SIAM J. SCI. STAT. COMPUT.
Vol. 9, No. 4, July 1988

004

A FAST ADAPTIVE MULTIPOLE ALGORITHM FOR PARTICLE SIMULATIONS*

J. CARRIER†, L. GREENGARD‡, AND V. ROKHLIN‡

Abstract. This paper describes an algorithm for the rapid evaluation of the potential and force fields in systems involving large numbers of particles whose interactions are described by Coulomb's law. Unlike previously published schemes, the algorithm of this paper has an asymptotic CPU time estimate of $O(N)$, where N is the number of particles in the simulation, and does not depend on the statistics of the distribution for its efficient performance. The numerical examples we present indicate that it should be an algorithm of choice in many situations of practical interest.

Key words. N-body problem, plasma physics, molecular dynamics, vortex method, potential theory

AMS(MOS) subject classifications. 65C20, 65D99, 77F05, 82A71, 70F10, 70F15

1. Introduction. The evaluation of Coulombic and gravitational interactions in large-scale ensembles of particles is an integral part of the numerical simulation of a large number of physical processes. Typical examples include celestial mechanics, plasma simulations, the vortex method in fluid dynamics, and the solution of the Laplace equation via potential theory (see [1]-[3], [8], [10]). In such cases, the potential has the form

$$\Phi = \Phi_{\text{external}} + \Phi_{\text{local}} + \Phi_{\text{far}}, \tag{1}$$

where Φ_{local} is a rapidly decaying function of distance (such as the Van der Waals potential in chemical physics), Φ_{external} is a function which is independent of the number and relative positions of the particles (such as an external gravitational field) and Φ_{far} is Coulombic or gravitational.

In the numerical evaluation of fields of the form (1), the cost of computing the terms Φ_{external} and Φ_{local} is of the order $O(N)$, where N is the number of particles in the ensemble. Indeed, Φ_{external} is evaluated separately for each particle, and Φ_{local} decays rapidly, involving the interactions of each particle with a small number of nearest neighbors. Unfortunately, evaluation of the term Φ_{far}, if done directly, requires order $O(N^2)$ operations, since the Coulombic potential decays slowly, and the interactions between each pair of particles have to be taken into account. In many situations, in order to be of physical interest, the simulation has to involve thousands of particles (or more), making the estimate $O(N^2)$ excessive in some cases, and prohibitive in others.

Several different approaches have been used to reduce the cost of the Coulombic part of the computation. For a detailed discussion of these algorithms, we refer the reader to [7] and to the original papers [1], [2], [8], [10]. Here, we just observe that each of the algorithms [1], [2], [7], [8], [10] imposes strong requirements on the statistics of the charge distribution. In particular, the methods of [1], [7], and [8] require that the distribution be reasonably uniform in a square-shaped region of interest, the algorithm of [10] assumes that the charges are located on a curve in $\mathbb{R}^2$, and the algorithm of [2] works fairly well for highly clustered distributions, but fails for uniform ones.

* Received by the editors February 2, 1987; accepted for publication October 8, 1987.

† Elf Aquitaine and Department of Computer Science, Yale University, New Haven, Connecticut 06520.

‡ Department of Computer Science, Yale University, New Haven, Connecticut 06520. The work of this author was supported in part by the Office of Naval Research under grant N00014-82-K-0184.

In this paper, we introduce an algorithm for the rapid evaluation of the potential and force fields of large-scale ensembles of particles. To evaluate all Coulombic interactions of N particles in $\mathbb{R}^2$, the algorithm requires an amount of work of the order $O(N)$, and this estimate does not depend on the statistics of the distribution.

The procedure described here is an adaptive version of the algorithm of [7]. In the following section, we introduce the analytical apparatus to be used. Section 3 contains a detailed description of the algorithm and its complexity analysis, and in § 4 we present numerical experiments demonstrating the actual performance of the scheme.

Remark 1.1. Given a collection of points $z_1, \cdots, z_n$ in $\mathbb{C}$, the Hilbert matrix associated with the points $\{z_i\}$ is defined as follows:

$$A_{ij} = \frac{1}{z_i - z_j} \quad \text{for } i \neq j, \qquad A_{ii} = 0.$$

It immediately follows from Lemma 2.1 and formula (3) that evaluating the fields of a set of charges of strengths $q_1, \cdots, q_n$ located at the points $z_1, \cdots, z_n$ at these same points is equivalent to applying the associated Hilbert matrix to the vector $(q_1, \cdots, q_n)$. Therefore, the algorithm of the present paper may be viewed as an order $O(n)$ procedure for applying an $n \times n$ Hilbert matrix to an arbitrary vector. Recently, several papers have been published on this subject, also referred to as the Trummer problem (see [6], [4], [5], [9]).

2. Analytical tools. We consider a two-dimensional physical model consisting of a set of particles whose pairwise interactions are described by Coulomb's law. More precisely, suppose that a point charge of unit strength is located at the point $\mathbf{x}_0 = (x_0, y_0) \in \mathbb{R}^2$. Then, for any $\mathbf{x} = (x, y) \in \mathbb{R}^2 \backslash \{\mathbf{x}_0\}$, the potential and electrostatic field due to this charge are described by the expressions

$$\phi_{\mathbf{x}_0}(\mathbf{x}) = -\log(\|\mathbf{x} - \mathbf{x}_0\|) \tag{2}$$

and

$$E_{\mathbf{x}_0}(\mathbf{x}) = \frac{(\mathbf{x} - \mathbf{x}_0)}{\|\mathbf{x} - \mathbf{x}_0\|^2}, \tag{3}$$

respectively.

It is well known that the function $\phi_{\mathbf{x}_0}$ is harmonic in any region not containing $\mathbf{x}_0$, and that for every harmonic function u, there exists an analytic function $w : \mathbb{C} \to \mathbb{C}$ such that u is the real part of w. In particular, we have

$$\phi_{\mathbf{x}_0}(\mathbf{x}) = \operatorname{Re}(-\log(z - z_0)). \tag{4}$$

In the remainder of this paper we shall work with analytic functions in $\mathbb{C}$, making no distinction between a point $(x, y) \in \mathbb{R}^2$ and a point $x + iy \in \mathbb{C}$. Following standard practice, we will refer to the analytic function $\log(z)$ as the potential due to a charge. For more complicated charge distributions, we will use other analytic functions and we will also refer to them as potentials.

Detailed proofs of Lemmas 2.1-2.4 and Theorem 2.1 below can be found in [7].

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

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

where w' is the derivative of w.

The following theorem gives the expression for the multipole expansion of the potential due to a set of charges and an estimate for the remainder of this expansion after k terms.

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

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

where

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

Furthermore, for any $p \geqq 1$,

$$\left| \phi(z) - Q \log(z) - \sum_{k=1}^{p} \frac{a_k}{z^k} \right| \leqq \alpha \left| \frac{r}{z} \right|^{p+1} \leqq \left(\frac{A}{c-1} \right) \left(\frac{1}{c} \right)^p, \tag{7}$$

where

$$c = \left| \frac{z}{r} \right|, \qquad A = \sum_{i=1}^{m} |q_i|, \qquad \alpha = \frac{A}{1 - |r/z|}. \tag{8}$$

We will use a simple example to demonstrate how multipole expansions can be used to speed up calculations with potentials. Suppose that $X = \{x_1, x_2, \cdots, x_m\}$ and $Y = \{y_1, y_2, \cdots, y_n\}$ are two finite sets of points in $\mathbb{C}$. We say that the sets X and Y are well separated (Fig. 1) if there exist two points $x_0, y_0 \in \mathbb{C}$ and a real number $r > 0$ such that

$$|x_j - x_0| < r \quad \text{for all } i = 1, \cdots, m,$$

$$|y_j - y_0| < r \quad \text{for all } i = 1, \cdots, n,$$

$$|x_0 - y_0| > 3r.$$

Suppose now that charges of strengths $\{q_1, q_2, \cdots, q_m\}$ are located at the points $\{x_1, x_2, \cdots, x_m\}$ and that we wish to evaluate the sum

$$\sum_{i=1}^{m} \phi_{x_i}(y_j) \tag{9}$$

for all $j = 1, 2, \cdots, n$. Clearly, this requires order $n \cdot m$ work (evaluating m fields at n points). Now suppose that we first evaluate the coefficients of a p-term multipole

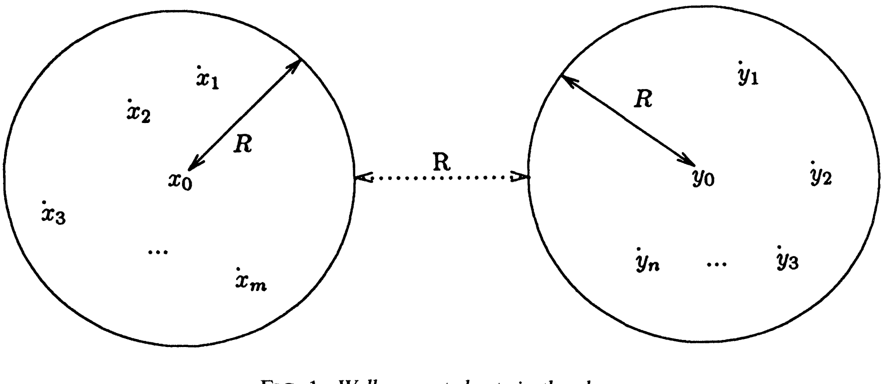

FIG. 1. *Well-separated sets in the plane.*

expansion due to the charges $\{q_1, q_2, \cdots, q_m\}$ about x_0, using Theorem 2.1. This requires a number of operations proportional to $m \cdot p$. Evaluating the resulting multipole expansion at all points y_j requires order $n \cdot p$ work, and the total computational effort is of the order $O(m \cdot p + n \cdot p)$. Furthermore, due to (7),

$$\left| \sum_{i=1}^{m} \phi_{x_i}(y_j) - Q \log (y_j - x_0) - \sum_{k=1}^{p} \frac{a_k}{(y_j - x_0)^k} \right| \leqq A \left(\frac{1}{2}\right)^p, \tag{10}$$

and in order to obtain a relative precision ε, p must be of the order $-\log_2(\varepsilon)$. Once the precision is specified, the amount of computation is reduced to $O(n+m)$, which is significantly smaller than $m \cdot n$ for large m and n.

The following three lemmas describe translation operators for multipole and power series expansions in $\mathbb{R}^2$, and provide error bounds allowing the manipulation of these expansions in the manner required by the algorithm. The first, Lemma 2.2, supplies a mechanism for shifting the center of a multipole expansion.

LEMMA 2.2. *Suppose that*

$$\phi(z) = a_0 \log (z - z_0) + \sum_{k=1}^{\infty} \frac{a_k}{(z - z_0)^k} \tag{11}$$

is a multipole expansion of the potential due to a set of m charges of strengths $q_1, q_2, \cdots, q_m$, 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 centered at the origin,

$$\phi(z) = a_0 \log (z) + \sum_{l=1}^{\infty} \frac{b_l}{z^l}, \tag{12}$$

where

$$b_l = \sum_{k=1}^{l} a_k z_0^{l-k} \binom{l-1}{k-1} - a_0 \left(\frac{z_0^l}{l}\right) \tag{13}$$

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

$$\left| \phi(z) - a_0 \log (z) - \sum_{l=1}^{p} \frac{b_l}{z^l} \right| \leqq \left(A \Big/ 1 - \left| \frac{|z_0| + R}{z} \right| \right) \left| \frac{|z_0| + R}{z} \right|^{p+1}, \tag{14}$$

with A defined in Theorem 2.1.

Lemma 2.3 describes the conversion of a multipole expansion into a local (Taylor) expansion in a circular region of analyticity.

LEMMA 2.3. *Suppose that m charges of strengths $q_1, q_2, \cdots, q_m$, 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* (11) *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:*

$$\phi(z) = \sum_{l=0}^{\infty} b_l \cdot z^l, \tag{15}$$

where

$$b_0 = a_0 \log (-z_0) + \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} (-1)^k \tag{16}$$

and

$$b_l = -\frac{a_0}{l \cdot z_0^l} + \left(\frac{1}{z_0^l} \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} \binom{l+k-1}{k-1} (-1)^k \right) \quad \textit{for } l \geqq 1. \tag{17}$$

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

$$\left|\phi(z) - \sum_{l=0}^{p} b_l \cdot z^l\right| < \frac{A(4e(p+c)(c+1)+c^2)}{c(c-1)}\left(\frac{1}{c}\right)^{p+1}, \tag{18}$$

where A is defined in Theorem 2.1 *and e is the base of natural logarithms.*

Remark 2.1. In Theorem 2.1, the charges $\{q_1, q_2, \cdots, q_m\}$ can be replaced with dipoles, or with finite linear combinations of multipoles of the form

$$\alpha_0 \log(z) + \sum_{i=1}^{n} \frac{\alpha_i}{z^i}. \tag{19}$$

In this case, the form of the expressions (5)-(8) is unchanged. However, the coefficients Q, A, α, $\{a_k\}$, $k=1,2,\cdots$, now depend on $\alpha_0, \alpha_1, \cdots, \alpha_m$, and can be easily determined by repeated differentiation of (6) with respect to z_i, $i=1,2,\cdots,m$.

Remark 2.2. If in Lemma 2.3, the field $\phi(z)$ is generated by a single charge located at z_0, then the only nonzero term in the expansion (11) is a_0, and $a_0 = q_1$. Similarly, if the field $\phi(z)$ is generated by a single dipole located at z_0, then the only nonzero term in the expansion (11) is a_1, and $a_1 = q_1$.

Lemma 2.4 provides a formula for shifting the center of a local expansion. The expression (20) below is an exact one, and no error bound is needed.

LEMMA 2.4. *For any complex* z_0, z, *and* $\{a_k\}$, $k=1,2,\cdots,n$,

$$\sum_{k=0}^{n} a_k(z-z_0)^k = \sum_{l=0}^{n}\left(\sum_{k=l}^{n} a_k \binom{k}{l}(-z_0)^{k-l}\right) z^l. \tag{20}$$

Remark 2.3. One of the advantages of using expansions of the forms (5) and (15) for representing potential fields is the fact that these expansions possess simple analytical derivatives. This permits the force fields to be obtained from the potentials by Lemma 2.1, without the use of numerical differentiation and the attendant loss of accuracy.

3. The adaptive multipole algorithm.

3.1. General strategy. In this section, we describe an adaptive algorithm for the rapid evaluation of the potential and electrostatic fields due to arbitrary distributions of charges and/or dipoles. The main strategy is similar to that described in [7]. It consists of clustering particles at different spatial lengths and using multipole expansions to evaluate the interactions between clusters that are sufficiently far away from each other. The interactions between nearby particles are computed directly.

To be more specific, consider the domain depicted in Fig. 2. N charges are arbitrarily distributed in $\mathbb{R}^2$, and, without loss of generality, we can assume that all of them are located inside a square with sides of length one, centered about the origin of the coordinate system. This square will be referred to as the computational box.

Given a machine precision ε, we set the number of terms in all expansions to $p \approx |\log_2(\varepsilon)|$, and specify that no interactions be evaluated via multipole expansions for groups of particles that are not well separated. This is precisely the condition needed for the error bounds (7), (14), (18) to apply with $c=2$. In order to impose such a condition, we introduce a hierarchy of meshes which refine the computational box into smaller and smaller regions (Fig. 3). Mesh level 0 corresponds to the entire computational box, while mesh level $l+1$ is obtained from mesh level l by subdividing each region into four equal parts. A tree structure is imposed on this mesh hierarchy,

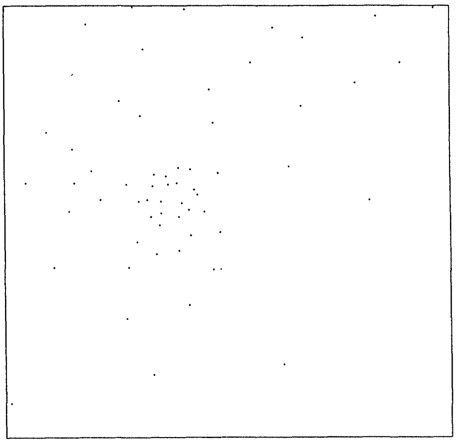

FIG. 2. *Nonuniform distribution of charges in the computational cell.*

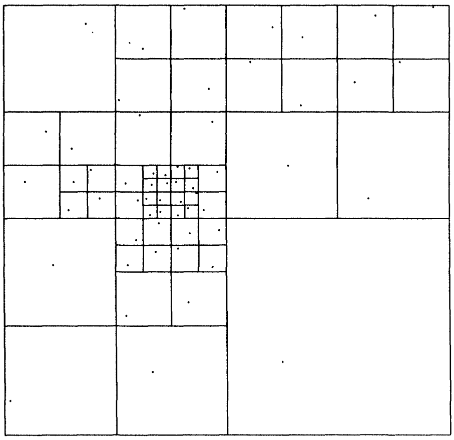

FIG. 3. *The hierarchy of meshes partitioning the computational cell.*

so that if b is a fixed box at level l, the four boxes at level $l+1$ obtained by subdividing b are considered its children. The four children of the same box will be referred to as brothers.

However, unlike the algorithm of [7], we do not use the same number of levels for all parts of the computational box. Instead, some integer $s>0$ is fixed, and at every level of refinement we subdivide only those boxes that contain more than s charges. Generally, this results in a large number of empty boxes at finer levels of the procedure. At every level of refinement, a table of nonempty boxes is maintained, so that once an empty box is encountered, its existence is immediately forgotten and it is completely ignored by the subsequent process.

OBSERVATION 3.1. It should be noted that for a fixed machine precision ε, only certain classes of particle distributions can be modeled, independently of the algorithm used. Namely, suppose that two charges c_1, c_2 in a distribution have positions x_1, x_2 and that $\|x_1-x_2\| < \varepsilon/2 \cdot \|x_1+x_2\|$. Obviously, under these conditions the particles c_1, c_2 cannot be discerned, and no meaningful simulation is possible. Since the smallest discernible distance between two particles depends on the position of these particles in the computational cell, such position-dependent condition cannot be imposed a priori. In order to make the simulation possible, we will simply require that $r_{\min} > \varepsilon$, where $r_{\min}$ is the smallest distance between any two particles in the simulation, and ε is the machine precision. Therefore, the maximum number of ancestors for any box in the computational cell is $p = |\log_2(\varepsilon)|$.

3.2. Notation. In this section, we introduce several definitions to be used in the description of the algorithm below.

For any subset A of the computational box, $T(A)$ will denote the set of particles that are contained in A.

B_l is the set of nonempty boxes at level l. B_0 consists of only the computational box itself. We will denote by *nlev* the highest level of refinement at any point.

If a box contains more than s particles, it is called a *parent box*. Otherwise, the box is said to be *childless*.

A *child box* is a nonempty box resulting from the division of a parent box into four.

Colleagues are adjacent boxes of the same size (at the same level). A given box has at most eight colleagues (Fig. 4).

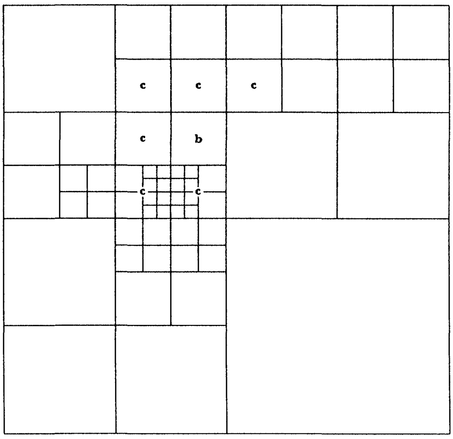

FIG. 4. *Box* (b) *and its colleagues* (c).

With each box b at level l we will associate five lists of other boxes, determined by their positions with respect to b. Following are the definitions of these lists (Fig. 5).

List 1 of a box b will be denoted by U_b; it is empty if b is a parent box. If b is childless, U_b consists of b and all childless boxes adjacent to b.

List 2 of a box b will be denoted V_b and is formed by all the children of the colleagues of b's parent that are well separated from b.

List 3 of a box b will be denoted by W_b. W_b is empty if b is a parent box, and consists of all descendants of b's colleagues whose parents are adjacent to b, but who are not adjacent to b themselves, if b is a childless box. Note that b is separated from each box w in W_b by a distance greater than or equal to the length of the side of w.

List 4 of a box b will be denoted by X_b and is formed by all boxes c such that $b \in W_c$. Note that all boxes in List 4 are childless and larger than b.

List 5 of a box b will be denoted by Y_b and consists of all boxes that are well separated from b's parent.

Φ_b will denote the p-term multipole expansion about the center of b of the field created by all particles in $T(b)$.

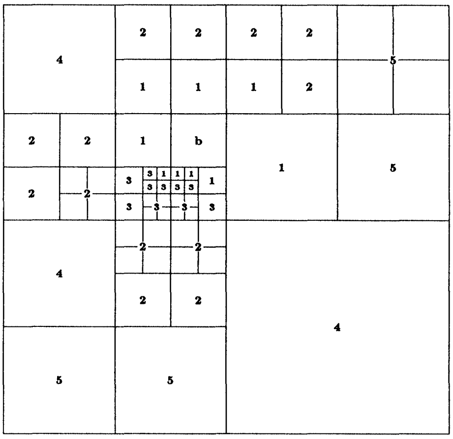

FIG. 5. *Box* (b) *and the associated Lists* 1-5.

Ψ_b will denote the p-term local expansion about the center of box b of the field created by all particles located outside $T(U_b) \cup T(W_b)$. $\Psi_b(r)$ is the result of the evaluation of the expansion Ψ_b at a particle r in $T(b)$.

Γ_b will denote the local expansion about the center of b of the field due to all particles in $T(V_b)$.

Δ_b will denote the local expansion about the center of b representing the field due to all charges located in $T(X_b)$.

$\alpha_b(r)$ will denote the field at $r \in T(b)$ due to all particles in $T(U_b)$.

$\beta_b(r)$ will denote the field at $r \in T(b)$ due to all particles in $T(W_b)$.

3.3. Informal description of the algorithm. The algorithm can be viewed as a recursive process of subdividing the computational cell into increasingly finer meshes (see Figs. 2-3). For a fixed box b at level l, the computational cell is partitioned into five subsets, U_b, V_b, W_b, X_b, and Y_b, and the following procedure is applied to the sets of particles $T(U_b)$, $T(V_b)$, $T(W_b)$, $T(X_b)$, and $T(Y_b)$.

(1) For each childless box b we combine the particles in $T(b)$ by means of Theorem 2.1 to form a multipole expansion Φ_b. For each parent box B we use Lemma 2.2 to merge the multipole expansions of its children b_1, b_2, b_3, b_4 into the expansion Φ_B.

(2) The interactions between particles in $T(b)$ and $T(U_b)$ are computed directly. For each particle $r \in T(b)$, the result of these calculations is $\alpha_b(r)$.

(3) We use Lemma 2.3 to convert the multipole expansion of each box in V_b into a local expansion about the center of b, and add the resulting expansions to obtain Γ_b.

(4) For every particle r in b, we compute the field $\beta_b(r)$ due to all particles in $T(W_b)$ by evaluating the p-term multipole expansions Φ_w of each box w in W_b at r, and adding them up.

(5) We convert the field of each particle in $T(X_b)$ into a local expansion about the center of box b (see Remark 2.1), and add up the resulting expansions to obtain Δ_b.

(6) We shift the center of the local expansion Γ_B of b's parent B to the centers of b and the other children of B by means of Lemma 2.4. We add the local expansion obtained to Γ_b.

(7) For each box b, we evaluate the sum of the local expansions Γ_b and Δ_b at every particle r in b and add the result to $\alpha_b(r)$ and $\beta_b(r)$ obtaining the field at r.

Remark 3.1. Note that in the above procedure we never explicitly evaluate the interactions between particles in $T(b)$ and those in $T(Y_b)$. Indeed, since all boxes in Y_b are well separated from b's parent, the interaction between $T(Y_b)$ and $T(b)$ have been accounted for during steps (3) and (5) at a coarser level.

3.4. Formal description of the algorithm.

ALGORITHM.

Comment [Choose main parameters.]

Choose precision ε to be achieved. Set the number of terms in all expansions to $p \approx \log_2(\varepsilon)$.

Choose the maximum number s of particles in a childless box.

Stage 1

Comment [Refine the computational cell into a hierarchy of meshes.]

```
do l = 1, 2, ···
  do $b_i \in B_l$
    if $b_i$ contains more than $s$ particles then
      subdivide $b_i$ into four boxes, ignore the empty boxes formed,
      add the nonempty boxes formed to $B_{l+1}$.
    end if
  end do
end do
```

Comment [We denote by *nlev* the highest level of refinement, and by *nbox* the total number of boxes formed in Stage 1.]

Stage 2

Comment [For every box b at every level l, form a multipole expansion representing the field outside b due to all the particles contained in b.]

Step 2.1

Comment [For each childless box b, use Theorem 2.1 to combine all charges inside b to obtain the multipole expansion about the center of b.]

```
do i = 1,nbox
  if $b_i$ is a childless box, use Theorem 2.1 to form a $p$-term expansion $\Phi_{b_i}$
  representing the field outside $b_i$ due to all charges located in $b_i$.
end do
```

Step 2.2

Comment [For each parent box b, use Lemma 2.2 to obtain the multipole expansion Φ_b by shifting the centers of the expansions of b's children to b's center, and adding the resulting expansions together.]

```
do l = nlev-1,1,-1
  do $b_i \in B_l$
    if $b_i$ is a parent box then
      use Lemma 2.2 to shift the center of each of $b_i$'s child box's expansion to
      $b_i$'s center. Add the resulting expansions together to obtain the expansion
      $\Phi_{b_i}$.
    end if
  end do
end do
```

Stage 3

Comment [For all particles in each childless box b, compute the interactions with all particles in $T(U_b)$ directly.]

```
do i = 1,nbox
  if $b_i$ is childless then
    for each particle $r$ in $b_i$, compute the sum $\alpha_b(r)$ of the interactions between
    $r$ and all particles in $T(U_{b_i})$.
  end if
end do
```

Stage 4

Comment [For each box b, use Lemma 2.3 to convert the multipole expansions of all boxes in V_b into local expansions about the center of box b.]

```
do i = 1,nbox
  do $b_j \in V_{b_i}$
    Convert multipole expansion $\Phi_{b_j}$ about $b_j$'s center into a local expansion
    about $b_i$'s center using Lemma 2.3. Add the resulting expansions to obtain $\Gamma_{b_i}$.
  end do
end do
```

Stage 5

Comment [For each childless box b, evaluate the multipole expansions of all boxes in W_b at every particle position in b.]

```
do i = 1,nbox
  if $b_i$ is childless then
    Evaluate the multipole expansion $\Phi_{b_j}$ of each box $b_j \in W_{b_i}$ to obtain $\beta_{b_i}(r)$
    for every particle $r$ in box $b_i$.
  end if
end do
```

Stage 6

Comment [For each box b, use Lemma 2.3 and Remark 2.2 to form local expansions about the center b representing the field due to all particles in $T(X_b)$.]

```
do i = 1,nbox
  Convert the field of every particle in $T(X_{b_i})$ into a local expansion about the
  center of $b$.
end do
```

Stage 7

Comment [Use Lemma 2.4 to shift the centers of local expansions of parent boxes to the centers of their children.]

```
do l = 1,nlev-1
  do $b_i \in B_l$
    if $b_i$ is a parent box then
      by using Lemma 2.4, shift the center of expansion $\Gamma_{b_i}$ to the center of each
      of $b_i$'s children $b_j$. Add the resulting expansion to $\Gamma_{b_j}$.
    end if
  end do
end do
```

Stage 8

Comment [For each childless box b, obtain Ψ_b as the sum of local expansions Γ_b and Δ_b. For each particle r in a childless box b, evaluate $\Psi_b(r)$ and obtain the field at r by adding $\Psi_b(r)$, $\alpha_b(r)$, and $\beta_b(r)$ together.]

```
do i = 1,nbox
  if $b_i$ is childless then
    Compute $\Psi_{b_i} = \Gamma_{b_i} + \Delta_{b_i}$.
    For each particle $r$ in $b_i$, evaluate $\Psi_{b_i}(r)$.
    Add $\Psi_{b_i}(r)$, $\alpha_{b_i}(r)$, and $\beta_{b_i}(r)$ to obtain the field at $r$'s position.
  end if
end do
```

3.5. Complexity analysis.

Stage number	Operation count	Explanation
Stage 1	Np	Each particle is assigned to a box at every level. There are at most p levels of refinement.
Stage 2		
Step 2.1	Np	Each particle contributes to the p-term expansion of one childless box.
Step 2.2	$\frac{5}{2}p^3N/s$	The center of the expansion of each box is shifted to the center of the parent box. The number of boxes is bounded by $5pN/s$ (see Lemma A.5), and each shift requires $p^2/2$ work (see Lemma 2.2).
Stage 3	$22pNs$	Each childless box b contains less than s particles and the work required to compute all interactions between particles in two boxes is $s^2/2$ when Newton's third law is used. The number of boxes in all List 1's is bounded by $44pN/s$ (see Lemmas A.1 and A.4).
Stage 4	$80p^3N/s$	For each box, List 2 has no more than 32 entries (Lemma A.2). There are at most $5pN/s$ boxes (Lemma A.5) and each shift requires $p^2/2$ work (Lemma 2.3).
Stage 5	$32p^2N$	Each childless box b contains less than s particles. The interactions of all particles in b and a box in W_b require ps work. The total number of boxes in List 3 is bounded by $32pN/s$ (Lemmas A.3 and A.4).
Stage 6	$32p^2N$	Each box in X_b contains less than s particles. The interactions between all particles in a box in X_b and box b require ps work. The total number of boxes in List 4 is bounded by $32pN/s$ (Lemmas A.3 and A.4).
Stage 7	$10p^3N/s$	Each box has at most four children. There are less than $5pN/s$ boxes (Lemma A.5) and a shift requires $p^2/2$ work (Lemma 2.4).
Stage 8	$Np + N$	A p-term expansion is evaluated at each particle position. The sums require an extra N work.

Summing up the CPU times for all stages above, we obtain the following time estimate:

$$T = N(92.5ap^3/s + 64bp^2 + 22cps + 3dp + e), \tag{21}$$

where the coefficients a, b, c, d, e depend on the computer system, language, implementation, etc. However, the parameter s (maximum permitted number of particles in a childless box) in (21) is not determined by the problem and can be chosen so as to minimize the resulting CPU time estimate. Differentiating (21) with respect to s, we obtain

$$s_{\min} = \sqrt{92.5a/18c}\, p \tag{22}$$

and

$$T_{\min} = N(\alpha p^2 + \beta p + \gamma) = N(\alpha \log_2^2(\varepsilon) + \beta |\log_2(\varepsilon)| + \gamma), \tag{23}$$

with the constants α, β, γ determined by the computer system, language, implementation, etc.

The storage requirements of the algorithm are determined by the number of nonempty boxes which is bounded by $5pN/s$. For each box we store the coefficients of a p-term multipole expansion and a p-term local expansion. The positions and charges of each particle also have to be stored. Therefore, the storage requirements are of the form

$$S = N(10fp/s + 3g) = N(10f|\log_2(\varepsilon)|/s + 3g), \tag{24}$$

where the coefficients f, g depend on the computer system, language, implementation, etc.

4. Numerical results. A computer program using the algorithm described in the preceding section has been implemented, and numerical experiments have been performed on a VAX-8600.

To evaluate the robustness of the adaptive scheme, we considered a variety of particle distributions. For each distribution, the corresponding fields were computed in four ways: by the algorithm of the present paper, by the algorithm described in [7], and directly in single and double precision. The direct calculation of the field in double precision was used as a standard for comparing the relative accuracies of the other three methods. In these experiments, the number of particles varied between 100 and 25,600, with charge strengths randomly assigned between 0 and 1.

The results are summarized in Tables 1-4. The first column of each table contains the number of particles N for which calculations have been performed. In the remaining

TABLE 1

Uniformly distributed particles. $p = 20$ and $s = 30$.

N	T_{alg}	T_{uni}	T_{dir}	E_{alg}	E_{uni}	E_{dir}	S_{alg}	S_{uni}
100	0.15	0.47	0.15	1.7×10^{-6}	4.0×10^{-7}	1.7×10^{-6}	866	4179
200	0.43	0.65	0.61	9.3×10^{-7}	4.3×10^{-7}	4.4×10^{-7}	2503	5479
400	1.01	1.94	2.47	7.0×10^{-7}	6.4×10^{-7}	6.4×10^{-7}	3763	16847
800	2.45	2.78	10.27	4.1×10^{-7}	4.0×10^{-7}	4.7×10^{-7}	11203	22047
1600	5.37	8.56	42.35	3.7×10^{-7}	4.2×10^{-7}	5.4×10^{-7}	15923	67519
3200	10.60	11.80	152.95	5.0×10^{-7}	5.3×10^{-7}	8.7×10^{-7}	44423	88319
6400	23.38	33.49	601.18	7.0×10^{-7}	5.4×10^{-7}	1.3×10^{-6}	65907	270207
12800	45.34	48.02	2433.20	6.0×10^{-7}	4.9×10^{-7}	1.6×10^{-6}	176631	353407
25600	96.72	137.68	9694.45	8.3×10^{-7}	8.9×10^{-7}	2.2×10^{-6}	268723	1080959

TABLE 2
Particles distributed on a curve. $p = 17$ *and* $s = 30$.

N	T_{alg}	T_{uni}	T_{dir}	E_{alg}	E_{uni}	E_{dir}	S_{alg}	S_{uni}
100	0.11	0.38	0.16	3.4×10^{-5}	3.2×10^{-5}	3.4×10^{-5}	1149	3927
200	0.30	0.54	0.57	8.9×10^{-6}	9.3×10^{-6}	8.9×10^{-6}	2694	5227
400	0.64	1.31	2.29	5.6×10^{-5}	5.6×10^{-5}	5.6×10^{-5}	5103	15827
800	1.46	3.13	9.30	9.4×10^{-5}	9.5×10^{-5}	9.5×10^{-5}	10133	21027
1600	2.66	5.94	37.41	2.0×10^{-5}	2.0×10^{-5}	2.0×10^{-5}	19241	63427
3200	5.93	12.50	149.21	7.8×10^{-6}	8.7×10^{-6}	8.8×10^{-6}	40055	84227
6400	12.42	29.66	597.95	4.2×10^{-5}	4.2×10^{-5}	4.2×10^{-5}	84429	253827
12800	25.11	79.47	2425.48	8.7×10^{-5}	8.7×10^{-5}	8.8×10^{-5}	167421	337027
25600	47.53	152.07	9581.20	8.9×10^{-5}	9.1×10^{-5}	8.9×10^{-5}	332927	1015427

TABLE 3
Highly nonuniform distribution of particles. $p = 17$ *and* $s = 30$.

N	T_{alg}	T_{uni}	T_{dir}	E_{alg}	E_{uni}	E_{dir}	S_{alg}	S_{uni}
100	0.19	0.45	0.15	2.7×10^{-6}	1.0×10^{-5}	2.8×10^{-6}	2508	3927
200	0.48	0.74	0.57	6.9×10^{-6}	7.6×10^{-6}	6.9×10^{-6}	4014	5227
400	1.13	2.26	2.33	1.9×10^{-6}	9.0×10^{-6}	1.9×10^{-6}	8307	15827
800	2.25	5.15	9.34	4.3×10^{-6}	6.0×10^{-6}	3.7×10^{-6}	13353	21027
1600	5.09	16.17	37.74	2.4×10^{-6}	1.6×10^{-6}	2.1×10^{-6}	25588	63427
3200	9.98	50.23	149.86	3.7×10^{-6}	1.4×10^{-6}	1.7×10^{-6}	46806	84227
6400	21.80	177.13	606.14	5.8×10^{-6}	4.0×10^{-6}	5.9×10^{-6}	90505	253827
12800	41.93	663.21	2420.33	4.0×10^{-6}	4.0×10^{-6}	4.2×10^{-6}	186226	337027
25600	90.05	2317.93	9622.63	2.9×10^{-6}	3.0×10^{-6}	4.0×10^{-6}	373639	1015427

TABLE 4
Nonuniform distribution of particles in a region of complicated shape. $p = 17$ *and* $s = 30$.

N	T_{alg}	T_{uni}	T_{dir}	E_{alg}	E_{uni}	E_{dir}	S_{alg}	S_{uni}
100	0.15	0.43	0.15	4.3×10^{-5}	5.5×10^{-5}	5.0×10^{-5}	1145	3927
200	0.39	0.68	0.59	3.3×10^{-5}	3.9×10^{-5}	3.3×10^{-5}	3224	5227
400	0.84	1.69	2.31	8.1×10^{-5}	7.1×10^{-6}	8.1×10^{-5}	6939	15827
800	2.11	5.03	9.39	4.3×10^{-5}	4.3×10^{-5}	4.3×10^{-5}	13406	21027
1600	4.35	11.34	37.74	9.2×10^{-5}	9.2×10^{-5}	9.2×10^{-5}	24913	63427
3200	9.16	30.85	153.76	1.1×10^{-5}	1.1×10^{-5}	1.1×10^{-5}	48902	84227
6400	19.22	48.62	611.82	5.4×10^{-6}	5.5×10^{-6}	5.4×10^{-6}	96153	253827
12800	37.92	155.75	2440.90	2.1×10^{-5}	2.0×10^{-5}	2.1×10^{-5}	194377	337027
25600	80.02	248.90	9798.34	4.4×10^{-5}	4.4×10^{-5}	4.5×10^{-5}	388624	1015427

columns, the upper case letters T, E, and S are used to denote the corresponding computational time, error, and storage, with the subscripts *alg*, *uni*, and *dir* referring to the adaptive algorithm, the algorithm described in [7], and the direct (single-precision) calculation, respectively. More specifically, columns 2 through 4 show the times, in seconds, required to compute the field by the three methods. The errors E_{alg}, E_{uni} and E_{dir} for the adaptive, nonadaptive and direct methods, respectively, are presented in the next three columns. They are defined by the formula

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

where f_i is the value of the field at the ith 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. The last two columns of the tables contain the storage requirements S_{alg} and S_{uni}, in single-precision words, for the two methods based on multipole expansions.

Remark 4.1. For the tests involving 12,800 and 25,600 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. Therefore, we have performed the direct calculations in double precision 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.

For the first set of tests, the positions of the charged particles were randomly distributed in a square, and the resulting particle density was roughly uniform (Fig. 6). The number of terms in the expansions was set to 20, and the maximum number of particles in a childless box was set to 30.

In the second set of experiments, the charged particles were distributed along a curve (Fig. 7). The number of terms in the expansions was set to 17 and the maximum number of particles in a childless box was set to 30.

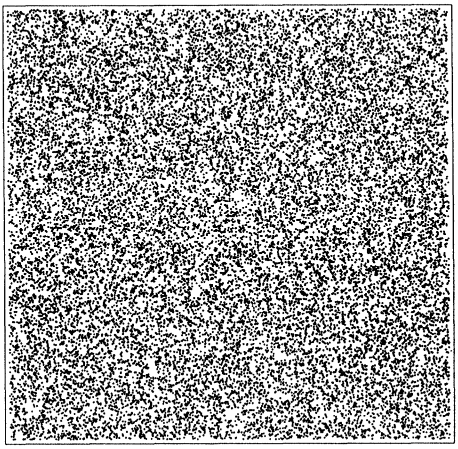

FIG. 6. 25,600 *uniformly located charges in the computational cell.*

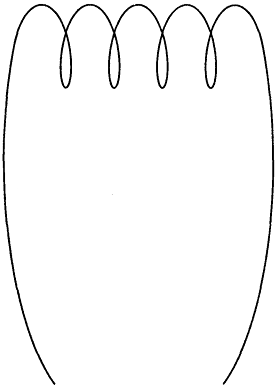

FIG. 7. 6400 *particles distributed on a curve.*

The third set of numerical experiments was performed on extremely nonuniform distributions of particles (Fig. 8). A fifth of the N particles were randomly assigned in a square of area 1. Two-fifths were randomly distributed about the center of the square in a circle of radius 0.003. The rest of the particles were assigned positions inside a circle of radius 0.5 with a density inversely proportional to the square of the distance from the center. The number of terms in the expansions was set to 17 and the maximum number of particles per childless box was set to 30.

In the last set of experiments, half of the particles were distributed along a curve similar to that of the second set of experiments and the rest of the particles were distributed inside four circles with a density inversely proportional to the square of the distance to the centers of the circles (Fig. 9). The number of terms in the expansions was set to 17 and the maximum number of particles per childless box was set to 30.

The following observations can be made from Tables 1–4, where the results of the experiments described above are summarized.

(1) The accuracies of the results obtained by the algorithms using multipole expansions are in agreement with the error bounds given in (7), (14), and (18). For the most part, the fast methods are slightly more accurate than the direct calculation.

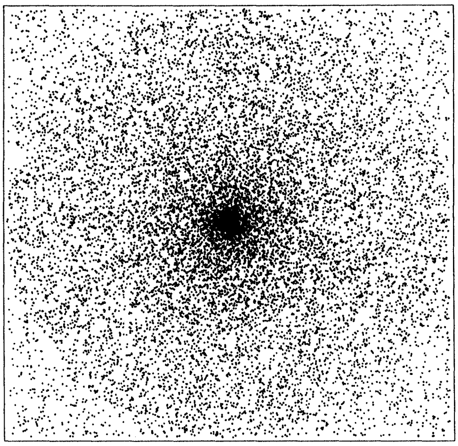

FIG. 8. *Highly nonuniform distribution of* 25,600 *charges.*

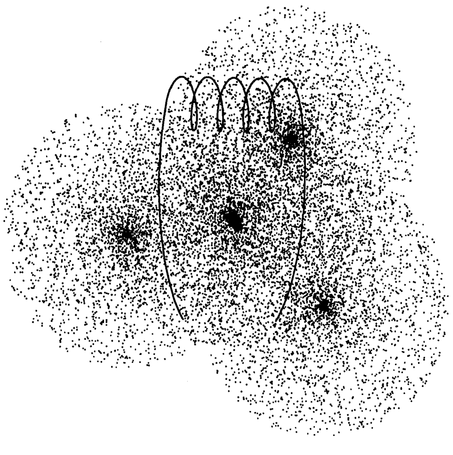

FIG. 9. *A nonuniform distribution of* 25,600 *particles in a region of complicated shape.*

(2) In all cases, the actual CPU time requirements of the adaptive algorithm grow linearly with N. The CPU time requirements of the nonadaptive counterpart grow linearly, except for extremely nonuniform distributions (see Tables 3 and 4).

(3) Even for uniform distributions of charges, the adaptive algorithm is about 30 percent faster than the nonadaptive one.

(4) The storage requirements of both fast algorithms are roughly proportional to the number of particles involved in the simulations. The storage requirements of the adaptive algorithm are about four times less than those of the nonadaptive version.

(5) By the time the number of particles reaches 25,600, the adaptive algorithm is about 100 times faster than the direct version for the case of the uniform distribution (see Table 1). When the charges are situated on a curve, the adaptive scheme is roughly 200 times faster than the direct algorithm, and about three times faster than the fast nonadaptive scheme (see Table 2). For the highly nonuniform case (see Table 3), the adaptive algorithm is slightly more efficient than for the uniform distribution. The nonadaptive scheme displays an almost quadratic growth of CPU time with N, and is about 25 times slower than its adaptive counterpart by the time $N = 25{,}600$.

(6) Even for as few as 1600 particles, the adaptive algorithm is about ten times faster than the direct calculation.

(7) The performance of the algorithm does not depend on the shape of the region where the charges are distributed (see Table 4).

5. Conclusions. An adaptive algorithm has been constructed for the rapid evaluation of the potentials and force fields due to large-scale ensembles of particles of the type encountered in plasma physics, molecular dynamics, fluid dynamics (the vortex method), and celestial mechanics. The algorithm is applicable whenever the fields to be evaluated are Coulombic or gravitational in nature, and it yields the potentials to within round-off error.

The asymptotic CPU time estimate for the algorithm is of the order $O(N)$, where N is the number of particles in the simulation, and this estimate is independent of the statistics of the charge distribution. Our numerical experiments indicate a tendency of the scheme to be more efficient for nonuniform distributions than for uniform ones. The storage requirements of the algorithm are of the order $O(N)$, do not depend on the statistics of the distribution, and tend to be quite acceptable even for very large numbers of particles.

In the present paper, a two-dimensional version of the algorithm is described. Generalizing it to the three-dimensional case is fairly straightforward, and will be reported at a later date.

Appendix. In this Appendix, we prove several combinatorial lemmas that are used in § 3.5 to estimate the complexity of the algorithm described in §§ 3.3 and 3.4. We begin by introducing some additional notation.

Given a subdivision S of the computational cell and a childless box b in S, we will denote by S_b the subdivision obtained from S by subdividing b into four equal boxes, and refer to the process of obtaining S_b from S as an elementary refinement of S.

For a subdivision S of the computational cell, we will denote by B_S the set of all boxes in S.

C_S will denote the subset of B_S consisting of all childless boxes, i.e., boxes that are nonempty and not subdivided.

F_S will denote the subset of B_S consisting of all nonempty boxes.

D_S is the subset of B_S consisting of all empty boxes that have a childless brother.

$G_S = C_S \cup D_S$ is the subset of B_S consisting of all boxes b such that b is either childless or has a childless brother.

For any set of boxes A, $N(A)$ will denote the number of boxes in A.

LEMMA A.1. *For any subdivision S of the computational cell*

$$\sum_{b \in C_S} N(U_b) \leqq 11 N(G_S). \tag{25}$$

Proof. We will prove the lemma by combining the following three observations:

(a) Inequality (25) holds for the undivided computational cell.

(b) Any subdivision of the computational cell can be obtained by a sequence of elementary refinements of the computational cell.

(c) If an elementary refinement is applied to a subdivision for which (25) holds, it also holds in the refined subdivision.

The statements (a) and (b) above are obvious, and following is a proof of (c).

Consider a subdivision S of the computational cell such that (25) is true for S, and a box b such that $b \in C_S$. Obviously

$$N(G_{S_b}) = N(G_S) + 3, \tag{26}$$

and we will denote by U_b and U'_b the List 1's of b with respect to S and S_b, respectively. The following obvious observations can be made about the List 1's of b and its children:

(1) For any box $c \in C_S$, if $b \in U_c$ then $c \in U_b$.

(2) Each child of b has its three brothers in its List 1.

(3) In the subdivision S_b, b is not childless and U'_b is empty.

(4) Each box c in U_b is in the List 1 of at least one child of b.

(5) The number of boxes of U_b that are in the List 1's of two children of b is bounded by eight.

It immediately follows from observations (1)-(5) above that

$$\sum_{p \in C_{S_b}} N(U_p) - \sum_{q \in C_S} N(U_q) = (4 \cdot 4) + 2[-N(U_b) + (N(U_b) + 8)] = 32, \tag{27}$$

and combining (26) and (27) we obtain

$$\sum_{p \in C_{S_b}} N(U_p) \leqq 11 N(G_{S_b}). \tag{28}$$

LEMMA A.2. *For any subdivision S of the computational cell*

$$\sum_{b \in F_S} N(V_b) \leqq 32 N(F_S). \tag{29}$$

Proof. Consider an arbitrary subdivision S of the computational cell, a box $c \in F_S$, and its parent box b. V_c is a subset of the children of b's colleagues, the maximum number of colleagues of c (or any other box) is eight, and each colleague can have four children. Therefore, the number of elements in V_c is bounded by 32.

LEMMA A.3. *For any subdivision S of the computational cell*

$$\sum_{c \in C_S} N(W_c) = \sum_{b \in F_S} N(X_b) \leqq 8 N(G_S). \tag{30}$$

Proof. The first part of the lemma is a direct consequence of the definition of List 4 (see § 3.2): If a box b belongs to W_c, then c belongs to X_b. Now, consider an arbitrary box $c \in F_S$, and its parent box b. The number of colleagues of b is certainly bounded by eight. We will denote by Z_b the set of all childless boxes which are adjacent to b and whose size is greater than or equal to that of b. The number of boxes in Z_b is bounded by eight, since each box in Z_b contains at least one of the eight colleagues of b, and no two such boxes can contain the same colleague. The second part of the lemma now follows from the obvious observation that $W_c \subset Z_b$.

LEMMA A.4. *For any subdivision S of the computational cell produced by the algorithm of* § 3,

$$N(C_S) \leqq N(G_S) \leqq 4p\frac{N}{s}. \tag{31}$$

Proof. Each parent box b at level l contains more than s particles (otherwise, it would not have been subdivided any further). Therefore, the total number of parent boxes at level l is bounded by N/s. Each of these boxes cannot have more than four children, and consequently the number of boxes in G_S at any level l is bounded by $4N/s$. Now, the conclusion of the lemma follows from Observation 3.1 and the obvious fact that $N(C_S) \leqq N(G_S)$.

LEMMA A.5. *For any subdivision S of the computational cell,*

$$N(F_S) \leqq 5p\frac{N}{s}. \tag{32}$$

Proof. The number of parent boxes at any level l is bounded by N/s, and each of them cannot have more than four childless boxes at level $l+1$. Therefore, the sum of the numbers of nonempty boxes (all childless and parent boxes) at all levels is bounded by $p(N/s+4N/s)$.

Acknowledgment. The authors thank Professor M. H. Schultz for several useful discussions and for his interest and support.

REFERENCES

[1] C. R. ANDERSON, *A method of local corrections for computing the velocity field due to a distribution of vortex blobs*, J. Comp. Phys., 62 (1986), pp. 111-123.

[2] A. W. APPEL, *An efficient program for many-body simulation*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 85-103.

[3] A. J. CHORIN, *Numerical study of slightly viscous flow*, J. Fluid. Mech., 57 (1973), pp. 785-796.

[4] A. GERASOULIS, M. GRIGORIADDIS, AND L. SUN, *A fast algorithm for Trummer's problem*, LCSR-TR-77, Department of Computer Science, Rutgers University, New Brunswick, NJ, 1985.

[5] A. GERASOULIS, *A fast algorithm for the multiplication of generalized Hilbert matrices with vectors*, LCSR-TR-79, Department of Computer Science, Rutgers University, New Brunswick, NJ, 1986.

[6] G. GOLUB, *Trummer problem*, SIGACT News 17 (1985), No. 2, ACM Special Interest Group on Automata and Computability Theory, p. 17.2.

[7] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, J. Comp. Phys., 73 (1987), pp. 325-348.

[8] R. W. HOCKNEY AND J. W. EASTWOOD, *Computer Simulation Using Particles*, McGraw-Hill, New York, 1981.

[9] L. REICHEL, *A matrix problem with applications to rapid solution integral equations*, Report, Department of Mathematics, University of Kentucky, Lexington, KY, 1986.

[10] V. ROKHLIN, *Rapid solution of integral equations of classical potential theory*, J. Comp. Phys., 60 (1985), pp. 187-207.