JOURNAL OF COMPUTATIONAL PHYSICS **135,** 280–292 (1997)
ARTICLE NO. CP975706

# A Fast Algorithm for Particle Simulations*

L. Greengard and V. Rokhlin

*Department of Computer Science, Yale University, New Haven, Connecticut 06520*

Received June 10, 1986; revised February 5, 1987

An algorithm is presented for the rapid evaluation of the potential and force fields in systems involving large numbers of particles whose interactions are Coulombic or gravitational in nature. For a system of $N$ particles, an amount of work of the order $O(N^2)$ has traditionally been required to evaluate all pairwise interactions, unless some approximation or truncation method is used. The algorithm of the present paper requires an amount of work proportional to $N$ to evaluate all interactions to within roundoff error, making it considerably more practical for large-scale problems encountered in plasma physics, fluid dynamics, molecular dynamics, and celestial mechanics. © 1987 Academic Press

## 1. INTRODUCTION

The study of physical systems by means of particle simulations is well established in a number of fields and is becoming increasingly important in others. The most classical example is probably celestial mechanics, but much recent work has been done in formulating and studying particle models in plasma physics, fluid dynamics, and molecular dynamics [5].

There are two major classes of simulation methods. Dynamical simulations follow the trajectories of $N$ particles over some time interval of interest. Given initial positions $\{x_i\}$ and velocities, the trajectory of each particle is governed by Newton's second law of motion,

$$m_i \frac{d^2x_i}{dt^2} = -\nabla_i \Phi \quad \text{for } i = 1, ..., N,$$

where $m_i$ is the mass of the $i$th particle and the force is obtained from the gradient of a potential function $\Phi$. When one is interested in an equilibrium configuration of a set of particles rather than their time-dependent properties, an alternative approach is the Monte Carlo method. In this case, the potential function $\Phi$ has to be evaluated for a large number of configurations in an attempt to determine the potential minimum.

Reprinted from Volume 73, Number 2, December 1987, pages 325–348.
* The authors were supported in part by the Office of Naval Research under Grant N00014-82-K-0184.

We restrict our attention in this paper to the case where the potential (or force) at a point is a sum of pairwise interactions. More specifically, we consider potentials of the form

$$\Phi = \Phi_{\text{far}} + (\Phi_{\text{near}} + \Phi_{\text{external}}),$$

where $\Phi_{\text{near}}$ (when present) is a rapidly decaying potential (e.g., Van der Waals), $\Phi_{\text{external}}$ (when present) is independent of the number of particles, and $\Phi_{\text{far}}$, the far-field potential, is Coulombic or gravitational. Such models describe classical celestial mechanics and many problems in plasma physics and molecular dynamics. In the vortex method for incompressible fluid flow calculations [4], an important and expensive portion of the computation has the same formal structure (the stream function and the vorticity are related by Poisson's equation).

In a system of $N$ particles, the calculation of $\Phi_{\text{near}}$ requires an amount of work proportional to $N$, as does the calculation of $\Phi_{\text{external}}$. The decay of the Coulombic or gravitational potential, however, is sufficiently slow that all interactions must be accounted for, resulting in CPU time requirements of the order $O(N^2)$. In this paper a method is presented for the rapid (order $O(N)$) evaluation of these interactions for all particles.

There have been a number of previous efforts aimed at reducing the computational complexity of the $N$-body problem. Particle-in-cell methods [5] have received careful study and are used with much success, most notably in plasma physics. Assuming the potential satisfies Poisson's equation, a regular mesh is layed out over the computational domain and the method proceeds by:

(1) interpolating the source density at mesh points,

(2) using a "fast Poisson solver" to obtain potential values on the mesh,

(3) computing the force from the potential and interpolating to the particle positions.

The complexity of these methods is of the order $O(N + M \log M)$, where $M$ is the number of mesh points. The number of mesh points is usually chosen to be proportional to the number of particles, but with a small constant



0021-9991/97 $25.00

of proportionality so that $M \ll N$. Therefore, although the asymptotic complexity for the method is $O(N \log N)$, the computational cost in practical calculations is usually observed to be proportional to $N$. Unfortunately, the mesh provides limited resolution, and highly nonuniform source distributions cause a significant degradation of performance. Further errors are introduced in step (3) by the necessity for numerical differentiation to obtain the force.

To improve the accuracy of particle-in-cell calculations, short-range interactions can be handled by direct computation, while far-field interactions are obtained from the mesh, giving rise to so-called particle–particle/particle–mesh ($P^3M$) methods [5]. For an implementation of these ideas in the context of vortex calculations, see [1]. While these algorithms still depend for their efficient performance on a reasonably uniform distribution of particles, in theory they do permit arbitrarily high accuracy to be obtained. As a rule, when the required precision is relatively low, and the particles are distributed more or less uniformly in a rectangular region, $P^3M$ methods perform satisfactorily. However, when the required precision is high (as, for example, in the modeling of highly correlated systems), the CPU time requirements of such algorithms tend to become excessive.

Appel [2] introduced a ''gridless'' method for many-body simulation with a computational complexity estimated to be of the order $O(N \log N)$. It relies on using a monopole (center-of-mass) approximation for computing forces over large distances and sophisticated data structures to keep track of which particles are sufficiently clustered to make the approximation valid. For certain types of problems, the method achieves a dramatic speedup, compared to the naive $O(N^2)$ approach. It is less efficient when the distribution of particles is relatively uniform and the required precision is high.

The algorithm we present uses multipole expansions to compute potentials or forces to whatever precision is required, and the CPU time expended is proportional to $N$. The approach we use is similar to the one introduced in [7] for the solution of boundary value problems for the Laplace equation. In the following section, we describe the necessary analytical tools, while Section 3 is devoted to a detailed description of the method.

## 2. PHYSICAL AND MATHEMATICAL PRELIMINARIES

In this paper, we consider a two-dimensional physical model which consists of a set of $N$ charged particles with the potential and force obtained as the sum of pairwise interactions from Coulomb's law. Suppose that a point charge of unit strength is located at the point $(x_0, y_0) = \mathbf{x}_0 \in \mathbb{R}^2$. Then, for any $\mathbf{x} = (x, y) \in \mathbb{R}^2$ with $\mathbf{x} \neq \mathbf{x}_0$, the potential and electrostatic field due to this charge are described by the expressions

$$\phi_{\mathbf{x}_0}(x, y) = -\log(\|\mathbf{x} - \mathbf{x}_0\|)$$

and

$$E_{\mathbf{x}_0}(x, y) = \frac{(\mathbf{x} - \mathbf{x}_0)}{\|\mathbf{x} - \mathbf{x}_0\|^2},$$

respectively.

It is well known that $\phi_{\mathbf{x}_0}$ is harmonic in any region not containing the point $\mathbf{x}_0$. Moreover, for every harmonic function $u$, there exists an analytic function $w: \mathbb{C} \to \mathbb{C}$ such that $u(x, y) = \mathrm{Re}(w(x, y))$ and $w$ is unique except for an additive constant. In the remainder of the paper we will work with analytic functions, making no distinction between a point $(x, y) \in \mathbb{R}^2$ and a point $x + iy = z \in \mathbb{C}$. We note that

$$\phi_{\mathbf{x}_0}(\mathbf{x}) = \mathrm{Re}(-\log(z - z_0)),$$

and, following standard practice, we will refer to the analytic function $\log(z)$ as the potential due to a charge. As we develop expressions for the potential due to more complicated charge distributions, we will continue to use complex notation and will refer to the corresponding analytic functions themselves as the potentials. The following lemma is an immediate consequence of the Cauchy–Riemann equations.

LEMMA 2.1. *If $u(x, y) = \mathrm{Re}(w(x, y))$ describes the potential field at $(x, y)$, then the corresponding force field is given by*

$$\nabla u = (u_x, u_y) = (\mathrm{Re}(w'), -\mathrm{Im}(w')),$$

*where $w'$ is the derivative of $w$.*

The following lemma is used in obtaining the multipole expansion for the field due to $m$ charges.

LEMMA 2.2. *Let a point charge of intensity $q$ be located at $z_0$. Then for any $z$ such that $|z| > |z_0|$,*

$$\phi_{z_0}(z) = q \log(z - z_0) = q \left( \log(z) - \sum_{k=1}^{\infty} \frac{1}{k} \left( \frac{z_0}{z} \right)^k \right). \quad (2.1)$$

*Proof.* Note first that $\log(z - z_0) - \log(z) = \log(1 - z_0/z)$ and that $|z_0/z| < 1$. The lemma now follows from the expansion

$$\log(1 - \omega) = (-1) \sum_{k=1}^{\infty} \frac{w^k}{k},$$

which is valid for any $w$ such that $|w| < 1$. ∎

THEOREM 2.1. (Multipole expansion). *Suppose that $m$ charges of strengths $\{q_i,\ i = 1, ..., m\}$ are located at points $\{z_i,\ i = 1, ..., m\}$, with $|z_i| < r$. Then for any $z \in \mathbb{C}$ with $|z| > r$, the potential $\phi(z)$ is given by*

$$\phi(z) = Q \log(z) + \sum_{k=1}^{\infty} \frac{a_k}{z^k}, \tag{2.2}$$

*where*

$$Q = \sum_{i=1}^{m} q_i, \quad a_k = \sum_{i=1}^{m} \frac{-q_i z_i^k}{k}. \tag{2.3}$$

*Furthermore, for any $p \geq 1$,*

$$\left| \phi(z) - Q \log(z) - \sum_{k=1}^{p} \frac{a_k}{z^k} \right| \leq \alpha \left| \frac{r}{z} \right|^{p+1} \leq \left( \frac{A}{c-1} \right) \left( \frac{1}{c} \right)^p, \tag{2.4}$$

*where*

$$c = \left| \frac{z}{r} \right|, \quad A = \sum_{i=1}^{m} |q_i|, \quad \text{and} \quad \alpha = \frac{A}{1 - |r/z|}. \tag{2.5}$$

*Proof.* The form of the multipole expansion (2.2) is an immediate consequence of the preceding lemma and the fact that $\phi(z) = \sum_{i=1}^{m} \phi_{z_i}(z)$. To obtain the error bound (2.4), observe that

$$\left| \phi(z) - Q \log(z) - \sum_{k=1}^{p} \frac{a_k}{z^k} \right| = \left| \sum_{k=p+1}^{\infty} \frac{a_k}{z^k} \right|.$$

Substituting for $a_k$ the expression in (2.3), we have

$$\left| \sum_{k=p+1}^{\infty} \frac{a_k}{z^k} \right| \leq A \sum_{k=p+1}^{\infty} \frac{r^k}{k\,|z|^k} \leq A \sum_{k=p+1}^{\infty} \left| \frac{r}{z} \right|^k = \alpha \left| \frac{r}{z} \right|^{p+1} = \left( \frac{A}{c-1} \right) \left( \frac{1}{c} \right)^p.$$

In particular, if $c \geq 2$, then

$$\left| \phi(z) - Q \log(z) - \sum_{k=1}^{p} \frac{a_k}{z^k} \right| \leq A \left( \frac{1}{2} \right)^p. \quad \blacksquare \tag{2.6}$$

Finally, we demonstrate, with a simple example, how multipole expansions can be used to speed up calculations with potential fields. Suppose that charges of strengths $q_1$, $q_2$, ..., $q_m$ are located at the points $x_1, x_2, ..., x_m \in \mathbb{C}$ and that $\{y_1, y_2, ..., y_n\}$ is another set of points in $\mathbb{C}$ (Fig. 1). We say that the sets $\{x_i\}$ and $\{y_i\}$ are *well separated* if there exist points $x_0, y_0 \in \mathbb{C}$ and a real $r > 0$ such that

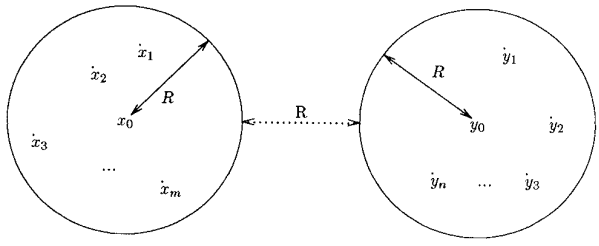


**FIG. 1.** Well-separated sets in the plane.

$$\begin{aligned} &|x_i - x_0| < r \qquad \text{for all } i = 1, ..., m, \\ &|y_j - y_0| < r \qquad \text{for all } j = 1, ..., n, \\ &|x_0 - y_0| > 3r. \end{aligned}$$

In order to obtain the potential (or force) at the points $\{y_j\}$ due to the charges at the points $\{x_i\}$ directly, we could compute

$$\sum_{i=1}^{m} \phi_{x_i}(y_j) \quad \text{for all} \quad j = 1, ..., n. \tag{2.7}$$

This clearly requires order $nm$ work (evaluating $m$ fields at $n$ points). Now suppose that we first compute the coefficients of a $p$-term multipole expansion of the potential due to the charges $q_1, q_2, ..., q_m$ about $x_0$, using Theorem 2.1. This requires a number of operations proportional to $mp$. Evaluating the resulting multipole expansion at all points $y_j$ requires order $np$ work, and the total amount of computation is of the order $O(mp + np)$. Moreover, by (2.6),

$$\left| \sum_{i=1}^{m} \phi_{x_i}(y_j) - Q \log(y_j - x_0) - \sum_{k=1}^{p} \frac{a_k}{|y_j - x_0|^k} \right| \leq A \left( \frac{1}{2} \right)^p,$$

and in order to obtain a relative precision $\varepsilon$ (with respect to the total charge), $p$ must be of the order $-\log_2(\varepsilon)$. Once the precision is specified, the amount of computation has been reduced to

$$O(m) + O(n),$$

which is significantly smaller than $nm$ for large $n$ and $m$.

### 2.1. *Translation Operators and Error Bounds*

The following three lemmas constitute the principal analytical tool of this paper, allowing us to manipulate multipole expansions in the manner required by the fast algorithm. Lemma 2.3 provides a formula for shifting the

center of a multipole expansion, Lemma 2.4 describes how to convert such an expansion into a local (Taylor) expansion in a circular region of analyticity, and Lemma 2.5 furnishes a mechanism for shifting the center of a Taylor expansion within a region of analyticity. We also derive error bounds associated with these translation operators which allow us to carry out numerical computations to any specified accuracy.

LEMMA 2.3. *Suppose that*

$$\phi(z) = a_0 \log(z - z_0) + \sum_{k=1}^{\infty} \frac{a_k}{(z - z_0)^k} \tag{2.8}$$

*is a multipole expansion of the potential due to a set of m charges of strengths $q_1, q_2, ..., q_m$, 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,*

$$\phi(z) = a_0 \log(z) + \sum_{l=1}^{\infty} \frac{b_l}{z^l}, \tag{2.9}$$

*where*

$$b_l = \left( \sum_{k=1}^{l} a_k z_0^{l-k} \binom{l-1}{k-1} \right) - \frac{a_0 z_0^l}{0}, \tag{2.10}$$

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

$$\left| \phi(z) - a_0 \log(z) - \sum_{l=1}^{p} \frac{b_l}{z^l} \right| \leq \left( A \Big/ \left( 1 - \left| \frac{|z_0| + R}{z} \right| \right) \right) \left| \frac{|z_0| + R}{z} \right|^{p+1} \tag{2.11}$$

*with A defined in* (2.5).

*Proof.* The coefficients of the shifted expansion (2.9) are obtained by expanding the expression (2.8) into a Taylor series with respect to $z_0$. For the error bound (2.11), observe that the terms $\{b_l\}$ are the coefficients of the (unique) multipole expansion about the origin of those charges contained in the circle $D$, and Theorem 2.1 applies immediately with $r$ replaced by $|z_0| + R$. ∎

*Remark.* Once the values $\{a_0, a_1, ..., a_p\}$ in the expansion (2.8) about $z_0$ are computed, we can obtain $\{b_1, ..., b_p\}$ exactly by (2.10). In other words, we may shift the center of a truncated multipole expansion without any loss of precision.

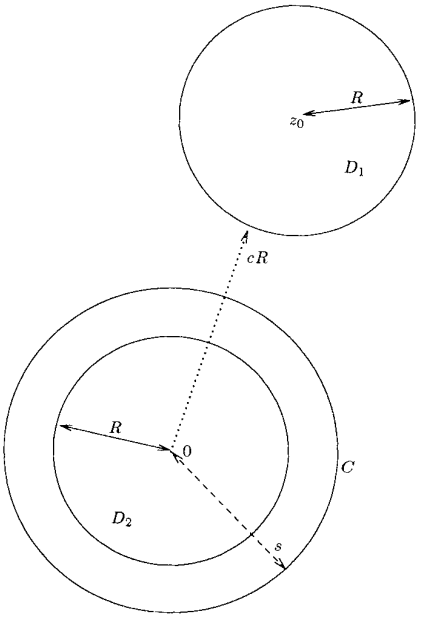


**FIG. 2.** Source charges $q_1, q_2, ..., q_1$ are contained in the circle $D_1$. The corresponding multipole expansion about $z_0$ converges inside $D_2$. $C$ is a circle of radius $s$, with $s > R$.

LEMMA 2.4. *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 $c > 1$ (Fig. 2). Then the corresponding multipole expansion* (2.8) *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,*

$$\phi(z) = \sum_{l=0}^{\infty} b_l \cdot z^l, \tag{2.12}$$

*where*

$$b_0 = \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} (-1)^k + a_0 \log(-z_0) \tag{2.13}$$

*and*

$$b_l = \left( \frac{1}{z_0^l} \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} \binom{l+k-1}{k-1} (-1)^k \right) - \frac{a_0}{l \cdot z_0^l} \quad \text{for } l \geq 1. \tag{2.14}$$

*Furthermore, for any $p \geq \max(2, 2c/(c-1))$, an error bound for the truncated series is given by*

$$\left| \phi(z) - \sum_{l=0}^{p} b_l \cdot z^l \right| < \frac{A(4e(p+c)(c+1) + c^2)}{c(c-1)} \left( \frac{1}{c} \right)^{p+1}, \tag{2.15}$$

*where A is defined in* (2.5) *and e is the base of natural logarithms.*

*Proof.* We obtain the coefficients of the local expansion (2.12) from Maclaurin's theorem applied to the multiple expansion (2.8). To derive the error bound (2.15), we let $\gamma_0 = a_0 \log(-z_0)$, $\gamma_l = -(a_0/l \cdot z_0^l$ for $l \geq 1$, and $\beta_l = b_l - \gamma_l$ for $l \geq 0$. Then

$$\left| \phi(z) - \sum_{l=0}^{p} b_l \cdot z^l \right| = \left| \sum_{l=p+1}^{\infty} b_l \cdot z^l \right| \leq S_1 + S_2 \qquad (2.16)$$

with

$$S_1 = \left| \sum_{l=p+1}^{\infty} \gamma_l \cdot z^l \right|, \quad S_2 = \left| \sum_{l=p+1}^{\infty} \beta_l \cdot z^l \right|.$$

A bound for $S_1$ is easily found by observing that

$$S_1 = \left| \sum_{l=p+1}^{\infty} \gamma_l z^l \right| \leq |a_0| \sum_{l=p+1}^{\infty} \frac{z^l}{l \cdot z_0^l} \leq A \sum_{l=p+1}^{\infty} \frac{z^l}{l \cdot z_0^l}$$
$$\leq A \sum_{l=p+1}^{\infty} \left( \frac{1}{c+1} \right)^l < A \sum_{l=p+1}^{\infty} \left( \frac{1}{c} \right)^l = \left( \frac{A}{c-1} \right) \left( \frac{1}{c} \right)^p.$$

To obtain a bound for $S_2$, let $C$ be a circle of radius $s$, where $s = cR((p-1)/p)$ (Fig. 2). Note first that for any $p \geq 2c/(c-1)$,

$$R < \frac{cR + R}{2} < s < cR.$$

Defining the function $\phi_1 : \mathbb{C} \backslash D_1 \to \mathbb{C}$ by the expression

$$\phi_1(z) = \phi(z) - a_0 \cdot \log(z - z_0),$$

and using Taylor's theorem for complex analytic functions (see [6, p. 190]), we obtain

$$S_2 = \left| \phi_1(z) - \sum_{l=0}^{p} \beta_l z^l \right| = \left| \sum_{l=p+1}^{\infty} \beta_l z^l \right|$$
$$\leq M \Big/ \left( 1 - \frac{|z|}{s} \right) \left( \frac{|z|}{s} \right)^{p+1},$$

where

$$M = \max_C |\phi_1(t)|.$$

Obviously, for any $t$ lying on $C$,

$$|\phi_1(t)| \leq \sum_{k=1}^{\infty} \left| \frac{a_k}{(t - z_0)^k} \right|,$$

and it is easy to see that

$$|a_k| \leq AR^k, \quad |t - z_0| \geq R + cR - s = R + cR/p.$$

After some algebraic manipulation, we have

$$M \leq A \left( \frac{pR + cR}{cR} \right), \quad 1 - \frac{|z|}{s} \geq \frac{cR - R}{cR + R}.$$

Observing that for any positive integer $n$ and any integer $p \geq 2$,

$$\left( 1 + \frac{1}{n} \right)^n \leq e, \quad \left( 1 + \frac{1}{p-1} \right)^2 \leq 4,$$

we obtain

$$S_2 \leq \frac{A(pR + cR)(cR + R)}{cR(cR - R)} \left( \frac{|z|}{cR} \right)^{p+1} \left( \frac{p}{p-1} \right)^{p+1}$$
$$\leq \frac{A(p + c)(c + 1)}{c(c-1)} \left( \frac{1}{c} \right)^{p+1} \left( 1 + \frac{1}{p-1} \right)^{p-1} \left( 1 + \frac{1}{p-1} \right)^2$$
$$\leq \frac{4Ae(p + c)(c + 1)}{c(c-1)} \left( \frac{1}{c} \right)^{p+1}.$$

Adding the last expression to the error bound for $S_1$ completes the proof. ∎

The following lemma is an immediate consequence of Maclaurin's theorem. It describes an exact translation operation with a finite number of terms, and no error bound is needed.

LEMMA 2.5. *For any complex* $z_0$, $z$, *and* $\{a_k\}$, $k = 0, 1, 2, \ldots, n$,

$$\sum_{k=0}^{n} a_k (z - z_0)^k = \sum_{l=0}^{n} \left( \sum_{k=l}^{n} a_k \binom{k}{l} (-z_0)^{k-l} \right) z^l. \qquad (2.17)$$

## 3. THE FAST MULTIPOLE ALGORITHM

In this section, we present an algorithm for the rapid evaluation of the potentials and/or electrostatic fields due to distributions of charges. The central strategy used is that of clustering particles at various spatial lengths and computing interactions with other clusters which are suffi-

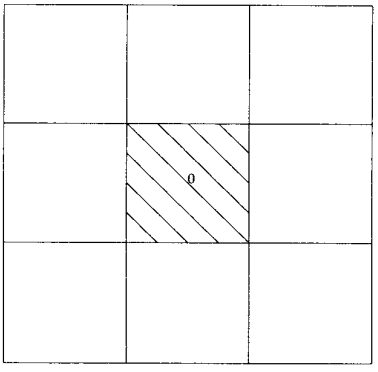


**FIG. 3.** The computational box (shaded) and its nearest periodic images. The box is centered at the origin "0" and has area one.

ciently far away by means of multipole expansions. Interactions with particles which are nearby are handled directly.

To be more specific, let us consider the geometry of the computational box, depicted in Fig. 3. It is a square with sides of length one, centered about the origin of the coordinate system, and is assumed to contain all $N$ particles of the system under consideration. The eight nearest neighbor boxes are also shown and will be needed in the next section when considering various boundary conditions. First, we will describe the method for free-space problems, where the boundary can be ignored and the only interactions to be accounted for involve particles within the computational box itself.

Fixing a precision $\varepsilon$, we choose $p \approx \log_2(\varepsilon)$ and specify that no interactions be computed for clusters of particles which are not *well separated.* This is precisely the condition needed for the error bounds (2.4), (2.11), and (2.15) to apply with $c = 2$, the truncation error to be bounded by $2^{-p}$, and the desired precision to be achieved. In order to impose such a condition, we introduce a hierarchy of meshes which refine the computational box into smaller and smaller regions (Fig. 4). Mesh level 0 is equivalent to the entire box, while mesh level $l + 1$ is obtained from level $l$ by subdivision of each region into four equal parts. The number of distinct boxes at mesh level $l$ is equal to $4^l$. A tree structure is imposed on this mesh hierarchy, so that if *ibox* is a fixed box at level $l$, the four boxes at level $l + 1$ obtained by subdivision of *ibox* are considered its children.

Other notation used in the description of the algorithm includes

$\Phi_{li}$ the $p$-term multipole expansion (about the box center) of the potential field created by the particles contained inside box $i$ at level $l$,

$\Psi_{l,i}$ the $p$-term expansion about the center of box $i$ at level $l$, describing the potential field due to all particles outside the box and its nearest neighbors.

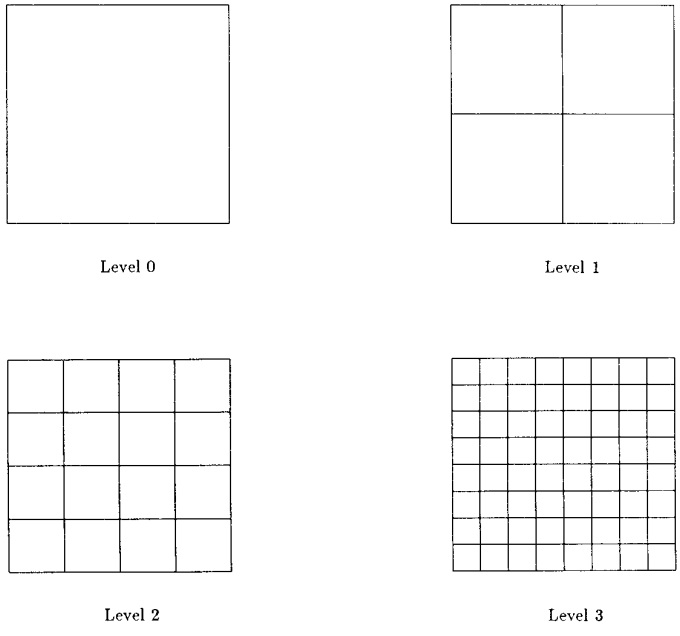


**FIG. 4.** The computational box and three levels of refinement.

$\tilde{\Psi}_{l,i}$ the $p$-term local expansion about the center of box $i$ at level $l$, describing the potential field due to all particles outside $i$'s *parent* box and the *parent* box's nearest neighbors.

*Interaction list* for box $i$ at level $l$, it is the set of boxes which are children of the nearest neighbors of $i$'s parent and which are well separated from box $i$ (Fig. 5).

Suppose now that at level $l - 1$, the local expansion $\Psi_{l-1,i}$ has been obtained for all boxes. Then, by using Lemma 2.5

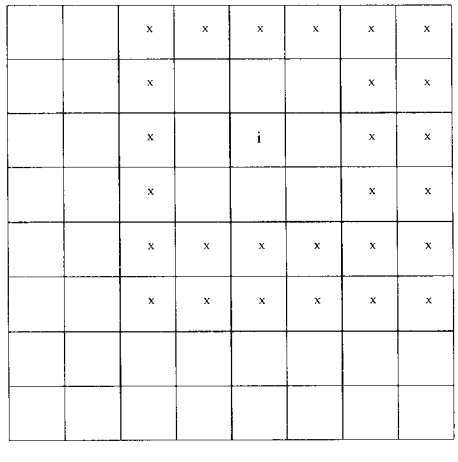


**FIG. 5.** Interaction list for box $i$. Thick lines correspond to mesh level 2 and thin lines to level 3. Boxes marked with an "x" are *well separated* from box $i$ and contained within the nearest neighbors of box $i$'s parent.

to shift (for all $i$) the expansion $\Psi_{l-1,i}$ to each of box $i$'s children, we have, for each box $j$ at level $l$, a local representation of the potential due to all particles outside of $j$'s parent's neighbors, namely $\tilde{\Psi}_{l,j}$. The interaction list is, therefore, precisely that set of boxes whose contribution to the potential must be added to $\tilde{\Psi}_{l,j}$ in order to create $\Psi_{l,j}$. This is done by using Lemma 2.4 to convert the multipole expansions of these interaction boxes to local expansions about the current box center and adding them to the expansion obtained from the parent. Note also that with free-space boundary conditions, $\Psi_{0,i}$ and $\Psi_{1,i}$ are equal to zero since there are no well-separated boxes to consider, and we can begin forming local expansions at level 2. The following is a formal description of the algorithm.

ALGORITHM.

**Initialization**

Choose a level of refinement $n \approx \log_4 N$, a precision $\varepsilon$, and set $p \approx \log_2(\varepsilon)$.

**Upward Pass**

**Step 1**

**Comment** [From multipole expansions of potential field due to particles in each box about the box center at the finest mesh level.]

**do** $ibox = 1, ..., 4^n$

Form a $p$-term multipole expansion $\Phi_{n,ibox}$, by using Theorem 2.1.

**enddo**

**Step 2**

**Comment** [Form multipole expansions about the centers of all boxes at all coarser mesh levels, each expansion representing the potential field due to all particles contained in one box.]

**do** $l = n - 1, ..., 0$

**do** $ibox = 1, ..., 4^l$

Form a $p$-term multipole expansion $\Phi_{l,ibox}$, by using Lemma 2.3 to shift the center of each child box's expansion to the current box center and adding them together.

**enddo**

**enddo**

**Downward Pass**

**Comment** [In the downward pass, interactions are consistently computed at the coarsest possible level. For a given box, this is accomplished by including interactions with those boxes which are well separated and whose interactions have not been accounted for at the parent's level.]

**Step 3**

**Comment** [Form a local expansion about the center of each box at each mesh level $l \leq n - 1$. This local expansion describes the field due to all particles in the system that are not contained in the current box or its nearest neighbors. Once the local expansion is obtained for a given box, it is shifted, in the second inner loop to the centers of the box's children, forming the initial expansion for the boxes at the next level.]

**Set** $\tilde{\Psi}_{1,1} = \tilde{\Psi}_{1,2} = \tilde{\Psi}_{1,3} = \tilde{\Psi}_{1,4} = (0, 0, ..., 0)$

**do** $l = 1, ..., n - 1$

**do** $ibox = 1, ..., 4^l$

Form $\Psi_{l,ibox}$ by using Lemma 2.4 to convert the multipole expansion $\Phi_{l,j}$ of each box $j$ in *interaction list* of box *ibox* to a local expansion about the center of box *ibox*, adding these local expansions together, and adding the result to $\tilde{\Psi}_{l,ibox}$.

**enddo**

**do** $ibox = 1, ..., 4^l$

Form the expansion $\tilde{\Psi}_{l+1,j}$ for *ibox*'s children by using Lemma 2.5 to expand $\Psi_{l,box}$ about the children's box centers.

**enddo**

**enddo**

**Step 4**

**Comment** [Compute interactions at finest mesh level.]

**do** $ibox = 1, ..., 4^n$

Form $\Psi_{l,box}$ by using Lemma 2.4 to convert the multipole expansion $\Phi_{l,j}$ of each box $j$ in *interaction list* of box *ibox* to a local expansion about the center of box *ibox*, adding these local expansions together, and adding the result to $\tilde{\Psi}_{l,ibox}$.

**enddo**

**Comment** [Local expansions at finest mesh level are now available. They can be used to generate the potential or force due to all particles outside the nearest neighbor boxes at finest mesh level.]

**Step 5**

**Comment** [Evaluate local expansions at particle positions.]

**do** $ibox = 1, ..., 4^n$

For every particle $p_j$ located at the point $z_j$ in box *ibox*, evaluate $\Phi_{n,ibox}(z_j)$.

**enddo**

**Step 6**

**Comment** [Compute potential (or force) due to nearest neighbors directly.]

**do** $ibox = 1, ..., 4^n$

For every particle $p_j$ in box *ibox*, compute interactions with all other particles within the box and its nearest neighbors.

**enddo**

**Step 7**

**do** *ibox* = 1, ..., $4^n$

For every particle in box *ibox*, add direct and far-field terms together.

**enddo**

*Remark.* Each local expansion is described by the coefficients of a $p$-term polynomial. Direct evaluation of this polynomial at a point yields the potential. But, by Lemma 2.1, the force is immediately obtained from the derivative which is available analytically. There is no need for numerical differentiation. Furthermore, due to the analyticity of $\Phi'$, there exist error bounds for the force of exactly the same form as (2.4), (2.11), and (2.15).

A brief analysis of the algorithmic complexity is given below.

| Step | Operation count | Explanation |
|---|---|---|
| 1 | order $Np$ | Each particle contributes to one expansion at the finest level. |
| 2 | order $Np^2$ | At the $l$th level, $4^l$ shifts involving order $p^2$ work per shift must be performed. |
| 3 | order $\leq 28Np^2$ | There are at most 27 entries in the interaction list for each box at each level. An extra order $Np^2$ work is required for the second loop. |
| 4 | order $\leq 27Np^2$ | Again, there are at most 27 entries in the interaction list for each box and $\approx N$ boxes. |
| 5 | order $\leq 27Np^2$ | One $p$-term expansion is evaluated for each particle. |
| 6 | order $\frac{9}{2}Nk_n$ | Let $k_n$ be a bound on the number of particles per box at the finest mesh level. Interactions must be computed within the box and its eight nearest neighbors, but using Newton's third law, we need only compute half of the pairwise interactions. |
| 7 | order $N$ | Adding two terms for each particle. |

The estimate for the running time is therefore

$$N(-2a \log_2(\varepsilon) + 56b(\log_2(\varepsilon))^2 + 4.5dk_n + e),$$

with the constants $a$, $b$, $c$, $d$, and $e$ determined by the computer system, language, implementation, etc.

In addition to the asymptotic time complexity, asymptotic storage requirements are an important characteristic of a numerical procedure. The algorithm requires that $\Phi_{l,j}$ and $\Psi_{l,j}$ be stored, as well as the locations of the particles, their charges, and the results of the calculations (the potentials and/or electric fields). Since every box at every level has a pair of $p$-term expansions, $\Phi$ and $\Psi$, associated with it, and the lengths of all other storage arrays are proportional to $N$, it is easy to see that the asymptotic storage requirements of the algorithm are of the form

$$(\alpha + \beta p) \cdot N$$

or

$$(\alpha - \beta \log_2(\varepsilon)) \cdot N,$$

with the coefficients $\alpha$ and $\beta$ determined, as above, by the computer system, language, implementation, etc. In our numerical experiments, the actual storage requirements were of the order

$$(25 - \log_2(\varepsilon)) \cdot N$$

single precision words.

*Remark.* It is clear that the operation count for Step 6 assumes a reasonably homogeneous distribution of particles. If the distribution were highly nonhomogeneous, then we would need to refine only those portions of space where the number of particles is large. Although its description is more involved, an adaptive version retains both the accuracy and the computational speed of the algorithm (see [3]).

## 4. BOUNDARY CONDITIONS

A variety of boundary conditions are used in particle simulations, including periodic boundary conditions, homogeneous Dirichlet or Neumann conditions, and several types of mixed conditions. The periodic case will be treated first in some detail. We then turn to the imposition of Dirichlet conditions and end with a brief discussion of the other cases.

### 4.1. *Periodic Boundary Conditions*

We begin by reconsidering the computational domain depicted in Fig. 5. At the end of the upward pass of the algorithm, we have a net multipole expansion

$$\Phi_{0,1}(z) = \sum_{k=1}^{p} \frac{a_k}{z^k} \tag{4.1}$$

for the entire computational box. This is then the expansion for each of the periodic images of the box with respect to its own center. All of these images except for the ones depicted in Fig. 3 are *well separated* from the computational box itself, and their induced fields are accurately representable by a $p$-term local expansion, where, as before, $p \approx -\log_2(\varepsilon)$ is the number of terms needed to achieve a relative precision $\varepsilon$. We assume that the periodic particle model

has no net charge and, therefore, that the local representation given by Lemma 2.4 can be written as

$$\Psi_{0,1} = \sum_{m=1}^{p} b_m \cdot z^m \tag{4.2}$$

with

$$b_m = \frac{1}{z_0^m} \sum_{k=1}^{p} \frac{a_k}{z_0^k} \binom{m+k-1}{k-1} (-1)^k \quad \text{with } m = 0, 1, ..., p, \tag{4.3}$$

with $z_0$ the center of the image box under consideration.

*Remark.* In certain problems (e.g., cosmology), the computational box obviously cannot satisfy the condition of no net charge (mass). This condition is necessary for the potential to be well defined, since the logarithmic term becomes unbounded as $z_0 \to \infty$. Force calculations, however, may still be carried out. Indeed, using the notation of the algorithm, $\Phi_{l,i}$, $\Psi_{l,i}$, $\tilde{\Psi}_{l,i}$ are expansions of analytic functions representing the potential, so that their derivatives are also analytic functions (with the same regions of analyticity). Moreover, it is clear from Theorem 2.1 that the derivatives $\Phi'_{l,i}$ are described by pure inverse power series. Therefore, the identical formal structure of the algorithm can, due to Lemma 2.1, be used to evaluate force fields everywhere, bypassing the difficulty introduced by the logarithmic term. The only change required is that the initial expansions computed be the derivatives of the multipole expansions and not the multipole expansions themselves.

Note now that well-separated images of the computational cell are boxes whose centers $z_0$ have integer real and imaginary parts, with $\mathrm{Re}(z_0) \geq 2$ or $\mathrm{Im}(z_0) \geq 2$. Let $S$ be the set of such centers. To account for the field due to all well-separated images, we form the coefficients for the local representation by adding the local shifted expansions of the form (4.3) for all $z_0 \in S$ to obtain

$$b_m^{\text{total}} = \sum_{k=1}^{p} a_k \binom{m+k-1}{k-1} (-1)^k \left( \sum_S \frac{1}{z_0^{m+k}} \right). \tag{4.4}$$

The summation over $S$ for each inverse power of $z_0$ can be precomputed and stored. For $(m + k) > 2$, the series is absolutely convergent. However, for $(m + k) \leq 2$, the series is not absolutely convergent, and the computed value depends on the order of addition. Choosing a reasonable value for the sum of the series requires careful consideration of the physical model.

Suppose first that the only particle in the simulation is a charge of unit strength located at the origin. Then the periodic model corresponds to a uniform lattice of charges, and Newton's third law requires that the net force on each particle be zero. But the net force on the particle at the origin corresponds to the summation over $S$ of $1/z_0$, so that we set

$$\sum_S \frac{1}{z_0} = 0.$$

To determine a value for the second term,

$$\sum_S \frac{1}{z_0^2},$$

suppose that the only particle in the simulation is a dipole of strength one, oriented along the $x$-axis and located at the origin. Then the periodic model is again a uniform lattice and the difference in potential between the equivalent sites $(-\frac{1}{2}, 0)$ and $(\frac{1}{2}, 0)$ must be zero; i.e.,

$$\Phi_{(1/2,0)} - \Phi_{(-1/2,0)} \equiv \delta\Phi = 0. \tag{4.5}$$

The contribution to the potential difference, $\delta\Phi$, of a single dipole located at $z_0$ is

$$\frac{1}{z_0 - 1/2} - \frac{1}{z_0 + 1/2} = \frac{1}{z_0^2 - 1/4}.$$

Thus, we find that the potential difference due to the original dipole located at the origin is $-4$. For an image dipole located at $z_0$, with $|z_0| \geq 1$, we can expand the contribution to $\delta\Phi$ as

$$\frac{1}{z_0^2 - 1/4} = \frac{1}{z_0^2} + \frac{1}{4z_0^4 - z_0^2}.$$

Now let $S'$ be the set of the centers of all image boxes. That is, $S'$ is the set of all points $z_0$ with integer real and imaginary parts, excluding the origin. Then

$$\delta\Phi = -4 + \sum_{S'} \frac{1}{z_0^2} + \sum_{S'} \frac{1}{4z_0^4 - z_0^2}.$$

A somewhat involved calculation shows that

$$\sum_{S'} \frac{1}{4z_0^4 - z_0^2} = 4 - \pi.$$

Therefore, to satisfy (4.5), we set

$$\sum_{S'} \frac{1}{z_0^2} = \pi.$$

Now

$$\sum_{S'} \frac{1}{z_0^2} = \sum_{S} \frac{1}{z_0^2} \sum_{S' \setminus S} \frac{1}{z_0^2},$$

and the sum $\Sigma_{S' \setminus S}\,(1/z_0^2)$ is easily evaluated and found to be equal to zero. Therefore, we have

$$\sum_{S} \frac{1}{z_0^2} = \pi,$$

and the summation over $S$ for every inverse power of $z_0$ is defined.

The procedure of converting the multipole expansion of the whole computational cell $\Phi_{0,1}$ into a local expansion $\Psi_{0,1}$ which describes the potential field due to all well-separated images can be written, in the notation of the algorithm, as

$$\Psi_{0,1} = T \cdot \Phi_{0,1},$$

where $T$ is a constant $p$ by $p$ matrix whose entries are defined by the formula

$$T_{m,k} = \binom{m+k-1}{k-1} (-1)^k \left( \sum_{S} \frac{1}{z_0^{m+k}} \right).$$

This can be viewed as the first step in the downward pass of the algorithm for periodic boundary conditions. At this point, we have accounted for all interactions excluding the ones within the immediate neighbors of the computational box as depicted in Fig. 3. But the expansions $\Phi_{l,i}$ for boxes inside the computational cell are also the expansions of the corresponding boxes inside the nearest neighbor images of the computational cell. By adding to the interaction list the appropriate boxes, we maintain the formal structure of the algorithm and the associated computational complexity.

### 4.2. *Dirichlet Boundary Conditions*

We turn now to the imposition of homogeneous Dirichlet boundary conditions, namely

$$\Phi(x, y) = 0 \quad \text{for } (x, y) \in \partial D,$$

where $\partial D$ is the boundary of the computational domain. Analytically speaking, this can be accomplished by the method of images, described in detail below. In general terms, we consider the potential field to be composed of two parts; that is,

$$\Phi = \Phi_{\text{sources}} + \Phi_{\text{images}},$$

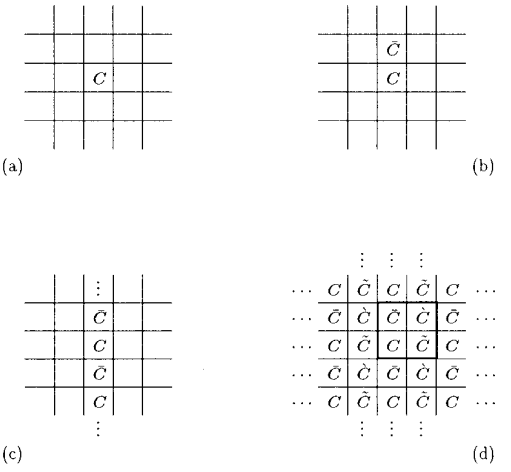


**FIG. 6.** The computational cell centered at the origin is represented by $C$. The cell $\overline{C}$ is the image of cell $C$ reflected across the top boundary, with corresponding particles assigned charges of opposite sign. The cell $\tilde{C}$ is the image of cell $C$ reflected across the left boundary, again with corresponding particles assigned charges of opposite sign. The cell $\dot{C}$ is the image of cell $C$ reflected through the origin, with corresponding particles assigned charges of the same sign. Successive reflections across the four boundaries of the computational cell yield an infinite expansion of image boxes as indicated in (d).

where $\Phi_{\text{sources}}$ is the field due to the particles inside the computational cell and $\Phi_{\text{images}}$ is the field due to selected image charges located outside the computational cell. The image charge positions and strengths are chosen so that

$$\Phi_{\text{sources}}(x, y) = -\Phi_{\text{images}}(x, y) \quad \text{for } (x, y) \in \partial D.$$

For the computational domain we are considering, appropriate locations for the image charges can be determined by an iterative process, illustrated in Fig. 6. We first reflect each particle $p_i$ of charge strength $\sigma_i$ in the computational cell across the top boundary line and place an image charge of strength $-\sigma_i$ at that location, generating an image box which we denote $\overline{C}$ (Fig. 6b)). The set of image charges is denoted by $V_1$, and the field they induce is called $\Phi_{V_1}$. Adding $\Phi_{V_1}$ to $\Phi_{\text{sources}}$ clearly enforces the desired condition along the top boundary. To impose the boundary condition along the bottom of the computational cell, we must reflect all charges (source and image) currently in the model across the bottom boundary, generating two more image boxes (which are copies of $C$ and $\overline{C}$). The set of all image charges after this second reflection step is denoted by $V_2$. Now, while $\Phi_{\text{sources}} + \Phi_{V_2}$ is equal to zero along the bottom boundary, the resulting field violates the top boundary condition. We therefore reflect again across the top boundary, creating two new image boxes and a new set of image charges $V_3$, such that $\Phi_{\text{sources}} + \Phi_{V_3}$ satisfies the top condition but violates the bottom one. By

iterating in this manner, we generate a sequence of sets of image charges $\{V_i\}$ with

$$V_1 \subset V_2 \subset V_3 \subset \cdots \subset V,$$

where $V = \bigcup_{i=1}^{\infty} V_i$ is the set of charges contained in the infinite array of image boxes depicted in Fig. 6c. It is easy to see that the corresponding sequence of image fields $\{\Phi_{V_i}\}$ converges inside the computational cell and that the potential field $\Phi_{\text{sources}} + \Phi_V$ does satisfy both the top and bottom boundary conditions.

In order to enforce the Dirichlet condition on the remaining two sides, we proceed analogously. First, we reflect all the charges currently in the model (the original sources plus the images in $V$) across the left boundary. This obviously does not affect the top and bottom conditions and enforces the homogeneous boundary condition along the left side of the computational cell. The current set of (all) image charges is now denoted $H_1$. Reflecting across the right boundary creates a new set $H_2$, witht he field $\Phi_{\text{sources}} + \Phi_{H_2}$ satisfying the Dirichlet condition along the right (but not the left) boundary. Repeated reflection across the left and right boundaries of the computational cell yields a sequence $\{H_i\}$ of infinite sets of image charges,

$$H_1 \subset H_2 \subset H_3 \subset \cdots \subset H,$$

where $H = \bigcup_{i=1}^{\infty} H_i$ is the set of charges contained in the two-dimensional family of image boxes depicted in Fig. 6d. It is easy to see that the sequence $\{\Phi_{H_i}\}$ converges inside the computational cell, and we denote its limit by $\Phi_H$. Finally, we observe that $\Phi_{\text{sources}} + \Phi_H = 0$ on the entire boundary $\partial D$.

From a computational point of view, the rate of convergence of the method of images is quite unsatisfactory. In conjunction with our algorithm, however, this method can be turned into an extremely efficient numerical tool. In the terminology previously introduced, all of the image boxes except the nearest neighbors of the computational cell are well separated and their induced fields can be represented by a single local expansion, denoted $\Psi_{0,1}$. Once the coefficients of this local expansion have been computed, we need only account for interactions within the nearest neighbors of the computational cell itself. To do this, as in the periodic case, we simply add the appropriate image boxes to the interaction lists of the boxes inside the computational cell.

Thus, it remains only to calculate $\Psi_{0,1}$. We first observe that the plane of images has a periodic structure with unit "supercell" centered at $(\frac{1}{2}, \frac{1}{2})$, indicated by thick lines in Fig. 6d. But then, by the method developed above for periodic problems, we can obtain an expansion about the point $(\frac{1}{2}, \frac{1}{2})$ which accounts for all interactions beyond the nearest neighbors of the suercell. This expansion can be converted, by using Lemma 2.5, into an expansion about the origin (the center of the computational cell), which we call $\tilde{\Psi}_{0,1}$. It remains to account for the well-separated boxes which are contained inside the supercell's nearest neighbors. There are exactly 27 of these boxes, and their multipole expansions can be shifted (by using Lemma 2.4) to local expansions about the origin which are then added to $\tilde{\Psi}_{0,1}$ to finally form $\Psi_{0,1}$.

### 4.3. *Other Boundary Conditions*

While in certain applications, periodic or Dirichlet boundary conditions are called for, in others, Neumann or mixed conditions have to be imposed on the boundary of the computational domain. A typical example of a problem with mixed conditions is the computational cell with Neumann conditions on two opposing sides and Dirichlet conditions on the two others. Other models require periodic boundary conditions on the left and right sides of the computational cell and Dirichlet or Neumann conditions on the top and bottom. The imposition of these conditions is achieved by a procedure essentially identical to the one described above. By reflection and/or periodic extension, one first generates an entire plane of images. The local expansion $\Psi_{0,1}$ is then computed by an appropriate summation over all well-separated image boxes, and the remaining image interactions are handled as above.

## 5. NUMERICAL RESULTS

A computer program has been implemented utilizing the algorithm of this paper and capable of handling free-space problems and problems with periodic, homogeneous Dirichlet or homogeneous Neumann boundary conditions.

For testing purposes, we randomly assigned charged particles to positions in the computational cell (Fig. 7), with charge strengths between 0 and 1, and with the numbers

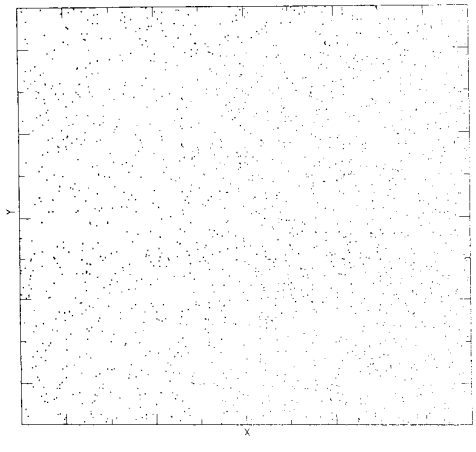

**FIG. 7.** 1600 randomly located charges in the computational cell.

of particles varying from 100 to 12,800. The calculations were performed on a VAX-8600, and the number of terms in the expansions $\Psi_{l,i}$, $\tilde{\Phi}_{l,i}$, $\Phi_{l,i}$ was set to 20, guaranteeing roughly 5-digit accuracy of the result. In each case, we performed the calculation in three ways: (1) via the algorithm of the present paper in single precision arithmetic; (2) directly (via formula (2.7)) in single precision arithmetic; and (3) via formula (2.7) in double precision arithmetic. The first two calculations were used to compare the speed and accuracy of our algorithm to those of the direct method. The direct evaluation of the field in double precision was used as a standard for comparing the relative accuracies of the first two computations. In all cases, the calculation was performed for a periodic model, the periodic boundary condition being imposed by means of the algorithm described in Section 4 of this paper.

The results of these numerical experiments are summarized in Table I. The first column of the table contains the numbers $N$ of particles for which calculations have been performed. The second and third columns contain the CPU times $T_{\text{alg}}$ that were required by the algorithm of the present paper to obtain the fields at all $N$ particles, and the greatest relative error $\delta_{\text{alg}}$ obtained at any of the particles, respectively. Columns 4 and 5 contain the CPU times $T_{\text{dir}}$ that were required by the direct algorithm (2.7) to obtain the fields at all $N$ particles, and the greatest relative error $\delta_{\text{dir}}$ obtained at any one particle, respectively.

*Remark.* For the example involving 12,800 particles, the algorithm of the present paper required about one minute of CPU time (see Table I). However, it was not considered practical to use the direct algorithm to evaluate the field at all 12,800 points, since it would take about 5 h of CPU time, without producing much useful information. Therefore, we used the direct algorithm to evaluate the field at only 100 of the 12,800 particles, both in single and double precision, and used the resulting data to estimate $\delta_{\text{alg}}$ and $\delta_{\text{dir}}$. The value for $T_{\text{dir}}$ in this case was estimated by scaling.

The following observations can be made from Table I:

**TABLE I**

Computational Results

| $N$ | $T_{\text{alg}}$ (s) | $\delta_{\text{alg}}$ | $T_{\text{dir}}$ (s) | $\delta_{\text{dir}}$ |
|---|---|---|---|---|
| 100 | 0.6 | $1.1 \times 10^{-5}$ | 1.1 | $1.9 \times 10^{-5}$ |
| 200 | 1.4 | $4.1 \times 10^{-5}$ | 4.5 | $3.2 \times 10^{-5}$ |
| 400 | 2.0 | $3.6 \times 10^{-5}$ | 18 | $6.6 \times 10^{-5}$ |
| 800 | 3.8 | $4.6 \times 10^{-5}$ | 69 | $7.3 \times 10^{-5}$ |
| 1,600 | 6.6 | $1.4 \times 10^{-5}$ | 272 | $7.0 \times 10^{-5}$ |
| 3,200 | 16.5 | $0.9 \times 10^{-5}$ | 1088 | $3.1 \times 10^{-5}$ |
| 6,400 | 24.7 | $7.2 \times 10^{-5}$ | 4480 | $6.8 \times 10^{-5}$ |
| 12,800 | 60.9 | $3.0 \times 10^{-5}$ | 17920 (est.) | $1.8 \times 10^{-5}$ |

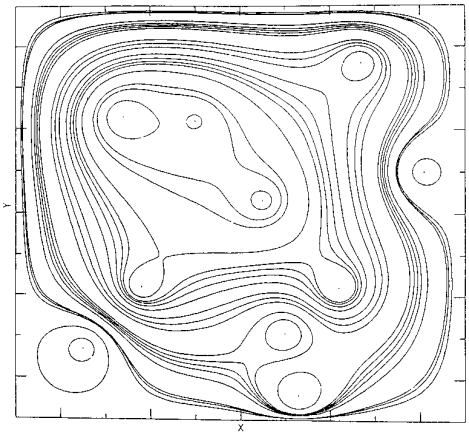

**FIG. 8.** The equipotential lines for the electrostatic field due to 10 randomly located charges in the computational cell, with homogeneous Dirichlet boundary conditions.

1. The accuracy of the results produced by the algorithm is about the same as that predicted by the estimates (2.4), (2.11), and (2.15) for the number of terms we are using in the expansions $\Phi_{l,i}$, $\tilde{\Psi}_{l,i}$, $\Psi_{l,i}$. There is no evidence of accuracy problems due to truncation errors.

2. The calculation time grows linearly with the number of charges in the model, even though its behavior is somewhat erratic.

3. For as few as 1600 particles in the model, the computational effort required by the direct algorithm is roughly 40 times greater than that required by the algorithm of the present paper. For 12,800 particles, the effort is nearly 300 times greater.

Similar calculations have been performed for homogeneous Dirichlet and Neumann boundary conditions, and the observations made above for the periodic model are equally applicable in these cases.

For illustration, the equipotential lines for a box with 10 randomly distributed particles and Dirichlet boundary conditions are shown in Fig. 8. The entire calculation required 15 s of CPU time; about half the time was spent evaluating the field at more than 10,000 points, while the rest was used up by the plotting routine.

## 6. CONCLUSIONS

An algorithm has been constructed for the rapid evaluation of potential fields generated by ensembles of particles of the type encountered in plasma physics, molecular dynamics, fluid dynamics (the vortex method), and celestial mechanics. The algorithm is applicable both in the context of dynamical simulations and Monte Carlo simulations, provided that the fields to be evaluated are Coulombic in nature. The asymptotic CPU time estimate for the algo-

rithm of the present paper is of the order $O(N)$, where $N$ is the number of particles in the simulation, and the numerical examples presented in Section 5 indicate that even very large-scale problems result in acceptable CPU time requirements. In the present paper, a two-dimensional version of the algorithm is described. Generalizing this result to three dimensions is fairly straightforward and will be reported at a later date.

## ACKNOWLEDGMENTS

It is the authors' pleasure to thank Professor M. H. Schultz for drawing their attention to the subject of this paper and for his continuing interest and support.

## REFERENCES

1. C. R. Anderson, *J. Comput. Phys.* **62,** 111 (1986).
2. A. W. Appel, *SIAM J. Sci. Stat. Comput.* **6,** 85 (1985).
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. A. J. Chorin, *J. Fluid. Mech.* **57,** 785 (1973).
5. R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (McGraw–Hill, New York, 1981).
6. G. Polya and G. Latta, *Complex Variables* (Wiley, New York, 1974).
7. V. Rokhlin, *J. Comput. Phys.* **60,** 187 (1985).