0 is the modified Bessel function of the second kind. It has a logarithmic singularity at the origin, just like the fundamental solutions of the Laplace and Helmholtz equations. It is a non-oscillatory function that decays exponentially fast as $|\boldsymbol{x}-\boldsymbol{y}| \to \infty$. A summation problem like*

$$u_i = \sum_{j=1}^{N} K_0(\kappa|\boldsymbol{x}-\boldsymbol{y}|)\, q_j, \qquad i = 1, 2, \ldots, N,$$

can be solved using an FMM by exploiting the separation of variables, cf. (3.8) and (3.7),

$$K_0(\kappa|\boldsymbol{x}-\boldsymbol{y}|) = \sum_{p=-\infty}^{\infty} \underbrace{\Big(I_p(\kappa s)e^{ip\beta}\Big)}_{=R_p(\boldsymbol{y})} \underbrace{\big(K_p(\kappa r)e^{-ip\alpha}\big)}_{=S_p(\boldsymbol{x})}, \qquad \text{when } |\boldsymbol{x}| > |\boldsymbol{y}|. \tag{3.10}$$

The functions I_p are the modified Bessel functions of the first kind. This variation of the FMM is formally closely related to the Helmholtz FMM described in this section. Since the kernel is in this case

non-oscillatory, ranks do not grow like they do for the standard Helmholtz equation which simplifies matters. Details on the various translation operators and other machinery that is required can be found in [**13, 15, 24**].

3.4. Practical notes and further reading

We have provided only the briefest of introductions to the vast topic of Fast Multipole Methods. A fuller treatment can be found in numerous tutorials (e.g. [**2, 17**]), survey papers (e.g. [**20**]), and full length text books (e.g. [**5, 16**]).

3.4.1. Other interaction potentials (elasticity, Stokes, etc). Variations of the FMM have been constructed for most of the kernels associated with the elliptic PDEs of mathematical physics. This text touches on the Laplace and Helmholtz cases. For material on linear elasticity, see e.g. [**8**]. The Stokes and unsteady Stokes equations are considered in [**10**]. For time-harmonic Maxwell, see [**5**]. [**16**] provides a one point reference for all of these cases.

3.4.2. Using the FMM to apply integral operators. In this text, we focused on using the FMM to evaluate discrete sums. Very similar techniques can of course be used to evaluate integrals against continuum charge distributions. To illustrate, suppose that $q(\boldsymbol{y})$ is a given charge distribution in a domain Ω, which generates a harmonic potential

$$u(\boldsymbol{x}) = \int_{\Omega} \log|\boldsymbol{x}-\boldsymbol{y}|\, q(\boldsymbol{y})\, dA(\boldsymbol{y}).$$

Let $\boldsymbol{c}$ denote the center of Ω. Then for any target point $\boldsymbol{x}$ that lies outside of a disc centered on $\boldsymbol{c}$ that contains Ω, we can write the potential u as a sum of the form (1.13), with the multipole coefficients $\hat{q}_j$ defined via the formulas

$$\begin{aligned}
\hat{q}_0 &= \int_{\Omega} q(\boldsymbol{y})\, dA(\boldsymbol{y}),\\
\hat{q}_{2p-1} &= \int_{\Omega} q(\boldsymbol{y})\, C_{2p-1}(\boldsymbol{y},\boldsymbol{y}_{\mathrm{c}})\, dA(\boldsymbol{y}) = -\int_{\Omega} q(\boldsymbol{y})\, (r')^p \frac{1}{p}\cos(p\,\theta')\, dA(\boldsymbol{y}),\\
\hat{q}_{2p} &= \int_{\Omega} q(\boldsymbol{y})\, C_{2p}(\boldsymbol{y},\boldsymbol{y}_{\mathrm{c}})\, dA(\boldsymbol{y}) = -\int_{\Omega} q(\boldsymbol{y})\, (r')^p \frac{1}{p}\sin(p\,\theta')\, dA(\boldsymbol{y}).
\end{aligned}$$

Formulas of this type become particularly useful in the context of rapid application of boundary integral operators.

3.4.3. Kernel free FMMs. While FMMs can be developed for a broad range of kernels (cf. Section 3.4.1), it is quite labor intense to re-derive and re-implement the various translation operators required for each special case. The so called *kernel independent FMMs* [**30, 9**] overcome this difficulty by setting up a common framework that works for a broad range of kernels. The idea is to compute translation operators numerically by either performing low-rank approximation of certain full operators, or to solve least square systems exploiting physical knowledge that informs us that certain factorizations must exist.

3.4.4. Matrix operations beyond the matrix-vector product. The FMM performs a matrix-vector multiply $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$ involving certain dense $N \times N$ matrices in $O(N)$ operations. It achieves the lower complexity by exploiting rank-deficiencies in the off-diagonal blocks of the matrix $\mathbf{A}$. It turns out that such rank deficiencies can also be exploited to perform other matrix operations, such as matrix inversions, construction of Schur complements, LU factorizations, etc, in close to linear time. The so called $\mathcal{H}$-matrix methods [**14**] provide a general framework that can be applied in many contexts. Higher efficiency can be attained by designing direct solvers specifically for the linear systems arising upon the discretization of certain boundary integral equations [**19**]. A variation of these schemes is the *inverse FMM*, see [**28**].

3.4.5. High quality codes. Let us close with a practical note. For problems with non-oscillatory kernels, it is in general not that daunting of an endeavor to implement an FMM with linear or close to linear asymptotic scaling. However, it is another matter entirely to write a code that actually achieves high *practical* performance. This would be an argument against using FMMs were it not for the fact that the algorithms are very well suited for black box implementation. Some such codes are available publicly, and more are expected to become available in the next several years. Before developing a new code from scratch, it is usually worth-while to first look to see if a high-quality code may already be available.

CHAPTER 4

The potential evaluation map

In this chapter, we return to the potential evaluation problem that we described in Section 1.1. In this problem, we were given one set of source locations and a different set of target locations, and we are interested in the matrix that maps a vector of source strengths to a vector of potentials. We proved in Section Section 1.2 that the singular values of this matrix decay exponentially fast by analyzing an analytic separation of variables given by 2D multipole expansions. Let us now ask the following questions: How close to optimal is the factorization provided by a multipole expansion? Is it possible to improve on it? If the answer is yes, then can this fact be used to accelerate the FMM?

We start the discussion by considering the Laplace problem in Sections 4.1 — 4.4, and then briefly discuss the Helmholtz equation in Section 4.5.

4.1. Compression with optimal rank

Let us start with investigating exactly the same geometry that we started with in Section 1.1 and that was shown in Figure 1.1. This geometry involves two separate boxes Ω_σ and Ω_τ that hold source points $\{\boldsymbol{y}_j\}_{j=1}^n$ and target points $\{\boldsymbol{x}_i\}_{i=1}^m$, respectively. We then formed the $m \times n$ matrix $\mathbf{A}$ with entries $\mathbf{A}(i,j) = \log|\boldsymbol{x}_i - \boldsymbol{y}_j|$ and in Section 1.2 we analytically built an approximate factorization, cf. (2.8),

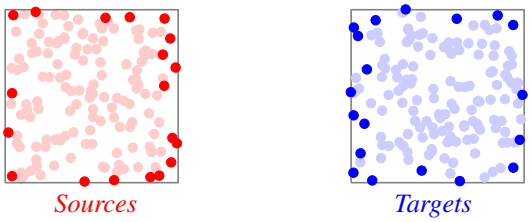
$$\underset{m\times n}{\mathbf{A}} \quad = \quad \underset{m\times k}{\mathbf{T}^{\text{tfo}}} \quad \underset{k\times n}{\mathbf{T}^{\text{ofs}}},$$

where k is a tuning parameter that trades accuracy for cost. Deriving a factorization via analysis is very appealing in part because it provides assurance that the method will always work reliably. Perhaps more importantly, however, having an analytic formula dispenses with the need to do any work to find the factorization; the formula provides simple analytic expressions for all the entries of the factors $\mathbf{T}^{\text{tfo}}$ and $\mathbf{T}^{\text{ofs}}$. The drawback is that the factorization is not necessarily close to optimal. To investigate this, let us for the particular distribution of points shown in Figure 4.1(a) form the actual matrix $\mathbf{A}$, and then compare the approximation errors obtained when computing a rank-k approximation $\mathbf{A}_k$ to $\mathbf{A}$ using three difference techniques, the FMM, the SVD, and the interpolative decomposition (ID) that was described in Section ??. In Figure 4.1(b) we plot the approximation errors

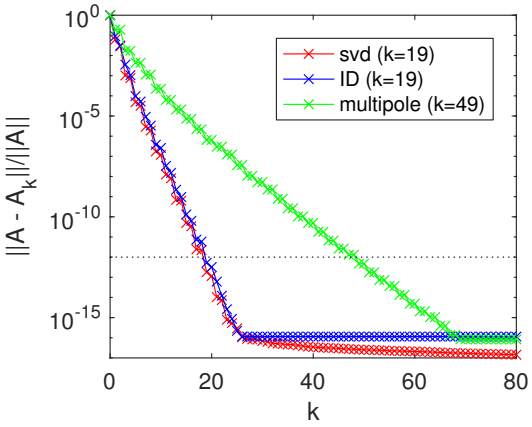
$$e_k = \frac{\|\mathbf{A} - \mathbf{A}_k\|}{\|\mathbf{A}\|}$$

for the three different methods as a function of k.[1] We observe that the SVD and the ID perform *far* better than the multipole expansion in this case. There are at least two reasons why this comparison may not be fair to the multipole expansion. First, the multipole expansion is valid in a much larger domain of target points, not just in the particular box considered here. Second, the multipole expansion works for any distribution of points in the two boxes, not just the particular one that we happened to choose.

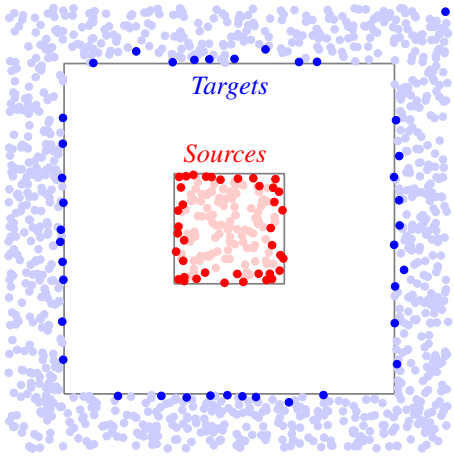
[1] The rank k that we use for the multipole expansion is the rank resulting from real valued arithmetic. In other words, each term in the complex expansion corresponds to two real valued functions, since $(\boldsymbol{x}-\boldsymbol{c})^{-p} = \frac{\cos(p\theta)}{r^p} + i\frac{\sin(p\theta)}{r^p}$ using polar coordinates, cf. (1.18). This means that a multipole expansion of order P corresponds to a real-valued factorization of rank $k = 1 + 2P$.



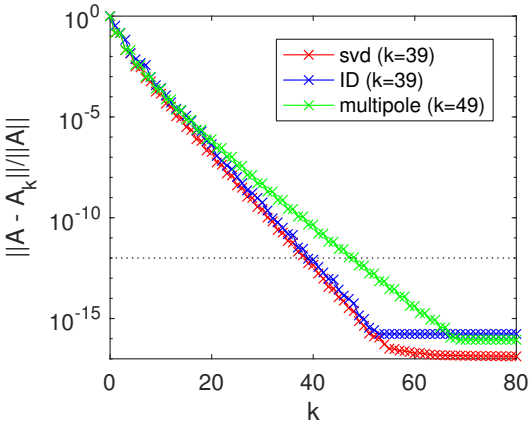
(a)



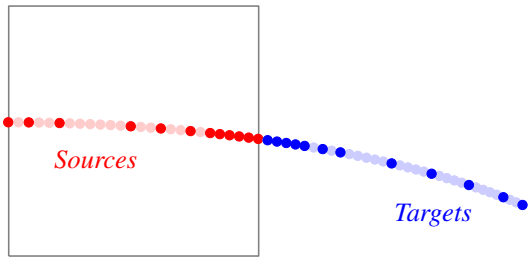
(b)



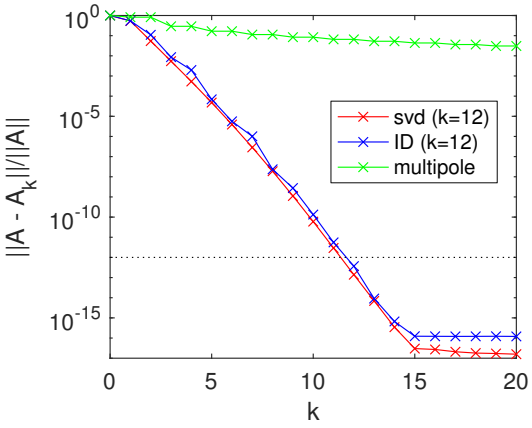
(c)



(d)



(e)



(f)

FIGURE 4.1. Actual ranks of some representative potential evaluation matrices, as discussed in Section 4.1. The left column shows three representative geometries, and the right column shows the corresponding approximation accuracies, using three different matrix compression techniques. The matrix is in each case $\mathbf{A}(i,j) = \log|\boldsymbol{x}_i - \boldsymbol{y}_j|$. The ranks listed in the legends are to relative precision $\varepsilon = 10^{-12}$. The darker source and target points are the ones chosen by the ID; these form what we call the *skeleton points*.

To investigate the importance of the locations of the target points, let us next consider a situation where the target points are distributed along all directions from the source box, but still well-separated from the source box, as shown in Figure 4.1(c). The resulting approximation errors in Figure 4.1(d) are very illuminating. The errors in the multipole expansion remain basically the same, which is what we would expect since all the target points as just as far away from the source box as in the geometry in Figure 4.1(a). But now the SVD and the ID perform far worse, and in fact are not too far off from the ranks produced by the multipole expansion.

The fact that the SVD and the multiple expansions are quite close in Figure 4.1(d) indicate that the second concern in regards to the fairness of comparing the SVD to an analytic expansion seems less likely to hold much water. As it happens, for a relatively dense distribution of sources, there is not much difference between the ranks generated for a particular realization, and the ranks generated by an optimal "generic" compression. We will discuss this point further in Sections 4.2 and 4.3.

For particle distributions that are not uniform in a box, it turns out to be possible to obtain far lower ranks via numerical compression than what you would get from a simple analytic formula such as a multipole expansion. To illustrate the point, let us consider the geometry shown in Figure 4.1(e). Here both source and target points are distributed along a curve. The two sets are hardly separated at all, so it is unsurprising that the multipole expansion converges very slowly, as shown in Figure 4.1(f). In contrast, the actual numerical rank of $\mathbf{A}$, as detected by the SVD, is in this case exceedingly low.

Let us finally comment on how the SVD and the ID compare to each other. In all three of the error plots in Figures 4.1(b,d,f), the ID results in errors that are only a hair larger than the theoretically optimal ones resulting from the SVD. This is in some sense unsurprising, given how rapidly the singular values of the matrices decay. However, we will in later chapters often work with problems where the ID has powerful advantages over the SVD, and it is very encouraging to see how close to optimal it is.

4.2. Proxy surfaces

Consider a box Ω_σ that holds some source points, a shown in Figure 4.2, and suppose that we seek to build a compressed representation of the field v that is generated by these sources. For the compressed representation to be useful in a fast summation scheme like the FMM, it must be accurate at *any* target point that is well-separated from Ω_σ. This may at first seem like a daunting task; how could we ever test every possible point? There is a surprisingly simple way out of this quandary, however. All we need to do is to observe that if we build a harmonic field v_{approx} that matches the exact field v on the *boundary* of the region of well-separated points, then the field v_{approx} automatically matches u at all well-separated points. This observation follows immediately from the uniqueness of solutions to the Laplace equation with Dirichlet data: Let Γ_{proxy} denote the boundary of the set of well-separated points, as shown with the green dashed line in Figure 4.2. Then observe that the field $w = v - v_{\text{approx}}$ is harmonic and satisfies the correct decay condition at infinity. If $w = 0$ on Γ_{proxy}, then w must be identically zero everywhere outside of Γ_{proxy}.

4.3. Compression of continuum operators

In Section 4.1 we investigated the numerical ranks of a number of matrices that each was derived from some particular choice of source and target points. We will next describe how to build matrices that represent the intrinsic geometry of the source and the target domains, rather than particular samples from them. Given a source region Ω_σ and a target region Ω_τ, the idea is to consider the continuum operator

$$A:\ L^2(\Omega_\sigma) \rightarrow L^2(\Omega_\tau):\ q \mapsto f(\boldsymbol{x}) = \int_{\Omega_\sigma} \log|\boldsymbol{x}-\boldsymbol{y}|\, q(\boldsymbol{y})\, d\boldsymbol{y} \tag{4.1}$$

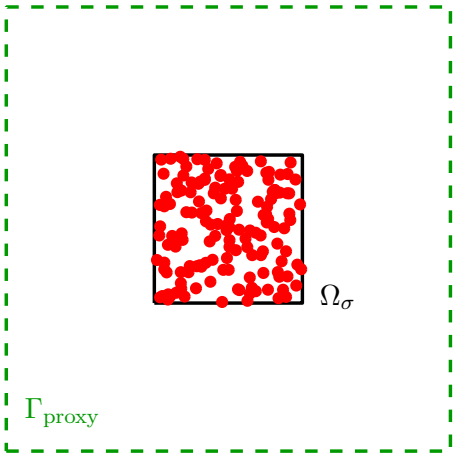


FIGURE 4.2. Geometry considered in Section 4.2. A box Ω_σ holds a number of source locations and generate a harmonic field v outside the box. If we can generate a harmonic field v_{approx} that matches v closely on the contour Γ_{proxy} then v and v_{approx} will closely match *everywhere* outside of Γ_{proxy}.

that maps a continuum source distribution q on Ω_σ to a continuum potential f on Ω_τ. When the two regions are separated by some minimal distance $d_{\min} > 0$, the kernel in the integral operator A is smooth, and can readily be approximated using numerical quadratures. To this end, let $\{\boldsymbol{x}_i, v_i\}_{i=1}^m$ be a quadrature for the target domain, and let $\{\boldsymbol{y}_j, w_j\}_{j=1}^n$ be a quadrature for the source domain. It is then convenient to scale the vectors that represent q and f by the square roots of the corresponding quadrature weights. In other words, we define $\mathbf{q} \in \mathbb{R}^n$ and $\mathbf{v} \in \mathbb{R}^m$ via

$$\begin{aligned} \mathbf{f}(i) &= \sqrt{v_i}\, f(\boldsymbol{x}_i), \\ \mathbf{q}(j) &= \sqrt{w_j}\, q(\boldsymbol{y}_j). \end{aligned}$$

The advantage is that then

$$\|\mathbf{f}\|_{\ell^2}^2 = \sum_{i=1}^m |\mathbf{f}(i)|^2 = \sum_{i=1}^m v_i |f(\boldsymbol{x}_i)|^2 \approx \int_{\Omega_\tau} |f(\boldsymbol{x})|^2 \, d\boldsymbol{x} = \|f\|_{L^2(\Omega_\tau)}^2,$$

with an analogous calculation showing that $\|\mathbf{q}\|_{\ell^2} \approx \|q\|_{L^2(\Omega_\sigma)}$. Finally, we let $\mathbf{A}$ denote the $m \times n$ matrix with entries

$$\mathbf{A}(i,j) = \sqrt{v_i}\, \log|\boldsymbol{x}_i - \boldsymbol{y}_j|\, \sqrt{w_j}. \tag{4.2}$$

Then the singular values/vectors of $\mathbf{A}$ are accurate approximations of the singular values/vectors of A.

REMARK 4.1 (Max-norm approximation). *The perhaps most desirable approximation of the continuum operator A would be one that is accurate pointwise everywhere, so that given some tolerance ε, we would build basis functions $\{S_p\}_{p=1}^P$ and $\{R_p\}_{p=1}^P$ such that*

$$\sup_{\boldsymbol{x}\in\Omega_\tau,\, \boldsymbol{y}\in\Omega_\sigma} \left| G(\boldsymbol{x},\boldsymbol{y}) - \sum_{p=1}^P R_p(\boldsymbol{y})\, S_p(\boldsymbol{x}) \right| \leq \varepsilon. \tag{4.3}$$

While a bound such as (4.3) would perhaps be preferable, we work with L^2 norms since they are computationally far more tractable, and allow the low-rank approximation problem to be resolved optimally using the SVD. In contrast, finding close to optimal basis functions such that (4.3) holds can be challenging. Heuristic methods can work very well in practice [**29**] *but come with no guarantee of optimality. More sophisticated methods* [**21**] *would likely lead to lower ranks.*

4.4. Three case studies for Laplace's equation

In this section, we will numerically investigate the singular values of the integral operator (4.1) that maps a source distribution q on a source domain Ω_σ to a potential f on a target domain Ω_τ. We consider the kernel function $G(\boldsymbol{x}, \boldsymbol{y}) = \log|\boldsymbol{x} - \boldsymbol{y}|$ associated with the Laplace equation, and the three geometries shown in the left column of Figure 4.3. The red and the blue dots in the figures show the quadrature points chosen by the ID on Ω_σ and Ω_τ, respectively.

Let us first compare the optimal accuracies resulting from the SVD (red lines in Figures 4.3(b,d,f)) to the accuracies resulting from the analytic formula provided by the multipole expansions (green lines). The first geometry that we consider, shown in Figure 4.3(a), consists of a disc of sources which are mapped to a concentric circle of target ponts. This geometry is the one that multipole expansions are designed for, and it is not surprising to see that in this case, there is no visible difference between the SVD and the multipole errors. The geometry shown in Figure 4.3(c) is the one of relevance in the 2D FMM. Here we see that the multipole expansions do result in larger errors for any given separation rank, but that the difference to the SVD is small enough that using multipole expansions in the FMM is well justified. (Still, at accuracy 10^{-12}, the ranks are 45 and 53 respectively, which is not entirely negligible given that the cost of the FMM scales as the rank squared. This difference provides an opening to optimize the speed by switching to numerically determined translation operators as described in Section 3.4.3.) In our final geometry in Figure 4.3(e) we consider a geometry where the targets are much close to the box of sources. In this case, some of the target points are in fact *not* well-separated from the source box, and in consequence, the multipole expansion diverges, as we see in Figure 4.3(f). It is of interest however that the interaction is still of low numerical rank, and in fact increases only from 45 in the previous example to 81 when the two boxes are very close.

Next we consider the difference between the errors results from the SVD (red lines) and the interpolatory decomposition (blue lines). The key observation is that in all the three geometries we consider, the difference is marginal, and we may freely chose either format as there is almost no difference in the resulting ranks. The one case where there is a noticeable difference is the last one, which is to be expected since the singular values decay more slowly in this case.

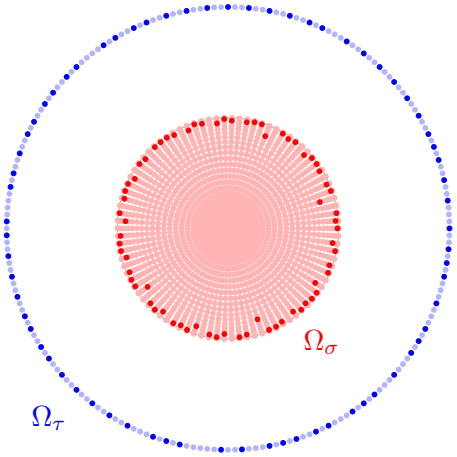
In Figures 4.3(a,c,e), the points chosen by the ID in the source and the target domains are drawn with a darker color. We observe that the dark red points are *universal* skeleton points. This means that given any source distribution inside Ω_σ, we can replicate the field these sources generate on Ω_τ to precision $\varepsilon = 10^{-12}$ by placing some equivalent sources at these dark red points. In order to carry out this computation in practice, we would use the computed right singular vectors of $\mathbf{A}$ to tell us what the shape of the relevant basis functions should be, using the interpolation implicit in the quadrature rule to create a continuum function from the sampled values identified in the singular vector.

REMARK 4.2. *Let us linger slightly on the geometry in Figure 4.3(a) where the source region is a disc, and the target region is a circle that is concentric to the disc. We see in Figure 4.3(b) that the multipole expansion in this case appears to be about as accurate as the SVD, which is theoretically optimal. Since the graph is small, it is difficult to see just how close they are, so let us look at a zoomed picture in Figure 4.4. We see that the accuracies still appear to be identical. It turns out that in this case, the harmonic basis functions are actually the right singular vectors associated with the continuum operator that are associated with non-zero singular values.*

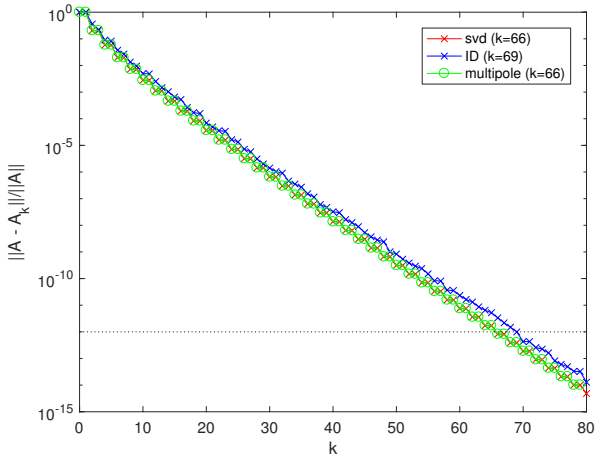
4.5. Three case studies for the Helmholtz equation

Having studied the potential evaluation map associated with the Laplace equation in Section 4.4, let us next turn our attention to the analogous integral operator

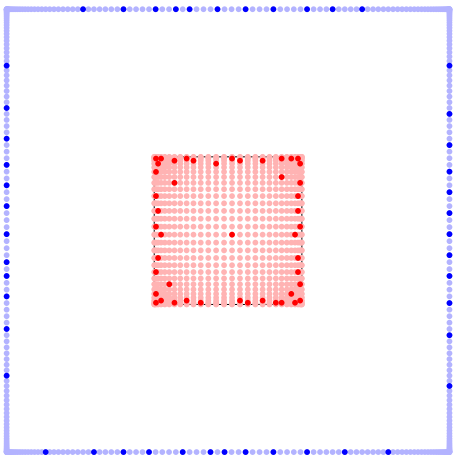
$$A:\ L^2(\Omega_\sigma) \to L^2(\Omega_\tau):\ q \mapsto f(\boldsymbol{x}) = \int_{\Omega_\sigma} H_0^{(1)}(\kappa|\boldsymbol{x} - \boldsymbol{y}|)\, q(\boldsymbol{y})\, d\boldsymbol{y} \tag{4.4}$$



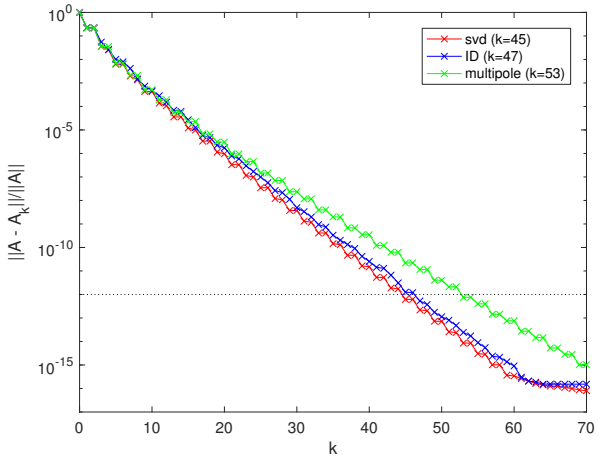
(a)



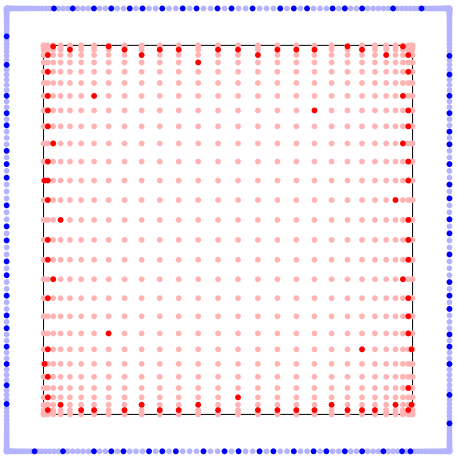
(b)



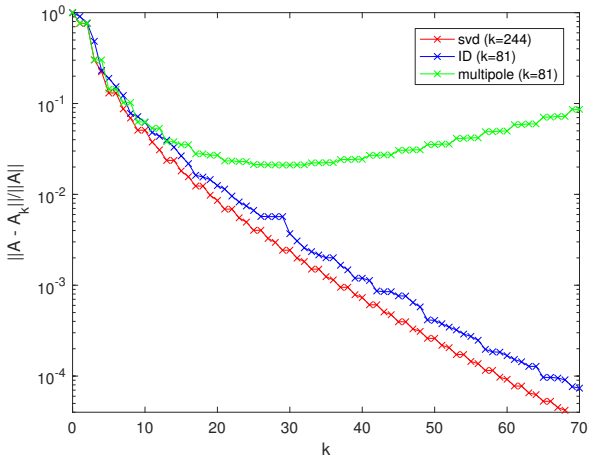
(c)



(d)



(e)



(f)

FIGURE 4.3. Ranks of the potential evaluation integral operators described in Section 4.4. The left column shows the three geometries, and the right column shows the corresponding approximation accuracies. The ranks listed in the legends are to relative precision $\varepsilon = 10^{-12}$. The darker source and target points are the skeleton points chosen by the ID.

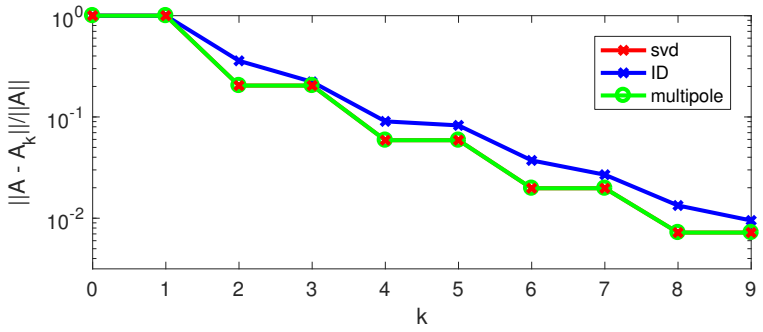
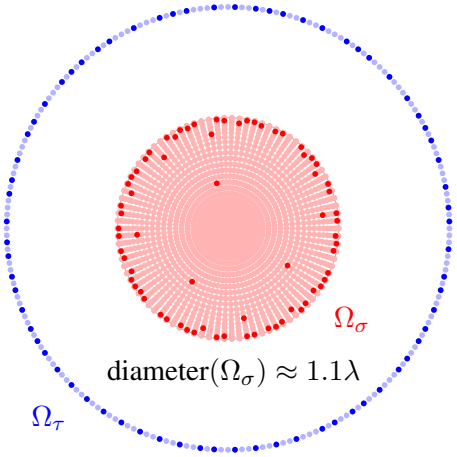


FIGURE 4.4. A close-up of the accuracy graph in Figure 4.3(b). We see that the errors from the SVD and the multipole expansion are identical. In fact, the multipole expansion *is* the continuum SVD in this case, cf. Remark 4.2.

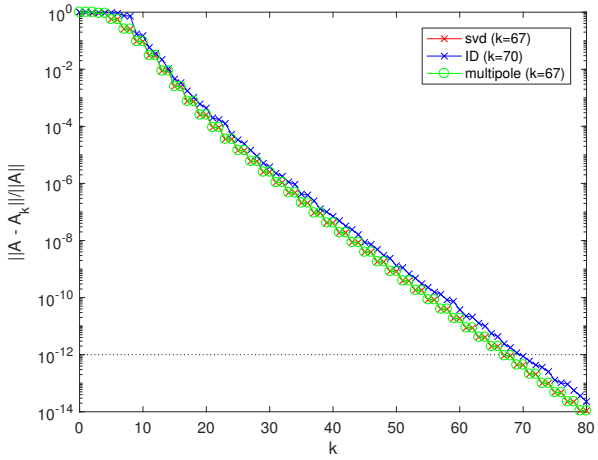
that is associated with the Helmholtz equation $-\Delta u - \kappa^2 u = f$. We consider the three geometries shown in Figures 4.5(a,c,e).

Let us first consider the rotationally symmetric geometry in Figure 4.5(a). This is the same geometry as in Figure 4.3(a), and we in this case choose the wave number κ so that the domain is just over 1 wave-length in diameter. We again see that in this case, the multipole expansion and the SVD give identical errors, just as for the Laplace case. We next consider the FMM geometry in Figure 4.5(c). The box with sources is around one and a half wave-length across, and we see that in this case, the accuracies behave in a manner entirely analogous to the Laplace case — there is some discrepancy between the multipole expansion and the SVD, but they are close. In the final example in Figure 4.5(e), we keep the geometry the same, but increase κ so that the domain is now about 11 wavelengths across. In this situation, the errors plots look quite different. We see that we get no accuracy whatsoever until the interaction ranks reach about 60, and then the exponential decay sets in. This is a result of the fact (cf. Section 3.3) that for the Helmholtz case, the interaction rank depends on the size of the domain in wave-lengths. Until P is large enough to resolve the source object to within the precision dictated by the wavelength, the expansion (3.9) will have no accuracy. Once the object is resolved, the accuracy improves rapidly.

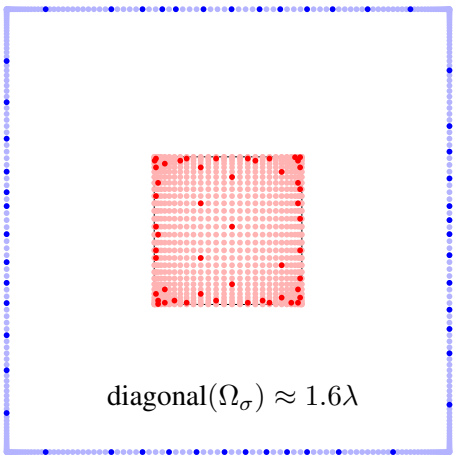
Let us next consider the accuracy from the ID in comparison to the accuracy of the SVD. We see that just as for the Laplace case, there is generally very little difference between the two. Using the ID leads to very slightly higher interaction ranks, but from a practical point of view, the ranks are essentially the same. One interesting observation is that for the short wave-length case in Figure 4.5(e), the ID selects plenty of points from the *interior* of the domain. In all prior examples, the points tended to cluster along the boundary of the domain, despite the fact that these points are weighted unfavorably by the quadrature scaling in (4.2). (When this scaling is not included, basically *all* skeleton nodes are picked right on the boundary in the Laplace case.)



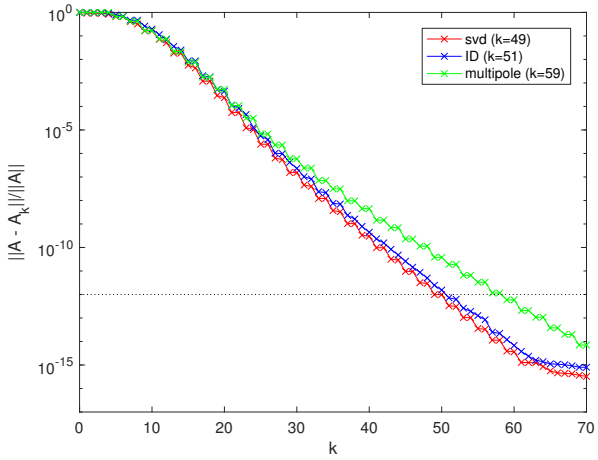
(a)



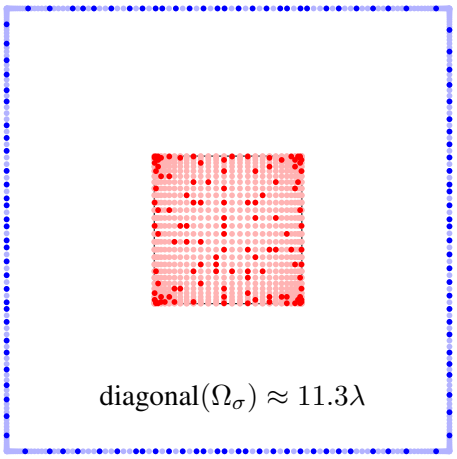
(b)



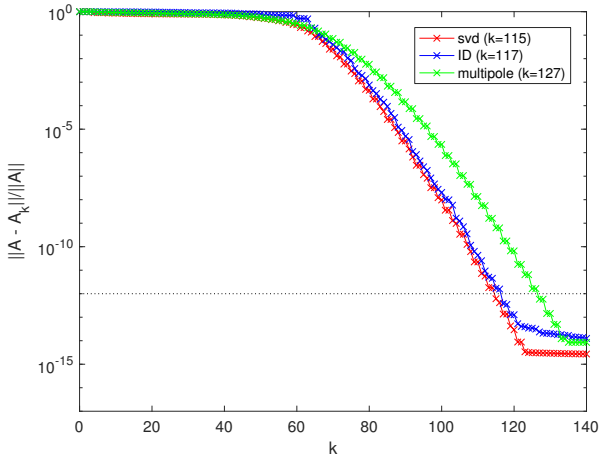
(c)



(d)



(e)



(f)

FIGURE 4.5. Ranks of the potential evaluation integral operators described in Section 4.5 involving the Helmholtz kernel $G(\boldsymbol{x}, \boldsymbol{y}) = H_0^{(1)}(\kappa|\boldsymbol{x}-\boldsymbol{y}|)$. The left column shows the three geometries, and the right column shows the corresponding approximation accuracies. The ranks listed in the legends are to relative precision $\varepsilon = 10^{-12}$. The darker source and target points are the skeleton points chosen by the ID.

Bibliography

[1] J. Barnes and P. Hut, *A hierarchical $o(n \log n)$ force-calculation algorithm*, Nature **324** (1986), no. 4.

[2] L. Beatson and L. Greengard, *A short course on fast multipole methods*, 1997.

[3] J Carrier, L Greengard, and V Rokhlin, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. SCI. STAT. COMPUT. **9** (1988), no. 4, 669–686.

[4] Hongwei Cheng, William Y. Crutchfield, Zydrunas Gimbutas, Leslie F. Greengard, J. Frank Ethridge, Jingfang Huang, Vladimir Rokhlin, Norman Yarvin, and Junsheng Zhao, *A wideband fast multipole method for the Helmholtz equation in three dimensions*, J. Comput. Phys. **216** (2006), no. 1, 300–325.

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

[6] E. Darve, *The fast multipole method I: Error analysis and asymptotic complexity*, SIAM J. NUMER. ANAL. **38** (2000), no. 1, 98 – 128.

[7] B. Engquist and L. Ying, *A fast directional algorithm for high frequency acoustic scattering in two dimensions*, Commun. Math. Sci. **7** (2009), no. 2, 327–345.

[8] Y. Fu, K. J. Klimkowski, G. J. Rodin, E. Berger, J. C. Browne, J. K. Singer, R. A. Van De Geijn, and K. S. Vemaganti, *A fast solution method for three-dimensional many-particle problems of linear elasticity*, International Journal for Numerical Methods in Engineering **42** (1998), no. 7, 1215–1229.

[9] Zydrunas Gimbutas and Vladimir Rokhlin, *A generalized fast multipole method for nonoscillatory kernels*, SIAM J. Sci. Comput. **24** (2002), no. 3, 796–817 (electronic). MR MR1950512 (2004a:65176)

[10] L. Greengard and M. C. Kropinski, *An integral equation approach to the incompressible navier-stokes equations in two dimensions*, SIAM J. Sci. Comput. **20** (1998), no. 1, 318–336.

[11] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys. **73** (1987), no. 2, 325–348.

[12] Leslie Greengard and Vladimir Rokhlin, *A new version of the fast multipole method for the Laplace equation in three dimensions*, Acta numerica, 1997, Acta Numer., vol. 6, Cambridge Univ. Press, Cambridge, 1997, pp. 229–269.

[13] Leslie F. Greengard and Jingfang Huang, *A new version of the fast multipole method for screened coulomb interactions in three dimensions*, Journal of Computational Physics **180** (2002), no. 2, 642 – 658.

[14] Wolfgang Hackbusch, *A sparse matrix arithmetic based on H-matrices; Part I: Introduction to H-matrices*, Computing **62** (1999), 89–108.

[15] Jingfang Huang, Jun Jia, and Bo Zhang, *Fmm-yukawa: An adaptive fast multipole method for screened coulomb interactions*, Computer Physics Communications **180** (2009), no. 11, 2331 – 2338.

[16] Y. J. Liu, *Fast multipole boundary element method: Theory and applications in engineering*, Cambridge University Press, 2009.

[17] Y.J. Liu and N. Nishimura, *The fast multipole boundary element method for potential problems: A tutorial*, Engineering Analysis with Boundary Elements **30** (2006), no. 5, 371 – 381.

[18] Brock A. Luty, Ilario G. Tironi, and Wilfred F. van Gunsteren, *Lattice-sum methods for calculating electrostatic interactions in molecular simulations*, The Journal of Chemical Physics **103** (1995), no. 8, 3014–3021.

[19] P.G. Martinsson and V. Rokhlin, *A fast direct solver for boundary integral equations in two dimensions*, J. Comp. Phys. **205** (2005), no. 1, 1–23.

[20] N. Nishimura, *Fast multipole accelerated boundary integral equation methods*, Applied Mechanics Reviews **55** (2002), no. 4, 299 – 324.

[21] Ricardo Pachón and Lloyd N. Trefethen, *Barycentric-remez algorithms for best polynomial approximation in the chebfun system*, BIT Numerical Mathematics **49** (2009), no. 4, 721.

[22] H.G. Petersen, E.R. Smith, and D. Soelvason, *Error estimates for the fast multipole method. i. the two-dimensional case*, Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences **448** (1995).

[23] _____, *Error estimates for the fast multipole method. ii. the three-dimensional case*, Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences **448** (1995).

[24] Bryan Douglas Quaife, *Fast integral equation methods for the modified helmholtz equation*, Ph.D. thesis, Science: Department of Mathematics, 2011.

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

[26] ———, *Diagonal forms of translation operators for the helmholtz equation in three dimensions*, Applied and Computational Harmonic Analysis **1** (1993), no. 1, 82 – 93.
[27] Dorth Sølvason and Henrik G. Petersen, *Error estimates for the fast multipole method*, Journal of Statistical Physics **86** (1997), no. 1, 391–420.
[28] Toru Takahashi, Pieter Coulier, and Eric Darve, *Application of the inverse fast multipole method as a preconditioner in a 3d helmholtz boundary element method*, Journal of Computational Physics **341** (2017), 406–428.
[29] A. Townsend and N. Trefethen, *Gaussian elimination as an iterative algorithm*, SIAM News **46** (2013), no. 02.
[30] L. Ying, G. Biros, and D. Zorin, *A kernel-independent adaptive fast multipole algorithm in two and three dimensions*, J. Comput. Phys. **196** (2004), no. 2, 591–626.