Accelerating the Fast Multipole Method Using The Fast Fourier Transform

Christopher Green

A final project for Math 596

June 7, 2001

AKB Systems Technical Report #01-06052001

Abstract

The Fast Multipole Method is a rapid means of evaluating potentials and forces due to Coulombic (electrostatic) and Newtonian (gravitational) interactions in many-particle systems. In this paper, we describe how the three-dimensional version of this method can be accelerated using a Fast Fourier Transform. The resulting algorithm has an expected running time of $O((p^2 \log p)\, N)$ (where p is a measure of the desired accuracy of the computation and N is the number of particles), a significant improvement over the $O(p^4 N)$ running time of the original algorithm.

1 Introduction

The Fast Multipole Method (FMM) of Greengard and Rokhlin [1, 5] provides a rapid means of computing potentials and forces due to Coulombic (electrostatic) and Newtonian (gravitational) interactions in N-body systems. Direct computation of all interactions involved requires $O(N^2)$ work, which can be prohibitively expensive when N is large. The FMM provides an $O(N)$ algorithm for performing these computations. The essence of the algorithm is to replace interactions between individual particles with interactions between clusters of particles. The aggregate potential due to a cluster at points outside the cluster is represented by a multipole expansion; the computation of interactions between clusters then reduces to the manipulation of multipole expansions. These expansions can be then be translated to local Taylor expansions inside each cluster to determine the desired potentials.

In practice, one only keeps a finite number p of terms in the multipole expansion; the exact number of terms necessary to guarantee a given level of precision can be computed using error bounds derived by Greengard [1]. When manipulating multipole and local expansions in $\mathbb{R}^3$, each of the p^2 coefficients in the output expansion involves the evaluation of p^2 coefficients of the input expansion, leading to $O(p^4)$ total work. Thus, the manipulation of a multipole or a local expansion can be rather expensive if it must be performed many times.

In the original FMM, one must convert N multipole expansions to local expansions, leading to an $O(p^4 N)$ complexity bottleneck. Elliot and Brown [6] describe a more efficient manner of performing this conversion that involves translating the problem into an equivalent problem of convolving two discrete sequences. This problem has a known solution, the two-dimensional Fast Fourier Transform (FFT) (see [4]), which can be performed in $O(p^2 \log p)$ operations.

In the sequel, we shall only discuss the conversion of the original FMM to the Fourier domain. All results stated herein apply equally to the Adaptive FMM discussed in [2].

2 Potentials in Three Dimensions

Potential fields in $\mathbb{R}^3$ correspond to solutions of the Laplace equation $\Delta\Phi = 0$. Such fields describe the Newtonian gravitational potential, the Coulombic electrostatic potential, and similar “at-a-distance” interactions. For simplicity, we shall restrict ourselves to the electrostatic viewpoint.

2.1 Expansion in Terms of Radial Distances

Suppose we place a point charge of unit strength at the origin; then the potential Φ and the electrostatic field E at any point $P \neq (0,0,0)$ in $\mathbb{R}^3$ are given by

$$\begin{aligned} \Phi &= \frac{1}{|P|} && (1) \\ E &= -\nabla\Phi = \frac{P}{|P|^3}, \end{aligned}$$

where $|P|$ is the Euclidean distance from the origin to P.

If we now move the source to the point Q, Coulomb’s Law tells us that the potential at P will be inversely proportional to $|P - Q| = r'$. Let us define $r = |P|$; we seek to express the potential at P in terms of r. This can be done more easily in spherical coordinates, so let

$$\begin{aligned} P &= (r, \theta, \phi) \\ Q &= (\rho, \alpha, \beta) \end{aligned}$$

and let γ be the angle between the two vectors $\vec{P}$ and $\vec{Q}$. Then by the Law of Cosines we have

$$(r')^2 = r^2 + \rho^2 - 2r\rho\cos\gamma. \tag{2}$$

We can express γ in terms of θ, ϕ, α and β by using the definition of the dot product

$$\vec{P} \cdot \vec{Q} = |\vec{P}||\vec{Q}|\cos\gamma$$

and the formulae for converting rectangular coordinates to spherical coordinates:

$$\begin{aligned} P &: (x, y, z) \to (r\cos\theta\sin\phi, r\sin\theta\sin\phi, r\cos\phi) \\ Q &: (x', y', z') \to (\rho\cos\alpha\sin\beta, \rho\sin\alpha\sin\beta, \rho\sin\beta). \end{aligned}$$

Thus

$$\cos\gamma = \cos\phi\cos\beta + \sin\phi\sin\beta\cos(\theta - \alpha)\,. \tag{3}$$

The potential can now be expressed as a power series in r:

$$\begin{aligned} \frac{1}{r'} &= \frac{1}{\sqrt{r^2 + \rho^2 - 2r\rho\cos\gamma}} \\ &= \frac{1}{r\sqrt{1 - 2\frac{\rho}{r}\cos\gamma + \left(\frac{\rho}{r}\right)^2}} \end{aligned}$$

Setting $\mu = \frac{\rho}{r}$ and $u = \cos\gamma$, we have

$$\frac{1}{r'} = \frac{1}{r\sqrt{1 - 2u\mu + \mu^2}}$$

If $0 < \mu < 1$, we may expand the right-hand side in powers of μ to obtain

$$\frac{1}{r\sqrt{1 - 2u\mu + \mu^2}} = \frac{1}{r}\sum_{n=0}^{\infty} P_n(u)\mu^n,$$

where $P_n(u)$ is the n-th Legendre polynomial. (In fact, the left-hand side (without the factor of $1/r$) is the generating function for the Legendre polynomials.) Thus we have the **multipole expansion**

$$\frac{1}{r'} = \sum_{n=0}^{\infty} P_n(u)\frac{\rho^n}{r^{n+1}}. \tag{4}$$

The multipole expansion describes the far field due to the charge at Q, that is, the effect of the charge at Q on particles "far away". Notice $\mu < 1$ iff $r > \rho$, so the multipole expansion is only valid outside a sphere of radius $|Q|$.

Newton's third law, however, states that the force on P due to Q is equal to and opposite the force on Q due to P. The potentials necessarily satisfy the same relation. Therefore we may exchange the roles of r and ρ in (4) to obtain the **local expansion**

$$\frac{1}{r'} = \sum_{n=0}^{\infty} P_n(u) \frac{r^n}{\rho^{n+1}}, \tag{5}$$

which is valid inside the sphere of radius $|Q|$. The local expansion describes the "direct" field due to the charge at Q.

In the 2-D case (see [1]), we were able to obtain the potential at P due to multiple source charges Q_i by superposition. However, superposition does not work here, as the series we have developed depends upon the relative position of the particles: each charge will contribute a different angle γ_i. The Legendre polynomials, however, do not satisfy a "nice" addition formula, in the sense that there do not exist coefficients $a_{j,k}$ such that

$$P_n(\cos\gamma_1) + P_n(\cos\gamma_2) = \sum_{j,k} a_{j,k} P_j(\cos\gamma_1) P_k(\cos\gamma_2). \tag{6}$$

It is clear that we will not be able to use the series we have developed to describe the potential due to a general collection of particles. We must modify our approach to the problem.

2.2 Expansion in Terms of Spherical Harmonics

Since we will be working in spherical coordinates, it is natural to consider the Laplacian in spherical coordinates:

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{\partial}{\partial r}\Phi\right) + \frac{1}{r^2 \sin\theta}\frac{\partial}{\partial \theta}\left(\sin\theta \frac{\partial}{\partial \Phi}\right) + \frac{1}{r^2 \sin^2\theta}\frac{\partial^2}{\partial \theta^2} = 0. \tag{7}$$

Standard separation of variables techniques yield the solution to (7) in terms of **spherical harmonics**:

$$\Phi = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \left[\mathbf{L}_n^m r^n + \frac{\mathbf{M}_n^m}{r^{n+1}}\right] Y_n^m(\theta, \phi), \tag{8}$$

where

$Y_n^m(\theta,\phi)\, r^n$ is the spherical harmonic of degree n;

$\frac{Y_n^m(\theta,\phi)}{r^{n+1}}$ is the spherical harmonic of degree $-n-1$;

$\mathbf{L}_n^m$ is the **(local) moment** of the expansion;

$\mathbf{M}_n^m$ is the **(multipole) moment** of the expansion.

In the far field expansion, the potential must decay to zero at infinity (since $1/r \to 0$ as $r \to \infty$), so all the local moments $\mathbf{L}_n^m$ are zero in the multipole expansion for Φ. Likewise, the local expansion must be analytic in a sphere containing all the charges, so the multipole moments $\mathbf{M}_n^m$ are all zero.

By considering derivatives of the fundamental solution $1/r$ with respect to x, y, and z, we may derive the formulas[1] [1, 4]

$$Y_n^m(\theta,\phi) = \sqrt{\frac{(n-m)!}{(n+m)!}} P_n^m(\cos\theta) e^{im\phi}, \quad \text{for } m \geq 0, \tag{9}$$

$$Y_n^{-m}(\theta,\phi) = (-1)^m \overline{Y_n^m(\theta,\phi)} \tag{10}$$

where the P_n^m's are the **associated Legendre functions**, which can be defined in terms of the Legendre polynomials by the equation

$$P_n^m(x) = (-1)^m (1-x^2)^{m/2} \frac{d^m}{dx^m} P_n(x). \tag{11}$$

The associated Legendre functions are usually computed recursively using the recurrences

$$(2n+1)xP_n^m(x) = (n-m+1)P_{n+1}^m(x) + (n+m)P_{n-1}^m(x)$$
$$P_n^{m+2}(x) + 2(m+1)\frac{x}{\sqrt{1-x^2}} P_n^{m+1}(x) = (n-m)(n+m+1)P_n^m(x).$$

The advantage of expressing the potentials in terms of spherical harmonics is that the Legendre polynomials satisfy an addition theorem that involves spherical harmonics [1]:

Theorem 2.2.1 (Addition Theorem for Legendre Polynomials). *If $P = (r, \theta, \phi)$ and $Q = (\rho, \alpha, \beta)$ are points expressed in spherical coordinates, then*

$$P_n(\cos\gamma) = \sum_{m=-n}^{n} Y_n^{-m}(\alpha,\beta)\, Y_n^m(\theta,\phi), \tag{12}$$

where $\cos\gamma$ is the cosine of the angle between $\vec{P}$ and $\vec{Q}$, given by (3).

We can now derive the **Multipole Expansion** that will be used to develop the Fast Multipole Method.

Theorem 2.2.2 (Multipole Expansion). *Let $Q_i = (\rho_i, \alpha_i, \beta_i)$, $i = 1, \ldots, k$ be the locations of k point charges in $\mathbb{R}^3$. Let q_i be the strength of the charge at Q_i. Suppose there exists a real number $a > 0$ such that $|\rho_i| < a$ for all i. Then for any point $P = (r, \theta, \phi)$ with $r > a$, the potential $\Phi(P)$ at P is given by the multipole expansion*

$$\Phi(P) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{\mathbf{M}_n^m}{r^{n+1}} Y_n^m(\theta,\phi), \tag{13}$$

where the multipole moments are given by

$$\mathbf{M}_n^m = \sum_{i=1}^{k} q_i \rho_i^n Y_n^{-m}(\alpha_i,\beta_i). \tag{14}$$

[1] The standard definition of the spherical harmonic $Y_n^m(\theta,\phi)$ includes a factor of $\sqrt{(2n+1)/4\pi}$, which we have omitted for convenience.

The proof of Theorem 2.2.2 follows directly from formula (4) for the multipole expansion of a single particle and Theorem 2.2.1 [1].

In order to make use of Theorem 2.2.2, we must define the notion of well-separated sets.

Definition 2.2.1. *Let $\mathbf{Q} = \{Q_i\}$ and $\mathbf{P} = \{P_i\}$ be two sets of points in $\mathbb{R}^3$. We say that $\mathbf{P}$ and $\mathbf{Q}$ are* ***well-separated*** *if there exist elements $Q_0 \in \mathbf{Q}$, $P_0 \in \mathbf{P}$ and a real number $r > 0$ such that*

1. *every element of $\mathbf{Q}$ is contained in the ball of radius r about Q_0;*
2. *every element of $\mathbf{P}$ is contained in the ball of radius r about P_0; and*
3. *$|P_0 - Q_0| > 3r$.*

Informally, two sets of points A and B are well-separated if there is some $r > 0$ such that A is contained in a ball B_A of radius r, B is contained in a ball B_B of radius r and distance between the two sets B_A and B_B is at least r.

The multipole expansion is most appropriate when we wish to determine the potential due to a collection of point charges $\{Q_i\}, i = 1, \ldots, k$ at points $P_i, i = 1, \ldots, n$ that are well-separated from the sources. If we choose to keep p terms of (13), then the expansion will have p^2 coefficients. Contrast evaluating k fields at n points, which takes $O(nk)$ operations, with computing the multipole expansion once ($O(kp^2)$ operations) and evaluating it n times ($O(np^2)$ work). Once p is fixed the total work needed to evaluate the potentials at the P_i's using the multipole expansion is only $O(n) + O(k)$, which is asymptotically less than $O(nk)$.

In addition to the multipole expansion, we will need formulae for translating the centers of multipole expansions, converting multipole expansions to local expansions, and translating the centers of local expansions. Greengard [1] proves several addition formulae for spherical harmonics which are needed to prove the following theorems; however, the derivation of these formulae is rather technical and not critical to our discussion, so we shall state the following theorems without proof.

Theorem 2.2.3 (Translation of a Multipole Expansion). *Suppose that l charges of strengths $q_1, \ldots, q_l$ are located inside the sphere $B_a(Q_0)$ of radius a centered at $Q_0 = (\rho, \alpha, \beta)$, and that the potential due to the charges at any point P outside $B_a(Q_0)$ is given by the multipole expansion*

$$\Phi(P) = \sum_{n'=0}^{\infty} \sum_{m'=-n'}^{n'} \frac{\mathbf{M}_{n'}^{m'}}{r'^{n'+1}} Y_{n'}^{m'}(\theta', \phi'), \tag{15}$$

where $P - Q = (r', \theta', \phi')$. Then for any point $P = (r, \theta, \phi)$ outside $B_{a+\rho}(Q_0)$,

$$\Phi(P) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{\mathbf{O}_n^m}{r^{n+1}} Y_n^m(\theta, \phi), \tag{16}$$

where

$$\mathbf{O}_n^m = \sum_{n'=0}^{n} \sum_{m'=-n'}^{n'} \frac{\mathbf{M}_{n-n'}^{m-m'} J_{m'}^{m-m'} A_{n'}^{m'} A_{n-n'}^{m-m'} \rho^{n'} Y_{n'}^{-m'}(\alpha,\beta)}{A_n^m} \tag{17}$$

$$A_n^m = \frac{(-1)^n}{\sqrt{(n-m)!(n+m)!}} \tag{18}$$

$$J_m^{m'} = \begin{cases} (-1)^{min(|m|,|m'|)}, & \textit{if } mm' < 0; \\ 1, & \textit{otherwise} \end{cases} \tag{19}$$

Theorem 2.2.4 (Multipole Expansion to Local Expansion Conversion). *Suppose that l charges of strengths $q_1, \ldots, q_l$ are located inside the sphere $B_a(Q_0)$ of radius a centered at $Q_0 = (\rho, \alpha, \beta)$, and that $\rho > 2a$. Then the corresponding multipole expansion* (15) *converges inside the sphere $B_a(0)$ of radius a centered at the origin. Inside $B_a(0)$, the potential due to the charges $q_1, \ldots, q_n$ is given by the local expansion*

$$\Phi(P) = \sum_{n'=0}^{\infty} \sum_{m'=-n'}^{n'} \mathbf{L}_{n'}^{m'} Y_{n'}^{m'}(\theta,\phi)\, r^{n'}, \tag{20}$$

where

$$\mathbf{L}_{n'}^{m'} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{\mathbf{M}_n^m J_m^{m'} A_n^m A_{n'}^{m'} Y_{n'+n}^{m-m'}(\alpha,\beta)}{A_{n'+n}^{m'-m} \rho^{n'+n+1}} \tag{21}$$

$$J_m^{m'} = (-1)^{n'} \begin{cases} (-1)^{min(|m'|,|m|)}, & \textit{if } mm' > 0; \\ 1, & \textit{otherwise} \end{cases} \tag{22}$$

and A_n^m is defined as in (18).

Theorem 2.2.5 (Translation of a Local Expansion). *Let $Q = (\rho, \alpha, \beta)$ be the center of a local expansion*

$$\Phi(P) = \sum_{n'=0}^{p} \sum_{m'=-n'}^{n'} \mathbf{O}_{n'}^{m'} Y_{n'}^{m'}(\theta',\phi')\, r'^{n'}, \tag{23}$$

where $P = (r, \theta, \phi)$ and $P - Q = (r', \theta', \phi')$. Then

$$\Phi(P) = \sum_{n=0}^{p} \sum_{m=-n}^{n} \mathbf{L}_n^m Y_n^m(\theta,\phi)\, r^n, \tag{24}$$

where

$$\mathbf{L}_n^m = \sum_{n'=n}^{p} \sum_{m'=-n'}^{n'} \frac{\mathbf{O}_{n'}^{m'} J_{n'-n,m'-m}^{m'} A_{n'-n}^{m'-m} A_n^m Y_{n'-n}^{m'-m}(\alpha,\beta)\,\rho^{n'-n}}{A_{n'}^{m'}} \tag{25}$$

$$J_{v,w}^{m'} = (-1)^v \begin{cases} (-1)^w, & \textit{if } wm' < 0; \\ (-1)^{m'-w}, & \textit{if } wm' > 0 \textit{ and } |m'| < |w|; \\ 1, & \textit{otherwise} \end{cases} \tag{26}$$

and A_n^m *is defined as in* (18).

Formula (24) allows us to translate the center of a truncated local expansion exactly (i.e., no error is introduced in the translation).

We now possess all the tools we need to develop the Fast Multipole Method.

3 The Fast Multipole Method

As we stated earlier, the Fast Multipole Method (FMM) is an $O(N)$ method of computing all N^2 interactions in the N-body problem. The method is described in detail in [1]; we shall only give a brief overview here.

The (main idea) of the FMM is to replace interactions between particles with interactions between clusters of particles. These clusters of particles are obtained by recursively partitioning the computational domain (a subset of $\mathbb{R}^3$) into cubes of identical size, creating a hierarchical **"oct-tree"** [6] of cells. For our purposes, we will assume that the distribution of particles is uniform and that there is some lower bound k_n on the number of particles per cell. This assumption is not required by the FMM, as there is an adaptive version of the FMM that handles nonuniform distributions of particles efficiently; however, the uniform version of the FMM is much easier to describe. The analysis we present applies equally well to the adaptive version of the FMM.

In order to describe the concept of "well-separated" algorithmically, we make the following definitions. In this and future definitions we will often abuse notation and refer to the i-th box of a given level as i. We shall also make no distinction between a charge and its physical location when our meaning is clear from the context.

Definition 3.0.2. *The* ***nearest neighbors*** *of the i-th box of level l are all the boxes of level l that share at least one boundary point with i.*

The ***second nearest neighbors*** *of i are all boxes of level l that are nearest neighbors of i's nearest neighbors.*

The ***interaction list*** *of i is the set of boxes that are children of i's parents nearest and second nearest neighbors but not nearest or second nearest neighbors of i. These are precisely the boxes well-separated from i.*

In our description of the algorithm we will also make use of the following notation:

ϵ will denote the desired precision of the computation;

n will denote the number of levels in our oct-tree;

$\Phi_{l,i}$ will denote the p term multipole expansion about the center of the i-th box of level l;

$\Psi_{l,i}$ will denote the p term local expansion about the center of the i-th box of level l;

$\tilde{\Psi}_{l,i}$ will denote the local potential due to particles outside i's parent's box and the nearest and second nearest neighbors of i's parent;

ilist will denote the interaction list of i.

To initialize the algorithm, we set $p = \lceil -\log_2 \epsilon \rceil$ and $n = \lceil \log_8 N \rceil$. We then proceed as follows:

1. Form multipole expansions in each of the boxes of the finest level ($l = n$).
2. For every box i of each of the coarser levels (that is, those with $l < n$), form a multipole expansion in i by shifting the centers of the multipole expansions of each of i's eight children to i's center using Theorem 2.2.3 and accumulating the resulting expansions. Proceed up the oct-tree in this fashion.
3. Starting at level 2, form local expansions for each of the boxes of levels 2 through $n-1$. Use Theorem 2.2.4 to convert the multipole expansion of each box in i's interaction list into a local expansion about i's center, and accumulate the results. After all the contributions from boxes well-separated from i have been accounted for, add the resulting expansion to $\tilde{\Psi}_{l,i}$ to obtain the local expansion for i. After local expansions have been formed for all boxes at a given level, form $\tilde{\Psi}_{l+1,i}$ for each box at the next level of refinement by translating $\Psi_{l,i}$ to each i's children's centers.
4. Form local expansions at each box of level n using the same procedure described in step 3.
5. Compute the well-separated interactions by evaluating the local expansion of box i of level n at each of the charges inside i.
6. Compute the direct interactions for particles in the current cell and its nearest neighbors using the direct formula for the potential (1).
7. Add the results of steps 5 and 6 together. This is the total potential at each charge location due to all other charges.

The algorithm will compute the potentials to within the accuracy determined by ϵ.

3.1 Complexity Analysis of the Fast Multipole Method

The complexity analysis of the algorithm is summarized in Table 1.

The overall expected running time is thus

$$O(p^2N) + O(p^4N) + O(N) = O(p^4N).$$

Step	Work	Explanation
1	$O(p^2N)$	N multipole expansions with p^2 coefficients each
2	$O(p^4N)$	8^l shifts per level, p^4 work per shift ($\sum_{l=0}^{n} 8^l = O(N)$)
3	$O(876p^4N)$	The interaction list contains at most 875 entries; each entry requires $O(p^4N)$ work. The computation of $\tilde{\Psi}$ requires an additional $O(p^4N)$ work.
4	$O(875p^4N)$	Again, there are at most 875 entries in the interaction list of a given box.
5	$O(p^2N)$	We compute one p-th degree expansion with p^2 coefficients for each particle.
6	$\frac{25}{2}Nk_n$	Recall that k_n is a lower bound for the number of particles in a box. We compute all interactions between a given particle and all other particles in the current box and its nearest neighbors. By Newton's Third Law we only have to compute half the interactions.
7	$O(N)$	One addition per particle.

Table 1: Complexity Analysis of the Fast Multipole Method.

4 Accelerating Multipole-to-Local Translation

As the analysis presented in Table 1 shows, the performance-limiting subroutines of the Fast Multipole Method are the cell-to-cell translations (multipole-to-local, multipole-to-multipole, local-to-local). For each box at a given level, these subroutines require us to perform an operation involving the expansions from multiple boxes. A closer examination of the algorithm reveals that the multipole-to-multipole and local-to-local conversions only occur between a parent and its eight children, while the multipole-to-local translation occurs involves not only a given box but also all the boxes of its interaction list. Since the interaction list of an arbitrary box can be quite long, it stands to reason that we should focus our acceleration efforts on the multipole-to-local conversion subroutine.

The multipole-to-local translation dominates the runtime of the algorithm due to the large number of cells contained in an oct-tree and the resulting long interaction lists. If we label the initial domain as level 0, each additional level l of spatial decomposition produces an additional 8^l cells. If we specify a well-separated criterion of one cell-width (as Greengard [1] does), the interaction lists are at most 875 entries long; for a two cell-widths criterion, the lists can contain as many as 189 entries [6]. (Recall Definition 3.0.2: the interaction list of box i contains those boxes that are children of the nearest and second-nearest neighbors of i's parent but not neighbors of i. The parent of i has at most 26 nearest neighbors and 99 second-nearest neighbors. Each such box has 8 children, for a total of 1000 eligible boxes. But i has 26 nearest neighbors and 99 second-nearest neighbors, so 125 of these boxes can be eliminated. Thus the interaction list of i contains at most $1000 - 125 = 875$ boxes.)

A p-term multipole expansion of the form (13) involves p^2 coefficients. In an actual implementation, one often places these coefficients in a $p \times p$ matrix. Under this formulation, the M2L conversion involves the elementwise multiplication of two $p \times p$ matrices for each of the p^2 terms of the output local expansion, for a total of $O(p^4)$ work. These matrix multiplications, however, resemble two-dimensional discrete convolution, for which more efficient algorithms are known (namely, the two-dimensional Fast Fourier Transform).

4.1 Conversion to Fourier Domain

Recall the formula for multipole-to-local conversion given in Theorem 2.2.4:

$$\Phi(r) = \Phi'(r') = \sum_{n=0}^{p-1} \sum_{m=-n}^{n} \mathbf{M}_n^m \frac{Y_n^m(\theta, \phi)}{r^{n+1}} = \sum_{n'=0}^{p-1} \sum_{m'=-n'}^{n'} \mathbf{L}_{n'}^{m'} Y_{n'}^{m'}(\theta', \phi')\, r'^{n'+1} \tag{27}$$

As we discussed earlier, this conversion takes an input matrix $\mathbf{M}_n^m$ of multipole coefficients and a weighting matrix $\mathbf{T}_{n,m,n',m'}$ of translation coefficients and produces an output matrix $\mathbf{L}_{n'}^{m'}$ of local coefficients:

$$\mathbf{L}_{n'}^{m'} = \sum_{n,m} \mathbf{T}_{n,m,n',m'} \mathbf{M}_n^m, \tag{28}$$

where

$$\mathbf{T}_{n,m,n',m'} = \frac{(-1)^{n'+m'} A_n^m A_{n'}^{m'}}{A_{n'+n}^{m'-m}} \frac{\overline{Y_{n'+n}^{m'-m}(\alpha, \beta)}}{\rho^{n'+n+1}} \tag{29}$$

$$A_n^m = \frac{(-1)^{n+m}}{\sqrt{(n+m)!(n-m)!}}. \tag{30}$$

All matrices are approximately half-filled, since the coefficients are zero outside of the allowed range of m and l ($0 \leq l \leq p-1$, $-l \leq m \leq l$). The spherical harmonics $Y_l^m(\theta, \phi)$ equal zero for index values outside this range.

The key insight here is that the double sum (28) resembles linear convolution of the two-dimensional matrix $\mathbf{M}_n^m$ with the "four-dimensional" matrix $\mathbf{T}_{n,m,n',m'}$, where the four dimensions of $\mathbf{T}_{n,m,n',m'}$ are indexed by m, n, m', and n'. It would be computationally easier, however, if our weighting matrix were only two-dimensional. To this end, we endeavour to express the entries of $\mathbf{T}_{n,m,n',m'}$ as functions of the differences $n'-n$ and $m'-m$; the resulting transfer matrix $\mathbf{H}_n^m$ will only be two-dimensional. We can accomplish this by the so-called "warping" operation of [6].

4.1.1 Warping

Let us move all terms involving only primed indices to the left side of (28) and all terms involving only unprimed indices to the right side. The resulting equation

$$\left[(-1)^{-n'} \frac{\mathbf{L}_{-n'}^{m'}}{A_{-n'}^{m'}}\right] = \sum_{n,m} \left[\mathbf{H}_{-(n'-n)}^{m'-m}\right] \left[(-1)^m \mathbf{M}_n^m A_n^m\right], \tag{31}$$

where

$$H_n^m = \frac{\overline{Y_n^m(\alpha, \beta)}}{A_n^m \rho^{n+1}}, \tag{32}$$

"warps" the multipole matrix $\mathbf{M}_n^m$ with the normalization functions A_n^m and produces a warped local matrix [6]. The local expansions needed to compute potentials may be recovered simply by reversing the warping operation.

The advantage of the warping operation should be evident. Equation (31) expresses precisely the two-dimensional linear convolution relation discussed earlier. We can make the resemblance more plain by defining the "warped" functions

$$y(n,m) = \left[(-1)^{-n'} \frac{\mathbf{L}_{-n'}^{m'}}{A_{-n'}^{m'}}\right] \tag{33}$$

$$x(n,m) = [(-1)^m \mathbf{M}_n^m A_n^m] \tag{34}$$

$$h(n,m) = [\mathbf{H}_n^m]\,. \tag{35}$$

In terms of these functions, (31) becomes

$$y(-n',m') = \sum_{n,m} h(-(n'-n), m'-m) x(n,m) = h(-n',m') \otimes x(n',m'), \tag{36}$$

where $\otimes$ denotes discrete convolution. Note that in this form, it is no longer necessary to distinguish between primed and unprimed indices.

It is a well-known fact, however, that discrete convolution can be performed more efficiently using the Discrete Fourier Transform. It makes sense, therefore, to convert (36) to the Fourier domain:

$$Y(\omega_{-n}, \omega_m) = H(\omega_{-n}, \omega_m) X(\omega_{-n}, \omega_m), \tag{37}$$

where we have used capital letters to denote discrete Fourier transforms.

In theory, the quantities in (37) would be computed using a Fast Fourier Transform (FFT): we take the FFT of h and x, multiply the resulting matrices elementwise to get Y, then perform an inverse FFT to recover y. The two-dimensional FFT can be done in $O(p^2 \log_2 p)$ operations, while the matrix multiplication takes $O(p^2)$ operations. Thus the total work performed is $O(p^2 \log p)$ operations, which is asymptotically better than the $O(p^4)$ operations needed to do the conversion directly.

In practice, the structure of h and x requires us to take extra measures when implementing this scheme in order to achieve the correct answer.

4.2 Implementation Issues

While our FFT-accelerated FMM rests upon solid theoretical ground, it is numerically unstable and inefficient as described. Several modifications must be made in order to obtain a stable, correct algorithm.

First, note that our convolution equation (36) involves $-n$, not n as in a standard convolution product. This implies that the transfer matrix and the output matrix will be aligned with the input matrix in the m direction, but reversed in the n direction. This fact must be taken into account when extracting the values of the output matrix.

Second, we must pad our matrices with zeros in order to obtain our linear convolution from the FFT's circular convolution. The FFT matrices will be $4p \times 2p$ matrices instead of $p \times p$ matrices, since $-(p-1) \leq m \leq p-1$ and $0 \leq n \leq p-1$ [6]. This increases the complexity of the computation, making it more expensive than the direct method for typical values of p ($p \leq 16$). This padding also introduces into the output matrix nonzero coefficients outside the allowed range of n and m, since the FFT computes over the entire $4p \times 2p$ space.

Fortunately, much of this padding can be eliminated. We can shrink our FFT matrices down to $2p \times 2p$ by noticing that the padding will introduce nonzero coefficients into elements outside the allowable range of m, so it is unnecessary to compute those entries. Further simplifications can be made by taking advantage of the relationships

$$\mathbf{M}_n^{-m} = (-1)^m \overline{\mathbf{M}_n^m} \quad \mathbf{L}_n^{-m} = (-1)^m \overline{\mathbf{L}_n^m} \tag{38}$$

and the identity (10) to eliminate redundancies in the one-dimensional FFT in the m direction. The upshot of these simplifications is that one half of the FFT matrix in the m direction contains all the information needed about the multipole or local expansion coefficients; the remaining entries are only needed implicitly and do not need to be stored. Thus we need only compute and store the entries of a $p \times 2p$ block.

We must still handle the superfluous entries in the n direction. These entries must be cleared in the output matrix before further operations (cell-to-cell translations) can be performed if we are to obtain the correct answers. This can be accomplished by applying simple elementwise mask to the output matrix that zeros the bogus entries. Unfortunately, this operation is inefficient in the Fourier transform variable, as we must convolve the FFT of the mask with the FFT of the output matrix. Rather, we should perform an inverse FFT and apply the mask before continuing. While this procedure is more efficient that the convolution on the FFT side, it prevents us from performing all our cell-to-cell transformations in the transform variable: for example, we must transform the result of our multipole-to-local conversion back to (m, n) space and apply the mask before we can perform local-to-local conversions. We may, however, combine the results of conversions of the same type in Fourier space; this is fortunate, since combining multipole-to-local conversions in (m, n) space is the most computationally expensive part of the FMM and the synthesis can be done quickly in Fourier space.

Finally, Greengard and Rokhlin [3] have noted another numerically instability in the FFT-accelerated FMM. The instability arises from the warping operation, which makes uses of coefficients that can exceed the range of double-precision numbers, and improperly scaled dimensions (i.e, when the coefficients in the multipole matrix are large while the coefficients of the transfer matrix are small). Greengard and Rokhlin solved this problem by using a polynomial scaling scheme [3], while Elliot and Brown used a block decomposition scheme to mollify the instabilities [6]. The reader may consult the relevant references for more details on these schemes.

5 Conclusions and Further Thoughts

We have seen how conversion of the multipole-to-local translation operation to the Fourier domain results in a significant reduction in the expect running time of the Fast Multipole Method. Although working in the Fourier transform space leads to several complications in the algorithm, these are not insurmountable.

The analysis presented here for the multipole-to-local conversion can also be performed for the multipole-to-multipole and local-to-local conversions; see [6] for some explicit formulae.

One could probably perform a similar FFT acceleration of the FMM for two-dimensional problems, but the computational savings would not be as great: the direct method has an expected running time of $O(p^2 N)$, while a FFT version would theoretically have an expected running time of $O((p \log p)N)$. Since p is small in practice, one would not enough savings to justify the extra effort unless N were very large.

References

[1] Leslie F. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems.* MIT Press, Cambridge, MA, 1988.

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

[3] L. Greengard and V. Rokhlin. On the efficient implementation of the fast multipole algorithm. Technical Report RR-515, Dept. of Computer Science, Yale University, New Haven, CT, 1988.

[4] William H. Press et al. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, Cambridge-New York-Port Chester-Melbourne-Sydney, second edition, 1997.

[5] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *J. Comp. Phys.*, 60:187–207, 1983.

[6] William D. Elliott and Board, Jr., John A. Fast fourier transform accelerated fast multipole algorithm. *SIAM J. Sci. Comput.*, 17:398–415, March 1996.