AD-A191 167

DTIC
SELECTED
MAR 2 4 1988
D

LUX ET VERITAS

On the Efficient Implementation
of the Fast Multipole Algorithm

L. Greengard and V. Rokhlin

Research Report YALEU/DCS/RR-602
February 1988

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

88 3 18 025

The Fast Multipole Method (FMM) is a recently developed algorithm for the evaluation of potential fields in particle systems. In order to evaluate the fields induced by a set of N charges (or masses) on each other, the FMM requires order $O(N)$ work rather than the $O(N^2)$ work required by the direct evaluation of pairwise interactions. The constant of proportionality for the method depends on the cost of applying a translation operator to a multipole or Taylor expansion. In existing implementations, this is $O(p^2)$ in two dimensions and $O(p^4)$ in three, where p is the degree of the expansion. In this paper we describe a procedure permitting translation operators to be applied to p^{th} degree expansions for a cost proportional to $p \cdot \log p$ in two dimensions, and $p^2 \cdot \log p$ in three. The incorporation of this technique into the FMM scheme speeds up the execution of two-dimensional single precision codes by a factor of two or three, and the execution of three-dimensional codes by roughly a factor of eight.
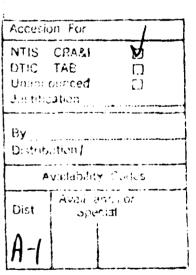
On the Efficient Implementation
of the Fast Multipole Algorithm

L. Greengard and V. Rokhlin

Research Report YALEU/DCS/RR-602
February 1988

1 Introduction

The Fast Multipole Method (FMM) is a recently developed scheme for the evaluation of potential fields which has been used in a variety of contexts, including the numerical solution of the Laplace equation, fluid dynamics, particle simulations, and numerical complex analysis [8,4,3,1,7]. In order to evaluate the fields induced by a collection of N charges on each other, the FMM requires an amount of work proportional to N rather than N^2. The constant of proportionality depends, in turn, on the cost of applying a translation operator to a multipole or Taylor expansion. In existing codes, the latter is $O(p^2)$ in two dimensions, and $O(p^4)$ in three, where p is the degree of the expansion (see [8,4,5]). The value of p used by the FMM is roughly equal to $\log_2(\frac{1}{\epsilon})$, where ϵ is the desired precision of the calculation.

This paper can be viewed as the sequel to [4] and [5]. Here, we describe a procedure permitting the translation operators for the Laplace equation to be applied to arbitrary p^{th} degree expansions (both multipole and Taylor) for a cost proportional to $p \cdot \log p$ in two dimensions, and $p^2 \cdot \log p$ in three. The incorporation of this technique into the general FMM scheme should speed up the execution of two-dimensional codes by a factor of two or three compared with the results reported in [3], and is expected to make large-scale three-dimensional simulations affordable.

2 Theory in Two Dimensions

2.1 Translation operators

The following three lemmas describe translation operators for multipole and power series expansions for the Laplace equation in $\mathbf{R}^2$, and provide error bounds allowing the manipulation of these expansions in the manner required by the Fast Multipole Method. Detailed proofs of Lemmas 2.1 - 2.3 below can be found in [4]. The first, Lemma 2.1, supplies a mechanism for shifting the center of a multipole expansion.

Lemma 2.1 (Translation of a Multipole Expansion) *Suppose that*

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

is a multipole expansion of the potential due to a set of m charges of strengths $q_1, q_2, \ldots, q_m$, all of which are located inside the circle D of radius R with center at z_0. Then for z outside the circle D_1 of radius $(R + |z_0|)$ and center at the origin,

$$\phi(z) = a_0 \log(z) + \sum_{l=1}^{\infty} \frac{b_l}{z^l}, \tag{2}$$

where

1

$$b_l = -\frac{a_0 z_0^l}{l} + \sum_{k=1}^{l} a_k z_0^{l-k} \binom{l-1}{k-1}, \tag{3}$$

with $\binom{l}{k}$ the binomial coefficients. Furthermore, for any $p \geq 1$,

$$\left| \phi(z) - a_0 \log(z) - \sum_{l=1}^{p} \frac{b_l}{z^l} \right| \leq \left(\frac{A}{1 - \left| \frac{|z_0| + R}{z} \right|} \right) \left| \frac{|z_0| + R}{z} \right|^{p+1} \tag{4}$$

with A defined by the formula

$$A = \sum_{i=1}^{m} | q_i |. \tag{5}$$

Lemma 2.2 describes the conversion of a multipole expansion into a local (Taylor) expansion inside a circular region of analyticity.

Lemma 2.2 (Conversion of a Multipole Expansion into a Local Expansion) *Suppose that m charges of strengths $q_1, q_2, ..., q_m$ are located inside the circle D_1 with radius R and center at z_0, and that $|z_0| > (c+1)R$ with $c > 1$. Then the corresponding multipole expansion (1) converges inside the circle D_2 of radius R centered about the origin. Inside D_2, the potential due to the charges is described by a power series:*

$$\phi(z) = \sum_{l=0}^{\infty} b_l \cdot z^l, \tag{6}$$

where

$$b_0 = a_0 \log(-z_0) + \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} (-1)^k, \tag{7}$$

and

$$b_l = -\frac{a_0}{l \cdot z_0^l} + \frac{1}{z_0^l} \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} \binom{l+k-1}{k-1} (-1)^k, \quad for \; l \geq 1. \tag{8}$$

Furthermore, for any $p \geq max \left(2, \frac{2c}{c-1} \right)$, an error bound for the truncated series is given by

$$\left| \phi(z) - \sum_{l=0}^{p} b_l \cdot z^l \right| < \frac{A(4e(p+c)(c+1) + c^2)}{c(c-1)} \left(\frac{1}{c} \right)^{p+1}, \tag{9}$$

where A is defined in (5) and e is the base of natural logarithms.

Lemma 2.3 provides a formula for shifting the center of a local expansion within a region of analyticity. In the formula (10) below, n can be either a natural number or ∞. In both cases, expression (10) is an exact one, and no error bound is needed.

Lemma 2.3 (Translation of a Local Expansion) *For any complex z_0, z and $\{a_k\}$, $k = 0, 1, 2, \ldots, n$,*

$$\sum_{k=0}^{n} a_k (z - z_0)^k = \sum_{l=0}^{n} b_l \cdot z^l \tag{10}$$

where

$$b_l = \sum_{k=l}^{n} a_k \binom{k}{l} (-z_0)^{k-l} \tag{11}$$

2.2 Convolution Form of Translation Operators

For the sake of simplicity, in the remainder of this paper we will assume that the zero-order term in all multipole expansions vanishes. This will not affect the resulting complexity estimates and allows us to ignore the logarithmic terms in all the preceding expressions.

Note now that equations (3), (7) - (8), and (10) define three infinite-dimensional linear operators connecting the sequences $\{a_i\}$ and $\{b_i\}$. In numerical calculations, both multipole expansions of the form (1) and Taylor expansions of the form (6), are truncated after a finite number of terms, and the resulting sums are used in place of the original series. The validity of this approximation is established by the error bounds (4) and (9). We therefore restrict our attention to the finite-dimensional versions of formulas (3), (7), (8), and (10). Truncating all expansions in Lemmas 2.1 - 2.3 after p terms ($p \geq 1$) leads to three linear operators $U^p, V^p, W^p : \mathbb{C} \times \mathbb{C}^p \to \mathbb{C}^p$ defined as follows. The vector $u = U^p(z_0, x)$ is given by

$$u_0 = x_0 = 0 \tag{12}$$

and

$$u_l = \sum_{k=1}^{l} x_k z_0^{l-k} \binom{l-1}{k-1} \qquad \text{for } l = 1, \ldots, p-1 . \tag{13}$$

The vector $v = V^p(z_0, x)$ is given by

$$v_l = \frac{1}{z_0^l} \sum_{k=1}^{p-1} \frac{x_k}{z_0^k} \binom{l+k-1}{k-1} (-1)^k \qquad \text{for } l = 1, \ldots, p-1 . \tag{14}$$

Finally, the vector $w = W^p(z_0, x)$ is given by

$$w_l = \sum_{k=l}^{p-1} x_k \binom{k}{l} \cdot (-z_0)^{k-l} \qquad \text{for } l = 1, \ldots, p-1 . \tag{15}$$

Expanding out the binomial coefficient, the first translation formula (13) may be written in the following manner:

$$\frac{u_l}{(l-1)!} = \sum_{k=1}^{l} \frac{x_k}{(k-1)!} \cdot \frac{z_0^{l-k}}{(l-k)!} \; . \tag{16}$$

Similarly, formulas (14) and (15) become

$$v_l \cdot l! \cdot (-1)^l = \sum_{k=1}^{p-1} \frac{x_k}{(k-1)!} \cdot \frac{(l+k-1)!}{(-z_0)^{l+k}} \; , \tag{17}$$

and

$$w_l \cdot l! \cdot (-1)^l = \sum_{k=l}^{p-1} \frac{x_k \cdot k!}{(-1)^k} \cdot \frac{z_0^{k-l}}{(k-l)!} \tag{18}$$

respectively.

In this form, it is clear that all three translation operators can be viewed as convolutions preceded and followed by diagonal scalings. They can therefore be computed by means of the Fast Fourier Transform (FFT) and the total cost of applying a translation operator to a p - term expansion can be reduced from $O(p^2)$ to $O(p \cdot \log p)$. From the point of view of complexity theory, this is a significant improvement. However, the FMM requires fewer than 20 terms for single precision calculations, and fewer than 50 terms for double precision, so that a straightforward application of the convolution theorem does not reduce the actual computation times significantly. Fortunately, a more considered use of the Fourier transform changes the situation considerably, and gives rise to a significantly faster implementation. In order to describe this approach, we will require a more formal analysis of the translation operators.

2.3 Diagonalization of Translation Operators

We proceed by introducing some of the notation needed in this section. Suppose that p and q are natural numbers with $p \geq q$. For a vector $x \in \mathbb{C}^p$, we will denote by $T^{p,q}(x)$ the vector $y \in \mathbb{C}^q$ defined by the formula

$$y_i = x_i \quad \text{for } i = 0, \ldots, q-1 \; , \tag{19}$$

and refer to the mapping $T^{p,q}$ as truncation. For any vector $y \in \mathbb{C}^q$, we will denote by $E^{q,p}(y)$ the vector $x \in \mathbb{C}^p$ defined by the formula

$$\begin{aligned} x_i &= y_i \quad \text{for } i = 0, \ldots, q-1 \; , \\ x_i &= 0 \quad \text{for } i = q, \ldots, p-1 \; , \end{aligned} \tag{20}$$

and refer to the mapping $E^{q,p}$ as embedding. Finally, for any natural number p, we will denote by $\mathcal{F}^p$ the mapping $\mathbb{C}^p \rightarrow \mathbb{C}^p$ defined by the Discrete Fourier Transform.

In the Fast Multipole Method, the operators U^p, V^p, W^p are repeatedly applied with various translation vectors z_0 to various expansions x representing the fields generated by specific combinations of particles. We will denote by $\overline{U^p}, \overline{V^p}$, and $\overline{W^p}$ the matrices representing the operators U^p, V^p, and W^p, respectively. The following three theorems describe the decomposition of these matrices in the desired form.

4

Theorem 2.1 *For any integer $p \geq 1$ and $z_0 \in \mathbb{C}$,*

$$
\begin{aligned}
\overline{U^p} &= S^p \circ T^{2p,p} \circ K_1^p \circ E^{p,2p} \circ (S^p)^{-1} \\
&= S^p \circ T^{2p,p} \circ (\mathcal{F}^{2p})^{-1} \circ D_1^{2p} \circ \mathcal{F}^{2p} \circ E^{p,2p} \circ (S^p)^{-1},
\end{aligned}
\tag{21}
$$

where S^p is a diagonal matrix defined by the formula

$$
(S^p)_{ii} = (i-1)! \, ,
\tag{22}
$$

K_1^p is a periodic convolution with the finite sequence R_1^p given by

$$
(R_1^p)_i = \begin{cases} \frac{z_0^i}{i!}, & \text{for } i = 0, \ldots, p-1; \\ 0, & \text{for } i = p, \ldots, 2p-1 \, , \end{cases}
\tag{23}
$$

and D_1^{2p} is a diagonal $2p \times 2p$ matrix with

$$
(D_1^{2p})_{ii} = (\mathcal{F}^{2p} \circ E^{p,2p}(R_1^p))_i.
\tag{24}
$$

Proof. The first equality in (21) follows from formula (16). The second is an immediate consequence of the (discrete) convolution theorem.

$\square$

Theorem 2.2 *For any integer $p \geq 1$ and $z_0 \in \mathbb{C}$,*

$$
\begin{aligned}
\overline{V^p} &= (Q^p)^{-1} \circ T^{2p,p} \circ K_2^p \circ P^{2p} \circ E^{p,2p} \circ (S^p)^{-1} \\
&= (Q_p)^{-1} \circ T^{2p,p} \circ (\mathcal{F}^{2p})^{-1} \circ D_2^{2p} \circ \mathcal{F}^{2p} \circ P^{2p} \circ E^{p,2p} \circ (S^p)^{-1} \\
&= (Q_p)^{-1} \circ T^{2p,p} \circ (\mathcal{F}^{2p})^{-1} \circ D_2^{2p} \circ P^{2p} \circ \mathcal{F}^{2p} \circ E^{p,2p} \circ (S^p)^{-1} \, ,
\end{aligned}
\tag{25}
$$

where S^p is defined by equation (22), Q^p is a diagonal matrix defined by

$$
(Q^p)_{ii} = i! \cdot (-1)^i \, ,
\tag{26}
$$

K_2^p is a periodic convolution with the finite sequence R_2^p given by

$$
(R_2^p)_i = \begin{cases} \frac{(i-1)!}{(-z_0)^i}, & \text{for } i = 0, \ldots, p-1; \\ 0, & \text{for } i = p, \ldots, 2p-1 \, , \end{cases}
\tag{27}
$$

D_2^{2p} is a diagonal $2p \times 2p$ matrix with

$$
(D_2^{2p})_{ii} = (\mathcal{F}^{2p}(R_2^p))_i \, ,
\tag{28}
$$

and P^{2p} is a permutation operator defined by the formula

$$
\{P^{2p}(x)\}_j = x_{-j \bmod 2p}
\tag{29}
$$

Proof. The first equality in (25) follows from formula (17). The permutation matrix P^{2p} simply changes the indexing scheme so that the summation is in convolution form. The third equality is obtained from the second by the observation that P^{2p} commutes with the discrete Fourier transform.

$\square$

Theorem 2.3 *For any integer $p \geq 1$ and $z_0 \in \mathbb{C}$,*

$$
\begin{aligned}
\overline{W^p} &= (Q^p)^{-1} \circ T^{2p,p} \circ K_3^p \circ E^{p,2p} \circ Q^p \\
&= (Q^p)^{-1} \circ T^{2p,p} \circ (\mathcal{F}^{2p})^{-1} \circ D_3^{2p} \circ \mathcal{F}^{2p} \circ E^{p,2p} \circ Q^p \ ,
\end{aligned} \tag{30}
$$

where Q^p is given by (26), K_3^p is a periodic convolution with the finite sequence R_3^p defined by

$$
(R_3^p)_i = \begin{cases} 1, & \text{for } i = 0; \\ 0, & \text{for } i = 1, \ldots, p; \\ \frac{z_0^{2p-i}}{(2p-i)!}, & \text{for } i = p+1, \ldots, 2p-1 \ , \end{cases} \tag{31}
$$

and D_3^{2p} is a diagonal $2p \times 2p$ matrix with

$$
(D_3^{2p})_{ii} = (\mathcal{F}^{2p} \circ E^{p,2p}(R_3^p))_i \ . \tag{32}
$$

Proof. The first equality in (30) follows from formula (18). The transfer function R_3^p has been ordered in the manner indicated in (31) so that the summation is in convolution form.

$\square$

3 Incorporation into FMM

Suppose now that we would like to apply the matrix $\overline{U^p}$ to a vector $x \in \mathbb{C}^p$. Using Theorem 2.1, we can evaluate the product $\overline{U^p}(x)$ in the following five steps:

A) Apply the diagonal matrix $(S^p)^{-1}$ to the vector x.

B) Embed the vector $(S^p)^{-1}(x)$ in $\mathbb{C}^{2p}$ by zero-padding, and compute its discrete Fourier transform with the FFT.

C) Multiply the resulting vector by the diagonal matrix D_1^{2p} given by (24).

D) Compute the inverse discrete Fourier transform of the resulting vector with the FFT.

E) Truncate this length $2p$ vector to one of length p, and apply to it the diagonal matrix S^p to obtain the desired result $\overline{U^p}(x)$.

Obviously, the cost of the above procedure is dominated by that of steps B and D, each of which is proportional to $p \cdot \log p$. Thus, the cost of applying the matrix $\overline{U^p}$ to an arbitrary vector has been reduced from $O(p^2)$ to $O(p \cdot \log p)$. A similar procedure permits the matrices $\overline{V^p}$ and

6

$\overline{W^p}$ to be applied to arbitrary vectors in a "fast" manner, with the help of Theorems 2.2 and 2.3, respectively. Nevertheless, as mentioned previously, the number p varies roughly between 10 and 50 in practical situations, so that this observation does not reduce the computation times significantly.

The surprising fact is that the bulk of the Fast Multipole Method can be carried out in Fourier space, where the translation operators are diagonal. More specifically, we recall that in Step 1 of the FMM as described in [4], we form the multipole expansions for all boxes at the finest level of refinement. In Step 2, we then obtain the multipole expansions for all boxes at all higher levels by merging (Theorem 2.1), each translation being carried out according to formula (16). But suppose that we add a new Step 1a, in which each expansion at the finest level is scaled by S^p and Fourier-transformed (Steps A and B above). The diagonal matrix D_1^{2p}, which depends only on the shift vector z_0, can then be applied to each expansion. The result is the shifted expansion, albeit in Fourier space and scaled by the matrix S^p. The merger of the four child box expansions in Step 2 can then be carried out by adding together the coefficients in Fourier space. But this sum is already in the form needed to carry out the subsequent shift. The way in which the convolution is written in formula (16) should make this clear. Thus the entire upward pass of the FMM can be carried out in Fourier space and we have the multipole expansion for each box at each level in the transform domain.

Let us turn then to the downward pass in which multipole expansions are converted to local expansions and local expansions are transmitted to child boxes. By examination of formula (17), it is clear that these scaled, Fourier-transformed multipole expansions are already in appropriate form for the conversion to be a diagonal procedure. In the decomposition of the operator V^p given by formula (25), we have in essence already applied the operators

$$\mathcal{F}^{2p} \circ E^{p,2p} \circ (S^p)^{-1} \ , \tag{33}$$

so that the shift corresponds to multiplication by the permutation operator P^{2p} given by (29) and the matrix D_2^{2p} given by (28). The new expansion is the desired local expansion, again in Fourier space and scaled by Q^p. The local expansions can then be accumulated in this dual form. Finally, in the second loop of Step 3, the net local expansions are transmitted to the child boxes by applying the operator W^p. But, as before, we have already implicitly applied the operators

$$\mathcal{F}^{2p} \circ E^{p,2p} \circ Q^p \ , \tag{34}$$

so that the shift corresponds to multiplication by the matrix D_3^{2p} given by formula (32). The resulting expansion is the local expansion for the child box, in Fourier space and scaled by Q^p. Once the finest level is reached, it remains only to apply the operators

$$(Q^p)^{-1} \circ T^{2p,p} \circ (\mathcal{F}^{2p})^{-1} \tag{35}$$

to each local expansion, in order to obtain the coefficients in the original coordinate space.

In the original formulation of the method, as described in [4], the amount of work required by all the shifting procedures is of the order

$$N_k \cdot 27 \cdot p^2 \tag{36}$$

complex operations, where N_k is the number of boxes in the refinement structure. As described in this section, it costs roughly

$$N_k \cdot 27 \cdot p + N_k \cdot 10 \cdot p \cdot \log_2 p \qquad (37)$$

complex operations, assuming that one application of the FFT to a vector of length p costs $5 \cdot p \cdot \log_2 p$ operations. Even for relatively small p, this is a significant improvement.

Unfortunately, the strategy indicated above will not work well when the desired accuracy is high (p is large). The reason for this is that the convolution operators contain factorial terms which exceed the precision of the machine fairly quickly. This problem can be overcome by scaling, leads to a more complicated scheme, and is addressed in more detail in section 5.

4 Theory in Three Dimensions

The three-dimensional version of the FMM is based on spherical harmonic expansions. These arise from consideration of the Laplace equation in spherical coordinates

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial \Phi}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial \Phi}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial \, 2\Phi}{\partial \phi^2} = 0. \qquad (38)$$

The standard solution of this equation by separation of variables results in an expression for the field as a series, the terms of which are known as spherical harmonics.

$$\Phi = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \left(L_n^m r^n + \frac{M_n^m}{r^{n+1}} \right) Y_n^m(\theta, \phi) \qquad (39)$$

In the above expansion, the terms $Y_n^m(\theta, \phi) r^n$ are usually referred to as spherical harmonics of degree n, the terms $Y_n^m(\theta, \phi)/r^{n+1}$ are called spherical harmonics of degree $-n-1$, and the coefficients L_n^m and M_n^m are known as the moments of the expansion. In a far field (multipole) expansion, the coefficients L_n^m are set to zero. In a local (Taylor) expansion, the coefficients M_n^m are zero.

The following three theorems are the generalizations to three dimensions of Lemmas 2.1, 2.2, and 2.3. Theorem 4.3 below can be found in a somewhat different form in the literature [2,9]. Theorems 4.1 and 4.2 are recent results, described by the authors in [5,6].

Theorem 4.1 (Translation of a Multipole Expansion) *Suppose that l charges of strengths $q_1, q_2, \cdots, q_l$ are located inside the sphere D of radius a with center at $Q = (\rho, \alpha, \beta)$, and that for points $P = (r, \theta, \phi)$ outside D, the potential due to these charges is given by the multipole expansion*

$$\Phi(P) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{O_n^m}{r'^{n+1}} \cdot Y_n^m(\theta', \phi') , \qquad (40)$$

where $P - Q = (r', \theta', \phi')$. Then for any point $P = (r, \theta, \phi)$ outside the sphere D_1 of radius $(a + \rho)$,

8

$$\Phi(P) = \sum_{j=0}^{\infty} \sum_{k=-j}^{j} \frac{N_j^k}{r^{j+1}} \cdot Y_j^k(\theta, \phi) , \tag{41}$$

where

$$N_j^k = \sum_{n=0}^{j} \sum_{m=-n}^{n} \frac{O_{j-n}^{k-m} \cdot J_m^{k-m} \cdot A_n^m \cdot A_{j-n}^{k-m} \cdot \rho^n \cdot Y_n^{-m}(\alpha, \beta)}{A_j^k} , \tag{42}$$

with A_r^s defined by

$$A_n^m = \frac{(-1)^n}{\sqrt{(n-m)! \cdot (n+m)!}} , \tag{43}$$

and where

$$J_m^{m'} = \begin{cases} (-1)^{min(|m'|,|m|)}, & \text{if } m \cdot m' < 0; \\ 1, & \text{otherwise.} \end{cases} \tag{44}$$

Furthermore, for any $p \geq 1$,

$$\left| \Phi(P) - \sum_{j=0}^{p} \sum_{k=-j}^{j} \frac{N_j^k}{r^{j+1}} \cdot Y_j^k(\theta, \phi) \right| \leq \left(\frac{\sum_{i=1}^{l} |q_i|}{r - (a+\rho)} \right) \left(\frac{a+\rho}{r} \right)^{p+1} . \tag{45}$$

Theorem 4.2 (Conversion of a Multipole Expansion into a Local Expansion) *Suppose that l charges of strengths $q_1, q_2, \cdots, q_l$ are located inside the sphere D_Q of radius a with center at $Q = (\rho, \alpha, \beta)$, and that $\rho > (c+1)a$ with $c > 1$ Then the corresponding multipole expansion (40) converges inside the sphere D_0 of radius a centered at the origin. Inside D_0, the potential due to the charges $q_1, q_2, \cdots, q_l$ is described by a local expansion:*

$$\Phi(P) = \sum_{j=0}^{\infty} \sum_{k=-j}^{j} N_j^k \cdot Y_j^k(\theta, \phi) \cdot r^j , \tag{46}$$

where

$$N_j^k = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{O_n^m \cdot J_k^{n,m} \cdot A_n^m \cdot A_j^k \cdot Y_{j+n}^{m-k}(\alpha, \beta)}{A_{j+n}^{m-k} \cdot \rho^{j+n+1}} , \tag{47}$$

with A_r^s defined by equation (49) and where

$$J_m^{n',m'} = \begin{cases} (-1)^{n'}(-1)^{min(|m'|,|m|)}, & \text{if } m \cdot m' > 0; \\ (-1)^{n'}, & \text{otherwise.} \end{cases} \tag{48}$$

Furthermore, for any $p \geq 1$,

$$\left| \Phi(P) - \sum_{j=0}^{p} \sum_{k=-j}^{j} N_j^k \cdot Y_j^k(\theta, \phi) \cdot r^{j+1} \right| \leq \left(\frac{\sum_{i=1}^{l} |q_i|}{ca - a} \right) \left(\frac{1}{c} \right)^{p+1} . \tag{49}$$

Theorem 4.3 (Translation of a Local Expansion)

Let $Q = (\rho, \alpha, \beta)$ be the origin of a local expansion

$$\Phi(P) = \sum_{n=0}^{p} \sum_{m=-n}^{n} O_n^m \cdot Y_n^m(\theta', \phi') \cdot r'^n , \tag{50}$$

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

$$\Phi(P) = \sum_{j=0}^{p} \sum_{k=-j}^{j} N_j^k \cdot Y_j^k(\theta, \phi) \cdot r^j , \tag{51}$$

where

$$N_j^k = \sum_{n=j}^{p} \sum_{m=-n}^{n} \frac{O_n^m \cdot J_{n-j,m-k}^m \cdot A_{n-j}^{m-k} \cdot A_j^k \cdot Y_{n-j}^{m-k}(\alpha, \beta) \cdot \rho^{n-j}}{A_n^m} , \tag{52}$$

with A_r^s defined by equation (43) and where

$$J_{n,m}^{m'} = \begin{cases} (-1)^n(-1)^m, & \text{if } m \cdot m' < 0; \\ (-1)^n(-1)^{m'-m}, & \text{if } m \cdot m' > 0 \text{ and } |m'| < |m|; \\ (-1)^n, & \text{otherwise.} \end{cases} \tag{53}$$

4.1 Convolution Form of Translation Operators

As in the two-dimensional case, equations (42), (47), and (52) define three infinite-dimensional linear operators connecting the sequences $\{O_n^m\}$ and $\{N_n^m\}$. Again, in practice, both multipole expansions of the form (40) and Taylor expansions of the form (46), are truncated after a finite number of terms, and the resulting sums are used in place of the original series. The error bounds (45) and (49) indicate the accuracy of these approximations. We restrict our attention therefore to the finite-dimensional versions of formulas (42), (47), and (52). Truncating all expansions in Lemmas 4.1 - 4.3 after the pth degree ($p \geq 1$) leads to three linear operators $U^p, V^p, W^p : \mathbf{R}^3 \times \mathbb{C}^{p \times 2p+1} \to \mathbb{C}^{p \times 2p+1}$.

We proceed now to write the translation operators in convolution form. To do so, we will require the following obvious lemma expressing the constants $J_m^{m'}$, $J_m^{n',m'}$, and $J_{n,m}^{m'}$ in terms of powers of $i = \sqrt{-1}$.

Lemma 4.1 *The constant $J_m^{m'}$ in equation (44) is given by*

$$J_m^{m'} = i^{|m+m'|-|m|-|m'|} \tag{54}$$

The constant $J_m^{n',m'}$ in equation (48) is given by

$$J_m^{n',m'} = (-1)^{n'} \cdot i^{|m-m'|-|m|-|m'|} \tag{55}$$

The constant $J_{n,m}^{m'}$ in equation (53) is given by

10

$$J_m^{m'} = (-1)^{n'} \cdot i^{|m-m'|+|m|-|m'|} \tag{56}$$

The first truncated formula (42) may then be written in the following manner:

$$\frac{N_j^k \cdot A_j^k}{i^{|k|}} = \sum_{n=0}^{j} \sum_{m=-n}^{n} \frac{O_{j-n}^{k-m} \cdot A_{j-n}^{k-m}}{i^{|k-m|}} \cdot \frac{A_n^m \cdot \rho^n \cdot Y_n^{-m}(\alpha,\beta)}{i^{|m|}} \ , \tag{57}$$

Similarly, formulas (47) and (52) become

$$\frac{N_j^k \cdot i^{|k|}}{A_j^k \cdot (-1)^j} = \sum_{n=0}^{p} \sum_{m=-n}^{n} \frac{O_n^m \cdot A_n^m}{i^{|m|}} \cdot \frac{Y_{j+n}^{m-k}(\alpha,\beta) \cdot i^{|k-m|}}{(-1)^{n+j} \cdot A_{j+n}^{m-k} \cdot \rho^{j+n+1}} \ , \tag{58}$$

and

$$\frac{N_j^k \cdot i^{|k|}}{A_j^k \cdot (-1)^j} = \sum_{n=j}^{p} \sum_{m=-n}^{n} \frac{O_n^m \cdot i^{|m|}}{A_n^m \cdot (-1)^n} \cdot \frac{A_{n-j}^{m-k} \cdot Y_{n-j}^{m-k}(\alpha,\beta) \cdot \rho^{n-j}}{i^{|m-k|}} \ , \tag{59}$$

respectively.

It is clear that all three translation operators can be viewed as two-dimensional convolutions preceded and followed by diagonal scalings. They can be computed by means of the FFT, and the total cost of applying a translation operator to a pth degree expansion can be reduced from $O(p^4)$ to $O(p^2 \cdot \log p)$. By contrast with the two-dimensional case, this already provides a noticeable improvement in computation time, since the net gain grows as $p^2/\log p$ rather than as $p/\log p$. As with the two-dimensional case, however, the entire algorithm can be formulated in the Fourier transform domain, giving rise to an even faster numerical scheme.

4.2 Diagonalization of Translation Operators

As in the two-dimensional case, we proceed by introducing some notation. Suppose that p and q are natural numbers with $p \geq q$. For a vector $x \in \mathbb{C}^{p \times 2p+1}$, we will denote by $T^{p,q}(x)$ the vector $y \in \mathbb{C}^{q \times 2q+1}$ defined by the formula

$$y_i^j = x_i^j \quad \text{for } i = 0, \ldots, q-1 \text{ and } j = -i, \ldots, i \tag{60}$$

and refer to the mapping $T^{p,q}$ as truncation. For any vector $y \in \mathbb{C}^{q \times 2q+1}$, we will denote by $E^{q,p}(y)$ the vector $x \in \mathbb{C}^{p \times 2p+1}$ defined by the formula

$$x_i^j = y_i^j \quad \text{for } i = 0, \ldots, q-1 \text{ and } j = -i, \ldots, i \ ,$$
$$x_i = 0 \quad \text{for } i = q, \ldots, p-1 \text{ and } j = -i, \ldots, i \ , \tag{61}$$

and refer to the mapping $E^{q,p}$ as embedding. Finally, for any natural number p, we will denote by $\mathcal{F}^p$ the mapping $\mathbb{C}^{p \times 2p+1} \to \mathbb{C}^{p \times 2p+1}$ defined by the two-dimensional discrete Fourier transform. We denote by $\overline{U^p}, \overline{V^p}$, and $\overline{W^p}$ the matrices representing the operators U^p, V^p, and W^p, respectively. The matrix decompositions described below are the analogs of Theorems 2.1 - 2.3.

Theorem 4.4 *For any integer $p \geq 1$ and $(\rho, \alpha, \beta) \in \mathbf{R}^3$,*

$$
\begin{aligned}
\overline{U^p} &= S^p \circ T^{2p,p} \circ K_1^p \circ E^{p,2p} \circ (S^p)^{-1} \\
&= S^p \circ T^{2p,p} \circ (\mathcal{F}^{2p})^{-1} \circ D_1^{2p} \circ \mathcal{F}^{2p} \circ E^{p,2p} \circ (S^p)^{-1},
\end{aligned}
\tag{62}
$$

where S^p is a diagonal operator defined by the formula

$$
\{S^p(x)\}_n^m = \frac{x_n^m \cdot A_n^m}{i^{|m|}},
\tag{63}
$$

K_1^p is a periodic two-dimensional convolution with the finite sequence $R_1^p \in \mathbb{C}^{2p \times 4p+1}$ given by

$$
(R_1^p)_n^m = \begin{cases} \frac{A_n^m \cdot \rho^n \cdot Y_n^{-m}(\alpha,\beta)}{i^{|m|}}, & \text{for } n = 0,\ldots,p\text{-}1 \text{ and } m = -n,\ldots,n \ ; \\ 0, & \text{otherwise} \ , \end{cases}
\tag{64}
$$

and D_1^{2p} is a diagonal operator with

$$
\{D_1^{2p}(x)\}_n^m = \{\mathcal{F}^{2p} \circ E^{p,2p}(R_1^p)\}_n^m.
\tag{65}
$$

Proof. The first equality in (62) follows from formula (57). The second is an immediate consequence of the (discrete) convolution theorem.

$\square$

Theorem 4.5 *For any integer $p \geq 1$ and $(\rho, \alpha, \beta) \in \mathbf{R}^3$,*

$$
\begin{aligned}
\overline{V^p} &= (Q^p)^{-1} \circ T^{2p,p} \circ K_2^p \circ P^{2p} \circ E^{p,2p} \circ (S^p)^{-1} \\
&= (Q_p)^{-1} \circ T^{2p,p} \circ (\mathcal{F}^{2p})^{-1} \circ D_2^{2p} \circ \mathcal{F}^{2p} \circ P^{2p} \circ E^{p,2p} \circ (S^p)^{-1} \\
&= (Q_p)^{-1} \circ T^{2p,p} \circ (\mathcal{F}^{2p})^{-1} \circ D_2^{2p} \circ P^{2p} \circ \mathcal{F}^{2p} \circ E^{p,2p} \circ (S^p)^{-1} \ ,
\end{aligned}
\tag{66}
$$

where S^p is defined by equation (63), Q^p is a diagonal operator defined by

$$
\{Q^p(x)\}_n^m = \frac{x_n^m \cdot i^{|m|}}{A_n^m \cdot (-1)^m},
\tag{67}
$$

K_2^p is a periodic two-dimensional convolution with the finite sequence $R_2^p \in \mathbb{C}^{2p \times 4p+1}$ given by

$$
(R_2^p)_n^m = \begin{cases} \frac{Y_n^{-m}(\alpha,\beta) \cdot i^{|m|}}{(-1)^n \cdot A_n^m \cdot \rho^{n+1}} \ , & \text{for } n = 0,\ldots,p\text{-}1 \text{ and } m = -n,\ldots,n \ ; \\ 0, & \text{otherwise} \ , \end{cases}
\tag{68}
$$

D_2^{2p} is a diagonal operator with

$$
\{D_2^{2p}(x)\}_n^m = \{\mathcal{F}^{2p} \circ E^{p,2p}(R_2^p)\}_n^m \ ,
\tag{69}
$$

and P^{2p} is a permutation operator defined by the formula

$$
\{P^{2p}(x)\}_n^m = x_{-j_{\bmod 2p}}^m
\tag{70}
$$

Proof. The first equality in (66) follows from formula (58). The permutation matrix P^{2p} changes the indexing scheme so that the summation is in convolution form. The third equality is obtained from the second by the observation that P^{2p} commutes with the two-dimensional discrete Fourier transform.

$\square$

Theorem 4.6 *For any integer $p \geq 1$ and $(\rho, \alpha, \beta) \in \mathbf{R}^3$,*

$$\overline{W^p} = (Q^p)^{-1} \circ T^{2p,p} \circ K_3^p \circ E^{p,2p} \circ Q^p$$
$$= (Q^p)^{-1} \circ T^{2p,p} \circ (\mathcal{F}^{2p})^{-1} \circ D_3^{2p} \circ \mathcal{F}^{2p} \circ E^{p,2p} \circ Q^p \ , \tag{71}$$

where Q^p is given by (67), K_3^p is a periodic two-dimensional convolution with the finite sequence $R_3^p \in \mathbb{C}^{2p \times 4p+1}$ defined by

$$(R_3^p)_n^m = \begin{cases} 1, & \text{for } n = 0 \text{ and } m = 0; \\[2mm] \dfrac{Y_{2p-n}^{2p-m}(\alpha,\beta) \cdot i^{|2p-m|} \cdot (-1)^{2p-n}}{A_{2p-n}^{2p-m} \cdot \rho^{2p-n+1}} \ , & \text{for } n = p+1, \ldots, 2p\text{-}1, \\ & \qquad m = n\text{-}2p, \ldots, 2p\text{-}n; \\[2mm] 0, & \text{otherwise} , \end{cases} \tag{72}$$

and D_3^{2p} is a diagonal operator with

$$\{D_3^{2p}(x)\}_n^m = \{\mathcal{F}^{2p} \circ E^{p,2p}(R_3^p)\}_n^m \ . \tag{73}$$

Proof. The first equality in (72) follows from formula (59). The transfer function R_3^p has been ordered in the indicated manner so that the summation is in convolution form.

$\square$

Remark: The three-dimensional version of the FMM has been modified to incorporate the translation operators in diagonal form. The use of Fourier space and the description of this modification are essentially identical to that described in Section 3. We simply note here that in the original formulation of the method, as described in [5,6], the amount of work required by all the shifting procedures is of the order

$$N_k \cdot 875 \cdot p^4 \tag{74}$$

complex operations, where N_k is the number of boxes in the refinement structure. In the new implementation, it costs roughly

$$N_k \cdot 875 \cdot p^2 + N_k \cdot 10 \cdot p^2 \cdot \log_2 p \tag{75}$$

13

complex operations, assuming that one application of the FFT to a vector of length p costs $5 \cdot p \cdot \log_2 p$ operations. The improvement is clearly more dramatic than in the two dimensional case.

Remark: The constant 875 in the operation count above refers to the number of boxes at a fixed refinement level with which each box must interact. Zhao, in [10], makes an observation allowing this number to be reduced to approximately 200. The idea is simply that if each of eight child boxes are in the interaction list of a given box, then that box can transmit its multipole expansion once to the parent node rather than eight times to the individual children.

5 Numerical Stability

We turn now to the matter of numerical stability. As indicated in Section 3, when the desired precision is high and the number of terms in the expansions is large, the strategy delineated there will not work well. The convolution operators in both two and three dimensions contain terms which grow as $p!$, where p is the degree of the expansion. While the FFT is stable in the sense that it is a unitary operator, these terms quickly exceed the available machine precision.

In order to reformulate the FMM in a way which avoids this problem, we must reexamine the convolution form of the translation operators. In this discussion, we will concentrate on the two-dimensional algorithm. The numerical considerations in the three-dimensional case are essentially the same.

There are two measures which may be taken to reduce the dynamic range of the problem. The first is to split the input vector into two (or more) blocks, and to transform each block separately. This requires that the translation procedures be reconstructed in block form. We will describe a second approach consisting simply of polynomial scaling, which allows the translation procedures to retain their form. For this purpose, we rewrite formula (16) in the following way:

$$\frac{u_l \cdot s^l}{(l-1)!} = \sum_{k=1}^{l} \frac{x_k s^k}{(k-1)!} \cdot \frac{z_0^{l-k} s^{l-k}}{(l-k)!} \, , \tag{76}$$

with s an arbitrary constant. Similarly, formulas (17) and (18) become

$$\frac{v_l \cdot l! \cdot (-1)^l}{s^l} = \sum_{k=1}^{p-1} \frac{x_k \cdot s^k}{(k-1)!} \cdot \frac{(l+k-1)!}{(-z_0)^{l+k} \cdot s^{l+k}} \, , \tag{77}$$

and

$$\frac{w_l \cdot l! \cdot (-1)^l}{s^l} = \sum_{k=l}^{p-1} \frac{x_k \cdot k! \cdot (-1)^k}{s^k} \cdot \frac{z_0^{k-l} \cdot s^{k-l}}{(k-l)!} \tag{78}$$

respectively. It remains to choose s. By inspection of the preceding formulas, it is clear that we wish to reduce the dynamic range of sequences whose n^{th} entry is of the form

$$\frac{n!}{(r \cdot s)^n} \quad \text{or} \quad \frac{(r \cdot s)^n}{n!} \, , \tag{79}$$

14

where $r = |z_0|$. This is true, not only of the transfer functions, but of the expansion sequences themselves. (It is easy to show that the multipole expansion coefficients at a given level of refinement grow approximately as r^n and the local expansion coefficients grow approximately as r^{-n}, where r is the average distance of a shift.)

A reasonable strategy for choosing s is to require that $r \cdot s = p$. To see this, let us consider the most troublesome sequence, namely the transfer function in equation (77). Unscaled, with $p = 16$, the sequence values range from $O(1)$ to $O(10^{35})$. With the above choice of s, the corresponding range is from $O(10^{-2})$ to $O(10^{-6})$, a reduction of 31 orders of magnitude. Unfortunately, this requires a different scaling constant at each refinement level, and the following modification of the FMM. Let us suppose that, at a given refinement level, the scaled translation operators have been applied. Before moving to the next level, either up or down, each expansion sequence must be rescaled. Since the scaling is done in the original coordinate space, this requires three steps:

1. Apply inverse FFT to each sequence.

2. Multiply k^{th} term in sequence by $(s_{old}/s_{new})^k$.

3. Apply forward FFT to each sequence.

The expansions will then be in appropriate form for the next set of translation procedures. The amount of extra work involved in the rescaling steps is roughly

$$\frac{N_k}{4} \cdot 10 \cdot p \cdot \log_2 p , \qquad (80)$$

where N_k is the number of boxes in the refinement structure. The factor of four arises from the fact that the rescaling of expansions is necessary at coarser levels, but not at the finest level. This is a clearly a small addition to the workload when compared with the order estimate for the whole algorithm (37), and does not affect the execution time in a substantial way.

It should be noted that scaling is not a universal remedy. The factorial and polynomial terms behave quite differently when p is large, and the dynamic range does grow. Nevertheless, for single precision calculations, the loss of accuracy can be held to less than one digit. For double precision calculations, approximately three digits are lost. If more precision is needed, a hybrid procedure can be constructed using both scaling and the block decomposition mentioned at the beginning of the section.

6   Numerical Results

Computer programs have been written in Fortran 77 using both the original formulations of the algorithm [4,5] and the formulations presented here. We compare the performance of these methods for three-dimensional free space calculations. Charged particles were randomly assigned positions in a cube, so that the resulting particle density was roughly uniform. The potential fields were computed in four ways: by the original algorithm in single precision, by the new algorithm in single precision, and directly in single and double precision. The direct

15

calculation of the field in double precision was used as a standard for comparing the relative accuracies of the other three methods. The number of particles varied between 1,000 and 64,000, with charge strengths randomly assigned between zero and one.

All calculations cited below have been carried out on a VAX-8600, and the results are summarized in Table 1. The first column of each table contains the number of particles N in the calculation. The second column contains the number of refinement levels $nlev$. In the remaining columns, the upper case letters T and E are used to denote the corresponding computational time and error, with the subscripts *new, old* and *dir* referring to the new algorithm, the original algorithm, and the direct (single-precision) calculation respectively. Columns 3 through 5 show the times, in seconds, required to compute the field by the three methods. The errors E_{new}, E_{old} and E_{dir} for the new, original and direct methods, respectively, are presented in the next three columns. They are defined by the formula

$$E = \left(\frac{\sum_{i=1}^{N} \left| f_i - \tilde{f}_i \right|^2}{\sum_{i=1}^{N} |f_i|^2} \right)^{1/2}$$

where f_i is the value of the field at the i-th particle position obtained by direct calculation in double precision and $\tilde{f}_i$ is the result obtained by one of the three methods being studied.

Remark: For the tests involving 8,000 or more particles, it was not considered practical to use the direct method to calculate the fields at all particle positions, since this would require prohibitive amounts of CPU time without providing much useful information. We, therefore, used the direct method to evaluate the field for only 100 of the particles, and used these results to evaluate the relative accuracies. The corresponding values of T_{dir} were estimated by extrapolation. The values of T_{old} for 32,000 and 64,000 points were also estimated by extrapolation.

On the basis of the data in Table 1, we may make the folowing observations:

1. The accuracies of the results obtained by the fast algorithms are in agreement with the error bounds given in (45) and (49).

2. The actual CPU time requirements of the fast algorithms appear to grow somwhat erratically and nonlinearly with N, since the linear bound for the execution time is well above the actual execution times until the number of particles is quite large.

3. By the time the number of particles reaches $64,000$ the new FMM is about 15 times faster than the direct method. This improvement is smaller than in the two-dimensional case, where the speed-up would be by a factor of 200. It should be pointed out, however, that the number of points required for the resolution of a given problem are much greater in three dimensions. $64,000$ points, for example, yield the resolution of a $40 \times 40 \times 40$ grid. In two dimensions, the same spatial refinement requires only 1600 points, at which point the fast algorithm yields a speed-up of only a factor of 8-10 (see [3]).

16

N	$nlev$	T_{new}	T_{old}	T_{dir}	E_{new}	E_{old}	E_{dir}
1000	3	23.4	222	17.6	$4.4\ 10^{-7}$	$4.4\ 10^{-7}$	$4.8\ 10^{-7}$
2000	3	59.0	343	71.5	$5.7\ 10^{-7}$	$5.6\ 10^{-7}$	$6.6\ 10^{-7}$
4000	3	179	504	291	$9.3\ 10^{-7}$	$9.3\ 10^{-7}$	$1.1\ 10^{-6}$
8000	4	403	3,939	1134	$3.5\ 10^{-6}$	$3.4\ 10^{-6}$	$1.2\ 10^{-6}$
16000	4	744	4,280	(4,490)	$3.8\ 10^{-6}$	$3.8\ 10^{-6}$	$3.1\ 10^{-6}$
32000	4	2490	(9,210)	(17,960)	$4.0\ 10^{-6}$	—	$4.1\ 10^{-6}$
64000	5	4,810	(44,230)	(71,830)	$4.5\ 10^{-6}$	—	$3.7\ 10^{-6}$

Table 1: CPU times and error estimates for N-body calculations in three space dimensions. Particles uniformly distributed. Degree of harmonic expansions $p = 8$.

7   Conclusions

A new implementation of the Fast Multipole Method has been developed by applying translation operators to both multipole and Taylor expansions in a system of coordinates where they are diagonal. We estimate that for single precision results, the speed-up for two-dimensional calculations is on the order of a factor of two or three. We have demonstrated that in three-dimensional calculations, the speed-up obtained varies between a factor of three and ten.

The irregularity of the speed-up, as well as the jumps in the CPU time requirements in Table 1, are due to the discrete number of refinement levels available to the non-adaptive algorithm. Work is in progress on an adaptive version of the method, which is the three-dimensional analog of the scheme described in [3]. We expect a significant improvement in the execution time, even for fairly homogeneous distributions of particles. The relative improvement of the new implementation should then remain fairly constant at a factor of eight. For double precision calculations, the relative improvement will be on the order of a factor of thirty-two.

References

[1] J. Ambrosiano, L. Greengard, and V. Rokhlin, *The Fast Multipole Method for Gridless Particle Simulations*, Comp. Phys. Comm., to appear.

[2] B. C. Carlson and G. S. Rushbrooke, *On the Expansion of a Coulomb Potential In Spherical Harmonics*, Proc. Cambridge Phil. Soc., 46 (1950), pp. 626-633.

[3] J. Carrier, L. Greengard, and V. Rokhlin, *A Fast Adaptive Multipole Algorithm for Particle Simulations*, Technical Report 496, Yale Computer Science Department, 1986 .

[4] L. Greengard and V. Rokhlin, *A Fast Algorithm for Particle Simulations*, J. Comput. Phys., 73 (1987), pp. 325-348

[5] L. Greengard and V. Rokhlin, *Rapid Evaluation of Potential Fields in Three Dimensions*, Technical Report 515, Yale Computer Science Department, 1987 .

[6] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, MIT Press, Cambridge, 1988.

[7] S. T. O'Donnell and V. Rokhlin, *A Fast Algorithm for the Numerical Evaluation of Conformal Mappings*, Technical Report 554, Yale Computer Science Department, 1986 .

[8] V. Rokhlin, *Rapid Solution of Integral Equations of Classical Potential Theory*, J. Comput. Phys., 60 (1985), pp. 187-207.

[9] M. E. Rose, *The Electrostatic Interaction of Two Arbitrary Charge Distributions*, J. Math. and Phys., 37 (1958), pp. 215-222 .

[10] F. Zhao, *A O(N) Algorithm for Three-dimensional N-body Simulations*, Research Report AI-TR-995, Massachusetts Institute of Technology, (1987).

END

DATE

FILMED

5-88

DTIC