A Brief Introduction to the Fast Multipole Method

February 28, 2018

1 Introduction

This paper will present the original two-dimensional fast multipole method (FMM), a fast algorithm to compute electrostatic interactions between a collection of particles. More generally, FMM algorithms exist to treat particles interacting under many types of potentials, and therefore have found applications in many fields. In simulations of galaxy formation, for example, one must repeatedly calculate the mutual gravitational forces between a large set of bodies, and in simulations of molecular dynamics or quantum many-body problems, one must calculate electrostatic forces between charged particles. Indeed, many of the systems arising in physics may be interpreted in terms of particle-particle interactions. Our main motivation is the use of the FMM to accelerate schemes which solve PDEs numerically using the theory of integral equations, and we will briefly present a representative example of such a problem in order to illustrate the importance of the FMM in solving problems of this type.

We begin by stating the computational problem which the FMM addresses. The Coulomb kernel in 2D is the fundamental solution of Laplace's equation,

$$\Phi(\mathbf{x}, \mathbf{x}') = \frac{1}{2\pi} \log \frac{1}{|\mathbf{x} - \mathbf{x}'|}$$

and may be interpreted as a two-dimensional version of the electrostatic potential induced at a position $\mathbf{x} \in \mathbb{R}^2$ by a source charge of unit strength located at $\mathbf{x}'$. Given a collection of N charged particles of charge strengths $q_1, \dots, q_N$, located at positions $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^2$, we wish to calculate, for each $i = 1, \dots, N$, the potential induced by all of the other charges at the point $\mathbf{x}_i$. This potential is given by superposition as

$$\sum_{j \neq i} q_j \Phi(\mathbf{x}_i, \mathbf{x}_j). \tag{1}$$

We must therefore compute N sums of N terms each, which together yield the potential on each particle due to all pairwise particle-particle interactions via the kernel Φ. If we were to replace Φ by its negative gradient, this calculation would give the electrostatic force on each particle due to all of the other particles. The FMM algorithm to compute these forces is not significantly different from the one we present here for the potential values, and the potential is the fundamental unknown quantity in electrostatic calculations, so we use the scalar case as our model.

The computation of these N sums may be interpreted as the multiplication of a matrix with entries $\Phi(\mathbf{x}_i, \mathbf{x}_j)$ by a vector with entries q_j. In total, there are N^2 terms, and the cost of the brute force calculation of the sums scales with N as $\mathcal{O}(N^2)$. The FMM is an algorithm which, given an error tolerance ε, computes the sums at a cost scaling as $\mathcal{O}(N)$, with an error guaranteed to be less than ε. This reduction in computational complexity is crucial for a variety of applications involving particle-particle interactions, much like the fast Fourier transform leads to the availability of highly advantageous methods for periodic PDE problems.

FMM algorithms are available for a broader class of problems than this, but all share a set of ideas which are illustrated by the simple version discussed here. We could replace the point charges by point dipoles; we could involve a self-interaction term in the sum; we could replace the charge strengths q_j with weights q_{ij} depending on both of the particles involved in an interaction; we could evaluate the potential at different locations than those of the particles; these are largely trivial changes which would lead to slightly different versions of the FMM algorithm but would keep the main features intact. Further modifications, like moving from 2D to 3D, or replacing the Coulomb kernel with one associated with another PDE, such as the Helmholtz kernel, require more dramatic changes to the algorithm. In general, however, the ideas associated with the FMM tend to be useful for computing pairwise interactions between sets of particles, or bodies, mediated by certain types of kernels.

Broadly speaking, the asymptotic speedup of the FMM is achieved by

1. The hierarchical organization of particles into groups which, at a given level of refinement, are well-separated;

2. The use of high-order, analytical far-field and local expansions to efficiently represent potentials due to groups of particles; and

3. The use of translation operators to manipulate these expansions.

The meaning of each of these points will become clearer once we describe the algorithm. The algorithm is adaptive, so that it can handle highly non-uniform distributions of particles without performance degradation. It also guarantees a user-provided accuracy. With the partial exception of certain modern variants, another feature of FMM algorithms are that they make heavy use of analytical features of the kernel involved - in the present case, the Coulomb kernel - and so provide a specific solution to a specific problem. In general, FMMs for different physical problems require their own mathematical analysis.

The main reference for this paper is the original article [1] of Greengard and Rokhlin on the FMM, and Section 3 will largely follow the treatment of the algorithm there. However, to provide motivation and context, we begin first with a brief discussion of a simple integral equation, the discretization of which will lead to a computational problem similar to the one introduced above.

2 Integral Equation for the Interior Laplace Dirichlet Problem

This section will give a quick introduction to the reformulation of the interior Laplace Dirichlet problem as an integral equation, which leads to numerical schemes involving the discretization of integral rather than differential operators. This is a classical topic which has seen a resurgence of popularity due to the development of numerical tools, in particular the FMM, which allow integral equation methods to be competitive with other schemes and to offer unique advantages for certain classes of problems. A classical treatment for the 3D case is given in Kellogg [2]; another introduction for the 3D cases is given in Guenther and Lee [3]; a more complete theoretical discussion which also touches on numerical topics is given in Kress [4]. Here, we will only give a taste of the topic as motivation for the importance of the FMM in solving PDE problems using integral equation methods, and we will gloss over all theoretical details.

The interior Laplace Dirichlet problem in 2D is stated as follows. Let Ω be a bounded domain in $\mathbb{R}^2$ with boundary Γ. We will assume Γ is sufficiently regular. Let f be a continuous function on Γ. Then the problem is to find a function $u(x)$ on Ω satisfying

$$\begin{cases} \Delta u = 0, & x \in \Omega \\ u = f, & x \in \Gamma. \end{cases} \tag{2}$$

The function u may represent, for example, the electrostatic potential in the charge-free cavity Ω given a fixed imposed potential f on Γ. It is straightforward to prove that solutions of the boundary value problem (BVP) are unique using, for example, the maximum principle or Green's identities. The existence of solutions may be proven by a variety of methods, one of which is the method of integral equations which we now outline.

Following the electrostatics interpretation, the idea is to find a charge or dipole density on Γ for which the induced potential obeys the given boundary data. In particular, representing u in terms of a dipole density leads to the cleanest theoretical analysis, and well-conditioned discretizations. We write u as a double layer potential,

$$u(x) = \int_\Gamma \frac{\partial \Phi(\mathbf{x}, \mathbf{x}')}{\partial n_{\mathbf{x}'}} \sigma(\mathbf{x}') dS(\mathbf{x}') \tag{3}$$

where $n_{\mathbf{x}'}$ is the unit outward normal vector to Γ at $\mathbf{x}'$. This is simply the electrostatic potential induced by a dipole density σ aligned with the outward normals on Γ.

Since $\Phi(\mathbf{x}, \mathbf{x}')$ is harmonic in $\mathbf{x}$ for $\mathbf{x} \neq \mathbf{x}'$, taking the Laplacian of this representation shows that u is automatically harmonic in Ω. Furthermore, it can be shown that the double layer potential u satisfies the following condition describing its limiting value as $\mathbf{x} \to \mathbf{x}_0$ for $\mathbf{x}_0 \in \Gamma$,

$$\lim_{\mathbf{x} \to \mathbf{x}_0} u(\mathbf{x}) = -\frac{\sigma(\mathbf{x}_0)}{2} + \int_\Gamma \frac{\partial \Phi(\mathbf{x_0}, \mathbf{x}')}{\partial n_{\mathbf{x}'}} \sigma(\mathbf{x}') dS(\mathbf{x}')$$

where the limit is taken along the direction normal to Γ, and the integral may be shown to exist. This condition is related to the discontinuity of an electric field across a material interface. If the double layer potential u is to satisfy the Dirichlet condition $u = f$ on Γ, according to this condition we must have

$$-\frac{\sigma(\mathbf{x})}{2} + \int_\Gamma \frac{\partial \Phi(\mathbf{x}, \mathbf{x}')}{\partial n_{\mathbf{x}'}} \sigma(\mathbf{x}') dS(\mathbf{x}') = f(\mathbf{x}) \tag{4}$$

for $\mathbf{x} \in \Gamma$. This is an integral equation which must be solved for the unknown dipole density σ on Γ. If σ satisfies (4), then the double layer potential u with density σ given by (3) satisfies the interior Dirichlet BVP with data f. A proof of the existence of a solution σ of (4) for given boundary data f then yields a constructive proof of the existence of a solution to the corresponding interior Laplace Dirichlet problem. The existence proof for the integral equation is obtained by noticing that the kernel of the integral operator is compact, so that the integral equation is of Fredholm's second kind, and using the Fredholm alternative.

In the end, we find that in order to solve the BVP (2), it suffices to solve the integral equation (4); the solution $u(x)$ may then be evaluated at any point $x \in \Omega$ using the expression (3). This approach offers several advantages over the direct solution of the BVP.

- The system unknown is now defined on the domain boundary Γ, rather than in Ω, so we have reduced the dimensionality of the space that must be discretized. This also leads to the availability of methods which deal efficiently with highly complex geometries.

- Second-kind Fredholm integral equations have well-conditioned discretizations; the operator in (4) is of the form $-I/2 + K$, where K is compact, which can be interpreted as a small perturbation of $-I/2$. Matrices discretizing this operator therefore tend to have eigenvalues clustered around $-1/2$, and the corresponding linear system may be solved efficiently using iterative methods.

- There are many tools available for discretization of various types of integrals to arbitrarily high order by quadrature, so that discretization methods are available which readily achieve accuracies near machine precision for this and related problems.

As indicated in the final point, the integral equation (4) may be discretized using quadrature. In the Nyström method, we define a set of quadrature nodes $\mathbf{x}_1, \ldots, \mathbf{x}_N$ on Γ, and a set of corresponding quadrature weights $w_1, \ldots, w_N$, designed to perform the integral in (4), for $x \in \Gamma$, with high-order accuracy. It turns out that for this particular integral equation, the kernel is smooth, so that trapezoidal rule quadrature is sufficient, but for other simple integral equations, more sophisticated quadrature rules for singular integrals are required. The integral equation is therefore replaced by a set of linear equations in the unknown σ_i,

$$-\frac{\sigma_i}{2} + \sum_{j=1}^{N} \frac{\partial \Phi(\mathbf{x}_i, \mathbf{x}_j)}{\partial n_{\mathbf{x}_j}} \sigma_j w_j = f_i$$

for $i = 1, \ldots, N$, where $f_i \equiv f(\mathbf{x}_i)$.

This constitutes a well-conditioned $n \times n$ linear system. Direct solution would carry a cost of $\mathcal{O}(N^3)$, but this cost may be reduced to $\mathcal{O}(N^2)$ by using an iterative method such as GMRES; a modest constant number of iterations are required, and on each iteration, we must apply the system matrix at a cost of $\mathcal{O}(N^2)$. The main step of applying the system matrix is computing the sum over j for each $i = 1, \ldots, N$. This can be interpreted as computing the interaction between a set of N particles, with associated charges $\sigma_j w_j$, via the kernel $\frac{\partial \Phi(\mathbf{x}, \mathbf{x}')}{\partial n_{\mathbf{x}'}}$, which is closely related to the Coulomb kernel. As discussed in the introduction, this is precisely the type of computation which the FMM is designed to accelerate; the sum may be compared with (1). Using the FMM, then, the cost of applying the system matrix is reduced to $\mathcal{O}(N)$, and we obtain an $\mathcal{O}(N)$ method to solve the integral equation with high-order accuracy.

Here, we have discussed one of the simplest problems which may be solved with integral equation methods. Other problems involve different types of boundary conditions; generalizations to 3D; integral equation theories corresponding to more complicated PDEs, such as the Helmholtz equation, Maxwell's equations, Stokes equation, and the heat equation; and inhomogeneous and variable coefficient PDEs. Tackling these problems requires a great deal of mathematical analysis, in addition to the study of quadrature techniques for singular and near-singular integrals, of methods for rapidly evaluating complicated integral kernels, and of discretizing complex geometries, just to name a few topics. However, this simple problem illustrates the importance of a fast algorithm to compute particle-particle interactions, and the need for such an algorithm in these integral equation methods is ubiquitious. The details of the algorithms vary a great deal from problem to problem, but the starting point is to understand the original FMM for the problem of Coulomb interactions.

3 The FMM Algorithm

We now return to the original task, which is to compute the sums (1). This section closely follows [1]. We first introduce complex notation, which simplifies the analysis slightly. Identifying each point $\mathbf{x} \in \mathbb{R}^2$, $\mathbf{x} = (x, y)$ with the complex number $z = x + iy$ and using that

$$\log |\mathbf{x} - \mathbf{x}'| = \operatorname{Re} \left(\log(z - z') \right) = \log |z - z'|,$$

we can consider the task of computing the complex sums

$$\sum_{j\neq i} q_i \log(z_i - z_j)$$

for $i = 1, \ldots, N$, and z_i the complex number corresponding to $\mathbf{x}_i$. The real part of this expression is equal, up to a scaling by $-\frac{1}{2\pi}$, to (1). We will refer to $q\log(z-z')$ as the potential at z due to a source charge of strength q located at z'.

The FMM algorithm requires some simple analytical tools which we now introduce.

3.1 Tools

In this subsection we compile a list of modules used in the algorithm, stating the theorems that they rely on and briefly sketching proofs. These tools involve forming and manipulating far-field and local representations of potentials, and the specific use of each of them will become clear once we describe the algorithm. More details are contained in [1].

3.1.1 Multipole expansions

We will first require an efficient far-field representation of the potential due to a collection of charges, which we sometimes call sources. Suppose we have a m charges, with strengths $q_1, \ldots, q_m$, located at positions $z_1, \ldots, z_m$ inside of a disk of radius r, $|z_i| < r$. The total potential due to these charges is equal to $\phi(z) = \sum_{i=1}^m q_i \log(z - z_i)$; however, in the region $|z| > r$, this total potential may written as a multipole expansion,

$$\phi(z) = Q\log(z) + \sum_{k=1}^{\infty} \frac{a_k}{z^k} \tag{5}$$

where $Q = \sum_{i=1}^m q_i$ is the total charge and the constants a_k, called multipole coefficients, are given explicitly in terms of the quantities q_i and z_i

$$a_k = -\frac{1}{k}\sum_{i=1}^m q_i z_i^k. \tag{6}$$

Furthermore, this multipole expansion can be truncated and the truncation error bounded in terms of known quantities. Suppose $|z| > R > r$, so that z is separated from the disk containing the charges by more than a known distance. Then, letting ϕ_p be the p-term truncation of the multipole expansion given above,

$$\phi_p(z) = Q\log(z) + \sum_{k=1}^{p} \frac{a_k}{z^k},$$

we have

$$|\phi(z) - \phi_p(z)| < \frac{A}{R/r - 1}\left(\frac{r}{R}\right)^p \tag{7}$$

where $A = \sum_{i=1}^m |q_i|$. For example, if $R = 2r$, then the truncation error is bounded by $A/2^p$. The error improves by increasing R as compared with r and by increasing p. Thus a desired error may be achieved, given R, by making an appropriate selection of p. We note that a p-term multipole expansion is formed in $\mathcal{O}(mp)$ by computing each of the p coefficients a_k using (6).

The proofs of these statements follow from the Taylor series expansion for $\log(1-w)$, $|w| < 1$, and simple estimates. Indeed, for $|z_i| < r < |z|$, we have

$$\log(z - z_i) = \log(z) + \log(1 - z_i/z) = \log(z) - \sum_{k=1}^{\infty} \frac{1}{k} \left(\frac{z_i}{z}\right)^k$$

Multiplying by q_i, summing over $i = 1, \ldots, m$, and switching the order of summation gives (5). The truncation error is obtained by elementary estimation of the magnitude of the tail of the series.

3.1.2 Multipole-to-multipole translations

We next need a method to shift the center of a multipole expansion. If we have the multipole expansion of a collection of charges in a disk of radius r centered at z_0, valid for z in $|z - z_0| > r$, then there is a multipole expansion of the same charges centered at the origin, valid for $|z| > |z_0| + r$. The situation is illustrated in Figure 1a. The origin is only chosen as the center for notational and computational convenience.

We can calculate the coefficients of the shifted expansion in terms of those of the original. If the expansion centered at z_0 is given by

$$\phi(z) = a_0 \log(z - z_0) + \sum_{k=1}^{\infty} \frac{a_k}{(z - z_0)^k}, \tag{8}$$

then in the region $|z| > |z_0| + r$, we have

$$\phi(z) = a_0 \log z + \sum_{l=1}^{\infty} \frac{b_l}{z_l}$$

with

$$b_l = \left(\sum_{k=1}^{l} a_k z_0^{l-k} \binom{l-1}{k-1} \right) - \frac{a_0 z_0^l}{l}.$$

This follows from expanding (8) in a Taylor series in z_0, or from algebraic manipulations.

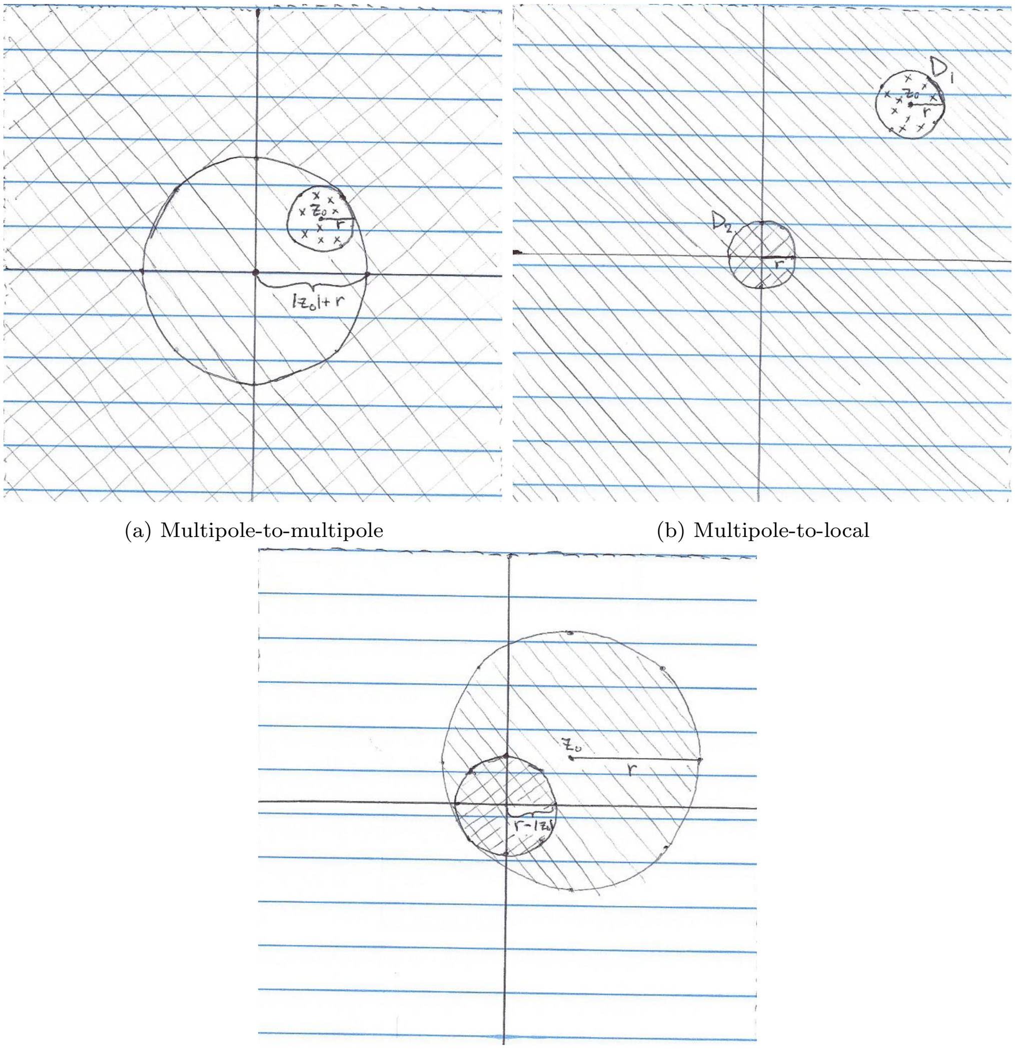
The new multipole expansion may also be truncated, and an error estimate analogous to (7) holds, with r replaced by $|z_0| + r$, and an outer radius R chosen so that $|z| > R > |z_0| + r$. Indeed, the multipole expansion centered at the origin is a valid multipole expansion for the collection of charges in the region $|z| > |z_0| + r$, so we can use the same reasoning to obtain the estimate.

We note that the coefficients b_k are linear combinations of the a_k. That is, the shifted expansion may be obtained by application of a $p \times p$ matrix to the truncated vector of multipole coefficients, at a cost of $\mathcal{O}(p^2)$ for a p-term expansion. We call this the multipole-to-multipole translation operator.

3.1.3 Multipole-to-local translations

We next require an efficient representation of the potential inside a disk due to a collection of source charges far from the disk. For this, a simple Taylor expansion of the potential inside the disk will suffice. However, if the charges far from the disk are contained in another disk and represented by a multipole expansion, there is an efficient way of obtaining the Taylor expansion coefficients from the multipole coefficients. We refer to this Taylor expansion as a local expansion.

Suppose, as in Figure 1b, that we have a multipole expansion (8) valid outside a disk D_1 of radius r centered at z_0, and we wish to obtain the coefficients of a local expansion representing the



(a) Multipole-to-multipole

(b) Multipole-to-local

(c) Local-to-local

Figure 1: Translation operators. Region of validity of first expansion indicated by striped lines; of second expansion by cross-hatched lines.

same potential inside a disk D_2 of radius r centered at the origin, with $D_1 \cap D_2 = \varnothing$. Then inside D_2, we have the local expansion

$$\phi(z) = \sum_{l=0}^{\infty} b_l z^l$$

with

$$b_0 = \sum_{k=1}^{\infty} \frac{a_k}{z_0^k}(-1)^k + a_0 \log(-z_0)$$

and

$$b_l = \left(\frac{1}{z_0^l} \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} \binom{l+k-1}{k-1} (-1)^k \right) - \frac{a_0}{l \cdot z_0^l}$$

when $l \geq 1$. This expression may be obtained by developing the multipole expansion in a Taylor series about the origin.

The local expansion may be truncated, and there is again an error estimate similar to (7),

$$|\phi(z) - \phi_p(z)| < \frac{A(4e(p+c)(c+1)+c^2)}{c(c-1)}\left(\frac{1}{c}\right)^{p+1}.$$

Here ϕ_p is the p-term truncated local expansion, $c > 1$ is such that $|z_0| - R > cr$ - it is the distance between D_1 and the origin in units of r - and A is as before. The salient features of this estimate are analogous to those of the previous; given an a priori-known separation distance between D_1 and D_2, it is straightforward to choose p to achieve a given truncation error.

We observe again that there is a $p \times p$ matrix taking the coefficients of a p-term truncated multipole expansion in D_1 to those of a p-term truncated local expansion in D_2; we call this the multipole-to-local translation operator.

3.1.4 Local-to-local translations

Suppose, as in Figure 1c, that we have a truncated local expansion in a disk of radius r centered at z_0, and we wish to shift its center to the origin. The new expansion will be valid inside a disk of radius $r - |z_0|$ centered at the origin, and is given by the following expression

$$\sum_{k=0}^{n} a_k (z - z_0)^k = \sum_{l=0}^{n} b_l z^l$$

with

$$b_l = \sum_{k=l}^{n} a_k \binom{k}{l} (-z_0)^{k-l}.$$

This may be obtained by binomial expansion. The $p \times p$ matrix taking the a_k coefficients to the b_l coefficients is called a local-to-local translation operator.

3.2 The algorithm

The tools above allow us to efficiently represent the potential at target points outside a disk due to a collection of source charges inside this disk, as well as the potential at target points inside a disk of a collection of source charges outside this disk. They also allow us to manipulate these representations by converting from one to the other, or by shifting their centers. The expansions come equipped with simple a priori truncation error estimates which depend only on the distance between sources and targets.

We can now describe the algorithm itself and describe how these tools work together. We repeat that we are given a list of N particle locations $z_1, \ldots, z_N$, and charge strengths $q_1, \ldots, q_N$, and we must compute the potential at each particle location due to all other particles,

$$\sum_{j \neq i} q_i \log(z_i - z_j)$$

for each $i = 1, \ldots, N$. We also assume we are given a tolerance ε, and we must guarantee that the sums are computed with an error of at most ε.

The algorithm proceeds in several steps. First, in a setup phase, we will group nearby particles together using a tree structure which divides up the computational domain into boxes at many levels of resolution. Next, in the upward pass phase, we will construct multipole expansions representing the far-field potential of each group of particles at every level of resolution. Then, in the downward pass phase, we will construct local expansions representing the potentials due to particles far away from each group of particles at every level of resolution. Finally, we will use these local expansions, in addition to directly-computed near-neighbor potentials, to evaluate the potential at each particle due to all other particles.

3.2.1 Setup phase and construction of tree structure

The first step of the algorithm is to divide particles into groups at many levels of resolution, such that each particle falls within one group at each level. To do this, we use a quadtree. We define a computational box, which is a square containing all of the particles, called the level 0 box. The level 1 boxes are obtained by subdividing the level 0 box into four child boxes, and in general, level $l+1$ boxes are obtained by dividing each of the level l boxes into four child boxes. We subdivide boxes until there are at most s particles per box, where s is a small fixed number which could be as small as 1.

The total number of levels, L, is on the order of $\log_4 N$, and the total number of boxes at level l is 4^l. We call a box occupied if it contains a particle. The total number of occupied boxes is at most $\sum_{l=1}^{L} 4^l = \frac{1}{3}(4^{L+1}-1)$, which is about $\frac{4}{3}N$.

Two boxes in a given level are said to be well-separated if they are separated by at least one box at that level. Two boxes are said to be neighbors if they share an edge; a box is its own neighbor. Box B is said to be in the interaction list of box A if it is well-separated from A, but is not the child of a box which is well-separated from the parent of A. An example of an interaction list is shown in Figure 2; here, the box labeled T (target), at level 3 of the tree, has an interaction list of 27 boxes labeled S (source).

In addition to building this data structure, we must select a truncation parameter p for all multipole and local expansions which is sufficient to achieve the specified error. It will be possible to choose p as a simple function of ε, independent of N, and we will show how to do this after presenting the algorithm.

3.2.2 Upward pass

The objective of this step is to obtain a multipole expansion for every occupied box in the tree representing the far-field potential of all particles inside that box, valid outside the box.

To do this, we could simply loop through every occupied box at every level of the tree and form the multipole expansion of the particles contained in that box. Since the cost of forming a multipole expansion for a box containing m particles is $\mathcal{O}(mp)$, and boxes at the coarsest levels contain on the order of N particles, this would be somewhat expensive.

Instead, we start by forming the multipole expansion corresponding to each box at the finest level, level L. Since each of N particles is placed in one box at the finest level, the total cost of forming all of these expansions is approximately Np.

We then form expansions for boxes in level $L-1$ using multipole-to-multipole translation operators. Fix a box B at level L, let the origin be defined by the center of the parent C of box B in level $L-1$, and let the center of B with respect to that origin be z_0. Then we have a multipole expansion for B centered at z_0 and valid outside the disk circumscribing B, which can be shifted to a multipole expansion for C centered at the origin and valid outside the disk circumscribing C using a multipole-to-multipole translation. If we do this for each of the children of C, and add the resulting multipole coefficients, we obtain a multipole expansion for C representing the potential due to all of the particles contained in C, since each such particle is represented by the multipole expansion of one of the children of C.

An example of this is shown in Figure 2. Here, the box T shifts its multipole expansion, valid outside D_1, to a multipole expansion valid outside D_2. Each of the other child boxes of the parent box of T do the same, and the multipole coefficients are added together to obtain the multipole expansion for the parent.

We can repeat this for all boxes in level L to obtain multipole expansions for every box in level $L-1$, and repeat the same procedure moving up the tree, using the multipole expansions for boxes in level $l+1$ to obtain those for boxes in level l. In the end, we obtain the desired multipole expansions for every occupied box in the tree. This procedure is called the upward pass. Each multipole-to-multipole translation costs approximately p^2 operations, and we require one such translation for each box in levels 1 through L, so the total cost of these translations is approximately $\frac{4}{3}p^2N$. The total cost of the upward pass, including the cost of forming the expansions at the finest level, is therefore approximately $pN + \frac{4}{3}p^2N$.

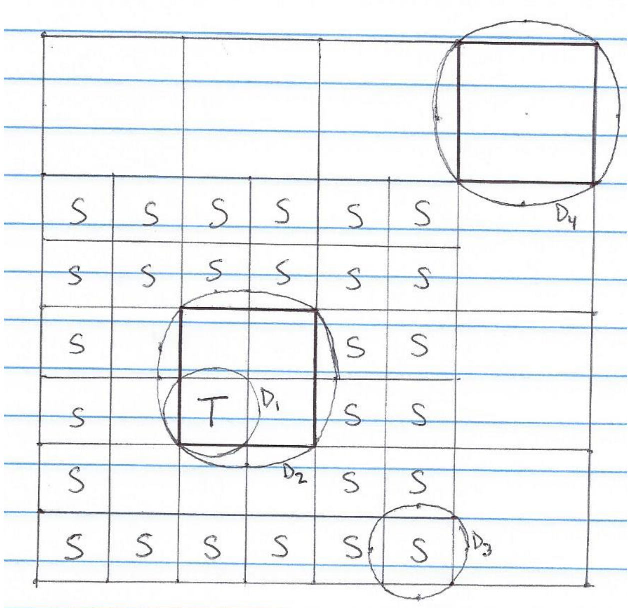


Figure 2: Upward pass: multipole expansion of box T, valid outside D_1, are shifted to multipole expansion of its parent box, valid outside D_2. Downward pass: Boxes labeled S (source) are in the interaction list of box T (target). Their multipole expansions - for example, the one valid outside of D_3 - are shifted to local expansions valid inside D_1. Also, the parent of box T shifts its local expansion, valid inside D_2, to a local expansion valid inside D_1. The local expansion inside D_2 already represents the potential due to boxes outside the interaction list of T; for example, the multipole expansion valid outside D_4 was shifted to a local expansion valid inside D_2 in the previous step of the downward pass.

3.2.3 Downward pass

The objective of this step is to obtain a local expansion for every occupied box in the tree representing the potential due to particles in all well-separated boxes, valid inside the box.

To do this, we proceed from the coarsest to the finest level. Levels 0 and 1 contain no well-separated boxes, so we begin at level 2, and fix a box B. At this level, the interaction list of B is simply the set of all boxes which are well-separated from it. For each such box C, we perform a multipole-to-local translation from the center of C to that of B, in order to obtain a local expansion in the disk circumscribing B of the potential due to all of the particles in C. We can then sum all of these local expansions over B to obtain a local expansion on B of the potential due to all of the particles in boxes well-separated from B at level 2. This is done for all boxes in level 2.

We next move to level 3, and fix a box B at that level. We wish to form a local expansion on B which represents the potential due to all particles in boxes well-separated from B. The boxes which are well-separated from B are either outside or inside of its interaction list. By the definition of the interaction list, the potential due to particles in any box outside of the interaction list of B has already been represented in the local expansion of the parent of B, so we can incorporate this potential into the local expansion of B by performing a local-to-local translation from the local expansion on the parent of B to B. We can then incorporate the potential due to particles in boxes in the interaction list of B using multipole-to-local translations, in the same manner as we did for the boxes in level 2. Adding together all of these local expansions, we obtain a local expansion on B which represents the potential due to particles in all boxes well-separated from B at level 3.

We can do this for all boxes in level 3. This process, for one box in level 3, is represented in Figure 2 and explained in the caption. We proceed down to the finest level in this manner, forming local expansions for every box B using local-to-local translations to obtain local expansions due to the particles in well-separated boxes which have already been incorporated by the parent box, and using multipole-to-local translations to obtain local expansions due to all particles in the interaction list of B. In the end we have a local expansion in each box which represents the field due to particles in all well-separated boxes.

Let us obtain a cost estimate for this step. We can first add up the cost of all local-to-local translations. Each costs approximately p^2 operations, and since each box requires one local-to-local translation to obtain the expansion from its parent, we have a total cost of approximately $\frac{4}{3}p^2N$. Each multipole-to-local translation also costs approximately p^2 operations, and we require one for each element of the interaction list of each box. The interaction list of a box contains at most 27 boxes, so the total cost of this step is approximately $27 \cdot \frac{4}{3}p^2N$. The downward pass therefore has a total cost of approximately $28 \cdot \frac{4}{3}p^2N$ operations.

3.2.4 Near neighbor interactions and evaluation of local expansions

We can now loop through each particle and obtain the potential due to all other particles, which are the desired sums. For a given particle, we can first evaluate the local expansion of the box containing that particle at the finest level at the particle position. This local expansion represents the potential due to all particles in well-separated boxes at the finest level; that is, all particles except those contained in neighbor boxes at the finest level. We can evaluate the potential due to particles in neighbor boxes at the given particle position directly.

Evaluating a single local expansion costs approximately p operations. There are 9 neighbor boxes, and at most s particles in each, so the direct evaluation of the potential due to nearby particles has a cost of approximately $9s$, or $\frac{9}{2}s$ if pairwise symmetries are taken into account. Since there are N particles, the total cost of this step is approximately $(p + \frac{9}{2}s)N$.

In the end, we have computed the desired sums at a cost which scales linearly with the number N of particles, rather than quadratically, as in the brute force algorithm. The accuracy of this computation is determined by the truncation parameter p, which we now discuss.

3.2.5 Choosing the truncation parameter p

The multipole expansion for a box is valid outside the disk circumscribing the box and is only evaluated in well-separated boxes. Local expansions for a box are similarly valid inside the disk circumscribing the box, and only represent potentials due to particles in well-separated boxes. One can calculate the ratio $\frac{r}{R}$ between the radius of any of these disks and the distance from its center to the edge of a well-separated box; it is $\frac{\sqrt{2}}{3} \approx .47$, which we round up to $\frac{1}{2}$. To obtain multipole expansions accurate to an error ε, then, (7) implies that it suffices to choose $A\left(\frac{1}{2}\right)^p < \varepsilon$, or $p > -\log_2 \frac{\varepsilon}{A}$. A truncation parameter for local expansions may be similarly obtained, and we can simply take p to be the larger of the two. Since the algorithm simply consists of forming and translating expansions which are correct to the truncation accuracy, the computed sums, obtained by direct evaluations of the Coulomb kernel and from local expansions at the finest level, will also by correct to the truncation accuracy. We have seen that the cost of the algorithm scales with p as p^2, so that it scales with ε as $(\log_2 \varepsilon)^2$.

4 Conclusion

We have introduced the FMM for Coulomb interactions in 2D, and given some motivation for its use in solving PDEs numerically using integral equation methods by giving an example in which the discretization of an integral equation reduces to a particle-particle interaction problem. We described an FMM algorithm which scales as $\mathcal{O}(N)$ with the number of particles and as $(\log_2 \varepsilon)^2$ with the desired accuracy ε. This algorithm succeeds in reducing the asymptotic cost of computing the sums (1).

In practice, the preconstant associated with this cost is rather large for the algorithm as stated, and becomes unwieldy for the three-dimensional version of the algorithm. There are improvements of the algorithm which reduce this constant dramatically by introducing more efficient implementations of translation operators, leading to much more practical versions of the FMM algorithm.

As we have indicated, particle-particle interaction problems arise in many areas of computational science, and the set of ideas provided by the FMM algorithm are therefore widely applicable in designing fast and robust numerical methods which approach problems from this viewpoint. Each new physical problem requires a new set of analytical tools which must be built into an FMM designed for that problem, but these tools, as in the case of the Coulomb kernel, tend to come from the standard toolbox of classical mathematical physics. Indeed, the FMM can be viewed as providing a set of ideas by which numerical methods making explicit use of the kernels arising in physics, and in other subjects, may be implemented efficiently. As these types of methods have begun to present an alternative to those emphasizing a PDE viewpoint, FMMs have become a standard tool in computational science.

References

[1] Leslie Greengard and Vladimir Rokhlin. A fast algorithm for particle simulations. *Journal of computational physics*, 73(2):325–348, 1987.

[2] O.D. Kellogg. *Foundations of Potential Theory.* Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen. Dover Publications, 1953.

[3] R.B. Guenther and J.W. Lee. *Partial Differential Equations of Mathematical Physics and Integral Equations.* Dover books on mathematics. Dover Publications, 1996.

[4] R. Kress. *Linear Integral Equations.* Applied Mathematical Sciences. Springer New York, 2013.