浙江大學

本科生毕业论文

二维快速多极算法的矩阵形式

学　　院：数学科学学院
专　　业：信息与计算科学
学生姓名：孙晨旭
学　　号：3170102187
指导教师：赖　俊

2021 年 2 月 19 日

摘要

. . . . . .

目录

Part I

毕业论文

1 绪论

2 正文

Part II

毕业论文开题报告

1 文献综述

2 开题报告

3 外文翻译

我们给出了一种快速评估包含多个粒子的系统的势能或场强的算法。更具体地说，我们考虑的是粒子间相互作用表现为自然界中库仑力或引力的形式。对于N 个粒子的系统，一般需要$O(N^2)$ 的工作量来评估其相互作用，除非用一些近似或截断的方法。本文所用方法在给定误差内工作量与N成正比，使得在电磁物理学、流体力学、分子动力学和天体力学中的大规模计算更为可行。

3.1 简介

一些领域已经很好地建立了利用粒子模拟的物理学系统研究方法，而且该方法正在被更多领域所重视。最典型的例子或许是天地动力学，许多近期的关于粒子模型的计算与研究工作却在电磁物理学、流体力学和分子动力学开展。

模拟方法主要分两大类。在某些感兴趣的时间间隔内，动态模拟追踪N个粒子的轨迹。给定一些初始位置$\{x_i\}$ 以及速度，每个粒子的轨迹由牛顿第二定律给出：

$$m_i \frac{\mathrm{d}^2 x_i}{\mathrm{d}t^2} = -\nabla_i \Phi \quad for\ i = 1, \ldots, N$$

其中m_i为第i个粒子的质量，然后力从势函数Φ 的梯度中获得。若我们对平衡时的情况更感兴趣而不是他们随时间变换的性质，我们可以需用Monte-Carlo 法。在这个例子中，势函数Φ包含大量粒子的计算，力图求出其最小值。

这篇文章中，我们只关注势能或力由一系列相互关系加和而成的情况。更具体地说，我们考虑的势有如下形式

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

其中Φ_{near}这一部分会快速衰减，$\Phi_{external}$ 这部分与粒子数量无关，Φ_{far} 这部分远场势是库伦或重力势。这类模型表述了许多天体物理、电磁物理学和分子动力学中的问题。在不可压缩流体的涡方法计算中，一个重要且昂贵的部分也有相同的结构（流量函数与涡旋是由Poisson 方程相关联的）。

在N个粒子的系统中，Φ_{near}需要与N成正比的计算量，$\Phi_{external}$ 也类似。但是库伦或重力势的衰减慢到要求我们计算所有的相互关系，这就导致CPU运算量达到$O(N^2)$ 阶。本文介绍的快速算法计算所有粒子间的相互关系只要$O(N)$阶。

先前已有很多关于减少N-体问题计算量的研究工作。Particle-in-cell methods [5]被仔细研究并成功应用，尤其在电磁物理学领域中。假定电势满足Poisson方程，计算区域被均匀网格划分，该方法有：

(1) 在网格点插值源点密度

(2) 应用“快速Poisson求解器”在网格上获得值

(3) 通过势与粒子位置的插值获得力

该方法的复杂度为$O(N + M\log M)$，其中M为网格点的个数。网格点的个数通常与粒子个数成比例，但是差一个小比例常数使得$M << N$。因此该方法渐进复杂度为$O(N\log N)$，实际中计算量通常被视为与N成比例。不幸的是，网格分割提供了有限的精度，高度不均匀的源点分布会导致性能降低。第三步中的数值微分将进一步的引入误差。

为了改进particle-in-cell的计算，近距离的相互作用可以通过直接结算获得，而远距离的相互关系通过网格划分获得，就得到了所谓particle-particle/particle-mesh （P^3M）方法。将此方法引用在涡旋计算中，见[1].然而，这些方法的性能都依赖于一种较为均匀分布的前提下，理论上确实可以允许获取任意高的精度。通常，若我们要求的精度相对低一些，这些粒子在某个矩形区域分布的均匀一些或不均匀一些都能通过P^3M 获得较为满意的结果。但若要求高精度（例如一些高度相关系统的建模中），运算时间就变得过多了。

Appel[2]介绍了一种“无网格”方法用于多体问题的计算，计算复杂度估计为$O(N\log N)$。该方法依赖于单极点（质量中心）近似用于计算远距离的力以及一些复杂的数据结构来追踪那些粒子簇成一团以使得近似计算有效。对特定类型的问题，该方法与单纯$O(N^2)$方法比有显著的加速。当粒子相对均匀与所需精度很高时性能就下降了。

我们给出的用多极展开求势能或力的方法可以应对任意精度，并且复杂度与N成比例。我们的方法与[7] 中解Laplace 方程的边值问题相似。在下一部分中，我们会给出必要的分析工具，在第三节中给出详细的算法描述。

3.2 物理与数学上的准备工作

在本文中，我们考虑一个二维物理模型，它由N个粒子组成，其势能与力是由库伦定律成对相互作用累加而成。考虑一个单位点电荷为于点$(x_0, y_o) = \mathbf{x}_0 \in \mathbb{R}^2$. 于是对任意点$\mathbf{x} = (x, y) \in \mathbb{R}^2$，其中$\mathbf{x} \neq \mathbf{x}_0$，由下述表达式描述了这种电荷引起的电位和静电场

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

以及

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

众所周知，$\phi_{\mathbf{x}_0}$在任意不包含$\mathbf{x}_0$ 的区域是调和函数。此外对任意调和函数u，存在一个解析函数$\omega : \mathbb{C} \to \mathbb{C}$ 使得$u(x, y) = Re(\omega(x, y))$, 并且在不考虑相差一个常量时$\omega$是唯一的。在本文剩余部分，我们考虑解析函数时，将不对二维点$(x, y) \in \mathbb{R}^2$和复数点$x + iy = z \in \mathbb{C}$做区分。注明

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

并且遵循标准做法，我们将$\log(z)$成为由电荷引起的的电势。介于我们将要引入更复杂的表达式，因此我们沿用复表示并将相应的解析函数称为电势。下一个引理是Cauchy-Riemann 方程的直接结果。

Lemma 3.2.1. *若$u(x, y) = Re(\omega(x, y))$表示$(x, y)$处的电势，那么对应的力场为*

$$\nabla u = (u_x, u_y) = (Re(\omega'), -Im(\omega'))$$

其中ω'是ω的导数。

下一个引理被用于获得m个电荷产生电势的多极展开。

Lemma 3.2.2. 令带电量为q的点电荷为于z_0，对任意z,满足$|z| > |z_o|$，有

$$\phi_{z_0}(z) = q\log(z - z_0) = q(log(z) - \sum_{k=1}^{\infty} \frac{1}{k}(\frac{z_0}{z})^k) \tag{3.2.1}$$

Proof. 注意到对$|z_0/z| < 1$有$\log(z - z_0) - \log(z) = \log(1 - z_0/z)$，于是如下表达式

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

在任意$|\omega| < 1$时成立。 □

Theorem 3.2.1. 假定m个点电荷强度为$\{q_i, i = 1, \ldots, m\}$分别为于点$\{z_i, i = 1, \ldots, m\}$，其中$|z_i| < r$，那么对于任意$z$，$|z| > r$，由电荷引起的电势为

$$\phi(z) = Q\log(z) + \sum_{k=1}^{\infty} \frac{a_k}{z^k} \tag{3.2.2}$$

其中

$$Q = \sum_{i=1}^{m} q_i \quad and \quad a_k = \sum_{i=1}^{m} \frac{-q_i z_i^k}{k} \tag{3.2.3}$$

进一步有，对任意$p \geq 1$

$$\begin{aligned} \left|\phi(z) - Q\log(z) - \sum_{k=1}^{p} \frac{a_k}{z^k}\right| &\leq \alpha \left|\frac{r}{z}\right|^{p+1} \\ &\leq (\frac{A}{c-1})(\frac{1}{c})^p \end{aligned} \tag{3.2.4}$$

其中

$$c = \left|\frac{z}{r}\right|, \quad A = \sum_{i=1}^{m} |q_i|, \quad \alpha = \frac{A}{1 - \left|\frac{r}{z}\right|} \tag{3.2.5}$$

Proof. 多级展开的形式(3.2.2)是由上面引理关于各源点直接累加获得，现在来分析误差界(3.2.4), 观察到

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

利用(3.2.3)带入可得

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

这就与上述结论一致了。注意到如果$c \geq 2$，则

$$\left|\phi(z) - Q\log(z) - \sum_{k=1}^{p} \frac{a_k}{z^k}\right| \leq A(\frac{1}{2})^p \tag{3.2.6}$$

□

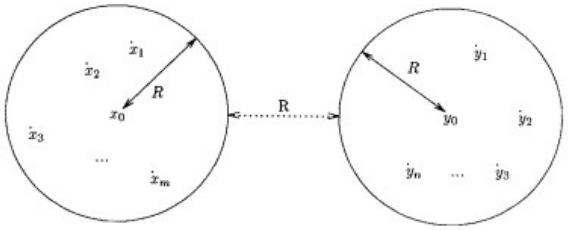

FIG. 1. Well-separated sets in the plane.

最后，我们用一个简单的例子说明多极展开可以加速电势场的运算。考虑电荷带电量为$q_1, q_2, \ldots, q_m$分别为于点$x_1, x_2, \ldots, x_m \in \mathbb{C}$并且$\{y_1, y_2, \ldots, y_n\}$是另一组点(见fig1).我们说点集$\{x_i\}$与$\{y_i\}$是well separated ,如果存在点$x_0, y_0 \in \mathbb{C}$以及实数$r > 0$使得

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

为了从点$\{x_i\}$直接获取点$\{y_i\}$处的电势（或力），我们可以计算

$$\sum_{i=1}^{m} \phi_{x_i}(y_i) \quad for\ all\ j = 1, \ldots, n \tag{3.2.7}$$

这显然需要nm阶的工作量。现在假定我们利用Theorem3.2.1先计算x_0的各个电荷的p 次多极展开式系数，这需要mp步。计算y_i处的多级展开式的值需要np 步，于是总工作量就是$O(mp + np)$ 阶的。进一步，利用(3.2.6)

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

然后，为了获得对应的精度ϵ,我们需要令p 为$-\log_2(\epsilon)$ 阶的。一旦精度明确了，总的计算量就减少为

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

对较大的n，m而言，这就远远小于nm的工作量。

1) 算子平移与误差界

接下来的三个引理是本文的主要分析工具，让我们操作多极展开式用以快速算法。Lemma3.2.3提供一种平移展开式中心的算法。Lemma3.2.4 展示如何将展开式在在一个解析性的圆形区域转化为局部Taylor 展开。Lemma3.2.5 提供了一种在分析区域内移动泰勒展开中心的机制。我们还导出了与这些平移算子相关联的误差界，使我们能够对任何指定的精度进行数值计算。

Lemma 3.2.3. 假定

$$\phi(z) = a_0 \log(z - z_0) + \sum_{k=1}^{\infty} \frac{a_k}{(z - z_0)^k} \tag{3.2.8}$$

为m个强度为$q_1, q_2, \ldots, q_m$的电荷在以z_0 为圆心，R 为半径的圆D内形成的电势的多极展开。对于以$R + |z_0|$ 为半径，原点为中心的圆D_1,z 落在其外，有

$$\phi(z) = a_0 \log(z) + \sum_{l=1}^{\infty} \frac{b_l}{z^l} \tag{3.2.9}$$

其中

$$b_l = -\frac{a_0 z_0^l}{l} + \sum_{k=1}^{l} a_k z_0^{l-k} \binom{l-1}{k-1} \tag{3.2.10}$$

其中$\binom{l}{k}$表示二项式系数。进一步，对任意$p \geq 1$，

$$\left| \phi(z) - a_0 \log(z) + \sum_{l=1}^{p} \frac{b_l}{z^l} \right| \leq \left(\frac{A}{1 - \left| \frac{|z_0|+R}{z} \right|} \right) \left| \frac{|z_0|+R}{z} \right|^{p+1} \tag{3.2.11}$$

其中A如(3.2.5)所定义。

Proof. 平移后的展开式系数(3.2.9)是由(3.2.8)关于z_0Taylor 展开得到的。对于误差界(3.2.11)，观察到$\{b_l\}$是圆D中包含电荷的唯一多级展开式的系数，于是由Theorem3.2.1中$|z_0| + R$ 替换r 即可的。 □

Remark. 展开式(3.2.8)中关于z_0的值$\{a_1, a_2, \ldots, a_p\}$被求出，我们就可以由(3.2.10) 得到$\{b_1, b_2, \ldots, b_p\}$。换言之，我们不必牺牲计算精度就可以平移截断的多级展开式中心。

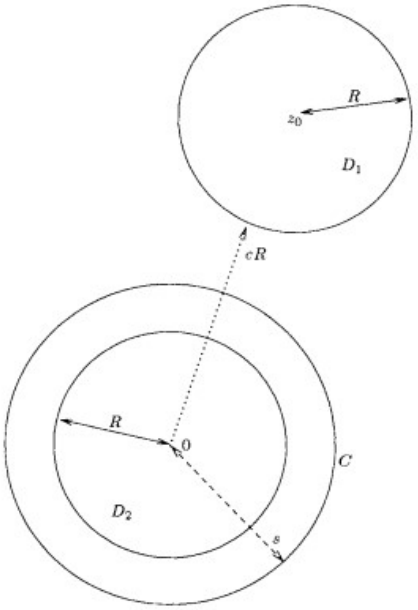

FIG. 2. Source charges $q_1, q_2, ..., q_1$ are contained in the circle D_1. The corresponding multipole expansion about z_0 converges inside D_2. C is a circle of radius s, with $s > R$.

Lemma 3.2.4. 假定m个强度为$q_1, q_2, \ldots, q_m$的电荷分布在以z_0 为圆心，R为半径的圆D_1 内，$|z_0| > (c+1)R, c > 1$*(fig2)*. 那么(3.2.8)在半径为R，原点为圆心的圆D_2 内收敛。在D_2 内，由电荷引起的电势可由幂级数表示

$$\phi(z) = \sum_{l=1}^{\infty} b_l \cdot z^l \tag{3.2.12}$$

其中

$$b_0 = a_0 \log(-z_0) + \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} (-1)^k \tag{3.2.13}$$

以及

$$b_l = -\frac{a_0}{l \cdot z_0^l} + \frac{1}{z_0^l} \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} \binom{l+k-1}{k-1} (-1)^k, \quad for\ l \geq 1 \tag{3.2.14}$$

进一步，对任意$p \geq \max(2, 2c/(c-1))$，截断误差为:

$$\left|\phi(z) - \sum_{l=1}^{p} b_l \cdot z^l\right| < \frac{A(4e(p+c)(c+1)+c^2)}{c(c-1)}(\frac{1}{c})^{p+1} \tag{3.2.15}$$

其中A如(3.2.5)定义，e是自然对数的底。

Proof. 我们利用Maclaurin定理从展开式(3.2.8) 获得(3.2.12),为了获得误差界(3.2.15),我们令$\gamma_0 = a_0 \log(-z_0), \gamma_l = -(a_0/l \cdot z_0^l)\ for\ l \geq 1$以及$\beta_l = b_l - \gamma_l\ for\ l \geq 0$。于是

$$\left|\phi(z) - \sum_{l=0}^{p} b_l \cdot z^l\right| = \left|\sum_{l=p+1}^{\infty} b_l \cdot z^l\right| \leq S_1 + S_2 \tag{3.2.16}$$

其中

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

S_1的界如下可得

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

为了获得S_2的界，我们令C为半径为s的圆，其中$s = cR((p-1)/p)$(fig2)，注意到对任意$p \geq 2c/(c-1)$

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

定义函数$\phi_1 : \mathbb{C} \backslash D_1 \to \mathbb{C}$

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

然后利用复解析函数的Taylor定理[6, p. 190]，我们有

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

其中

$$M = \max_{C} |\phi_1(t)|$$

显然对任意C中的t

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

并且可以得到

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

经过一些代数运算，我们有

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

观察到对任意正整数n以及整数$p \geq 2$

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

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

将最后的式子与S_1误差界结合即可得结果。 □

下面的引理是Maclaurin定理的直接结果。它描述了一个具有有限项展开式的精确平移操作，并且不需要误差界。

Lemma 3.2.5. *对任意复数z_0, z与$\{a_k\}k = 0, 1, \ldots, n$*

$$\sum_{k=0}^{n} a_k (z-z_0)^k = \sum_{l=0}^{n}\left(\sum_{k=l}^{n} a_k \binom{k}{l}(-z_0)^{k-l}\right) z^l. \tag{3.2.17}$$

3.3 快速多极算法

在本节中，我们给出一种基于电荷分布求出电势或电场的快速算法。使用的主要技巧是将粒子分为不同距离的点簇然后通过多级展开计算那些充分远的粒子间的相互作用。而那些离得近的粒子间的相互作用就直接计算。

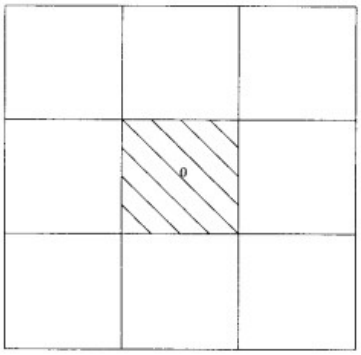

FIG. 3. The computational box (shaded) and its nearest periodic images. The box is centered at the origin "0" and has area one.

更具体地说，让我们考虑如fig3所示的计算区域。它是边长为1 的正方形，中心位于原点，并假定这个区域包含了所有N个粒子。8个相邻的区域也被画出，并且在下一节中考虑边界条件时被用到。首先我们考虑free-space问题，即边界可以被忽略且只有粒子间的相互作用需要被考虑进去。

给定一个精度ϵ，我们选取$p \approx \log_2(\epsilon)$并且明确没有well-separated的粒子簇间的相互作用不会被计算。这些条件与$c = 2$ 是满足误差界(3.2.4)、(3.2.11) 和(3.2.15)所需要的。此时截断误差为2^{-p}，也就满足了我们想要的误差。为了满足这个条件，我们引入一个将计算区域细分为更小区域的划分方法（fig4）.划分的第0层即为整个方形区域，第l+1层是由第l层通过等分为4个区域得来的。在第l层的方块数为4^l。我们利用树结构描述这个分层结构，因此如果第l层有一个区域$ibox$，那么l+1层的四个方块就视为$ibox$ 的子树。

假定我们

其他算法中使用的记号如下

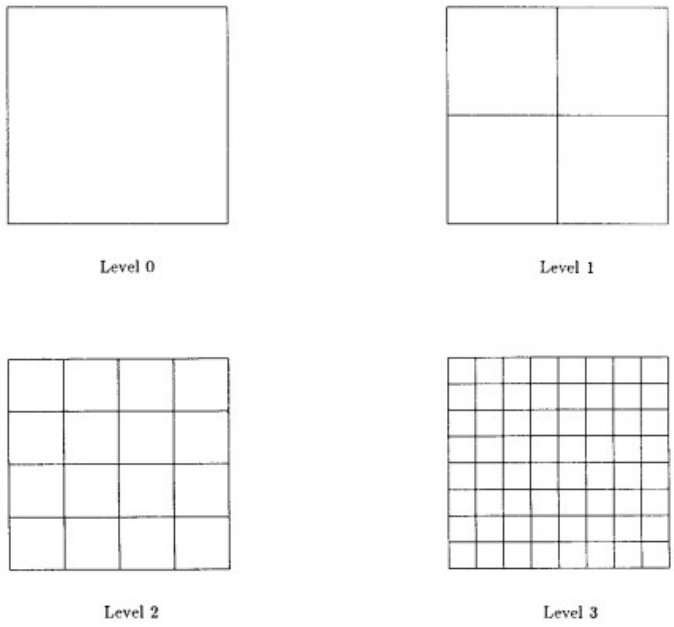

FIG. 4. The computational box and three levels of refinement.

Φ_{li} 由第l层第i个方形区域内的粒子产生的电势场的p 阶多级展开（关于该区域中心展开）

$\Psi_{l,i}$ 由第l层第i个方形区域以及其近邻区域外的粒子产生的电势场的p 阶展开式。

$\tilde{\Psi}_{l,i}$ 关于第l层第i个方形区域的局部展开，描述了由该方块的父区域以及父区域近邻区域内粒子产生的电势。

Interaction list 对第l层第i个方形区域，这指的是其父区域的近邻的子区域中，与它well-separated的部分。（fig5）

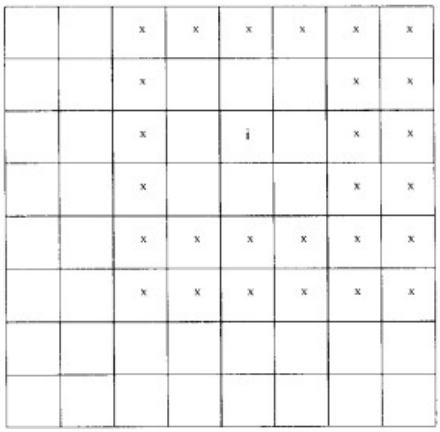

FIG. 5. Interaction list for box *i*. Thick lines correspond to mesh level 2 and thin lines to level 3. Boxes marked with an "x" are *well separated* from box *i* and contained within the nearest neighbors of box *i*'s parent.

假定我们已经得到了第l-1层的所有局部展开$\Psi_{l-1,i}$，于是应用lemma3.2.5可以将展开式$\Psi_{l-1,i}$平移至各个区域i的子区域中心，于是我们有，对任意第l层的区域j，我们获得了其父区域近邻外的所有粒子产生的电势展开式$\tilde{\Psi}_{l,i}$.因此需要在$\tilde{\Psi}_{l,i}$中加上Interaction list那部分才能获得$\Psi_{l,i}$。这一步需要应有lemma3.2.4将Interaction list 区域中的多级展开转化为关于当前区域的局部展开并将其加至先前由其父区域得到的展开式上。注意到对于free-space问题，$\Psi_{0,i}$与$\Psi_{1,i}$都是0因为它们都没有well-separated的部分,于是我们可以从第二层开始构建局部展开。下面是算法的一个流程描述。

ALGORITHM.

Initialization

取细分的层数$n \approx \log_4 N$，精度ϵ, 并设$p \approx \log_2(\epsilon)$

Upward Pass

Step 1

Comment [在最下层，关于各个区域中心求出内部粒子产生电势的多级展开]

do $ibox = 1, \ldots, 4^n$

应用Theorem 2.1求出p阶多级展开$\Phi_{n,ibox}$

enddo

Step 2

Comment [求出关于更上层区域中心的多级展开，每个展开式表示一个区域中各个粒子产生的电势场]

do $l = n-1, \ldots, 0$

do $ibox = 1, \ldots, 4^l$

应用Lemma3.2.3将子区域的p阶多级展开平移至当前父区域中心并将它们相加，得到$\Phi_{l,ibox}$

enddo

enddo

Downward Pass

Comment [在这一部分，相互作用总是在尽可能高的层上考虑。给定一个区域，这一步通过将well-separated部分和那些未被计算进来的父区域相加完成。]

Step 3

Comment [在每个第l层（$l \leq n-1$），关于各个区域中心构建局部展开。这部分局部展开描述的电势是由所有在当前区域及其近邻外的粒子产生的。一旦局部展开被求出，就将其平移至下一层对应子区域的中心，此后这些展开式将作为下一轮循环的父展开式。]

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

do $l = 1, \ldots, n-1$

do $ibox = 1, \ldots, 4^l$

利用lemma3.2.4将$ibox$的Interaction list中的多级展开$\Phi_{l,j}$ 转化为当前区域的局部展开，将他们各个相加，最后加上$\tilde{\Psi}_{l,ibox}$得到$\Psi_{l,ibox}$

enddo

do $ibox = 1, \ldots, 4^l$

应用lemma3.2.5将$\Psi_{l,box}$扩展至其子区域的中心以获得$ibox$各个子区域的展开式$\tilde{\Psi}_{l+1,j}$

enddo

enddo

Step 4

Comment [求出最下层的相互作用]

do $ibox = 1, \ldots, 4^n$

应用lemma3.2.4将区域j的Interaction list 内的区域的多级展开$\Phi_{l,j}$转化为对应中心的局部展开，将它们相加，最后加上$\tilde{\Psi}_{l,ibox}$以求出$\Psi_{l,ibox}$

enddo

Comment [现在，最下层的局部展开已经可用了。它们可以用以计算某个区域近邻外的所有粒子产生的电势或力场。]

Step 5

Comment [在各个粒子的位置计算局部展开]

do $ibox = 1, \ldots, 4^n$

对每个为于$ibox$内z_j点处的粒子p_j，计算$\Phi_{n,ibox}(z_j)$

enddo

Step 6

Comment [直接计算近邻区域内的电势或力场]

do $ibox = 1, \ldots, 4^n$

对每个位于区域$ibox$内的粒子p_j，计算它与各个位于ibox及其近邻内粒子的相互作用。

enddo

Step 7

do $ibox = 1, \ldots, 4^n$

对每个位于区域$ibox$内的粒子,将直接计算的部分与远场域部分相加。

enddo

Remark. 每个局部展开式都由p阶多项式的系数来描述。在某个点直接求多项式的值表示电势。但是，根据lemma3.2.1,力可以由可解析的导数得到，因此不必进行数值微分。更进一步，由于Φ'的解析性，力的误差界就与(3.2.4)、(3.2.11)和(3.2.15)有完全相同的形式。

接下来是一个简单的算法复杂性的分析（List 1）

于是运行时间大约为

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

其中a，b，c，d，e都是由计算机系统、语言、实现等决定的常数。

除了渐进时间复杂度，渐进存储空间也是一个数值进程的重要特征。该算法需要$\Phi_{l,j}$和$\Psi_{l,j}$ 被存储，同时还要存点的位置、带电量以及计算的结果（电势或静电力）.介于每层中各区域都有一对p阶多项式Φ 和Ψ与之相关，并且所有其他存储的向量都与N成比例，于是可得算法的渐进存储空间有如下形式

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

或

$$(\alpha - \beta\log_2(\epsilon)) \cdot N$$

其中系数α, β,如前文所说，由计算机系统、语言、实现等决定。在我们的数值实验中，实际的存储空间为(单精度数)

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

Step	Operation count	Explanation
1	order Np	Each particle contributes to one expansion at the finest level.
2	order Np^2	At the lth 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.

List 1

Remark. 显然，我们在Step 6的复杂度评估中假定一个适当均匀的粒子分布。如果粒子分布高度不均匀，那么我们只需要将粒子数多的区域进行细分。尽管这种描述更复杂，一种自适应的方法需要同时保持精度与计算速度（见[3]）.

3.4 边界条件

一系列的边界条件被应用在做粒子模拟的时候，包括周期性边界条件、齐次的Dirichlet或Neumann条件，含有一些混合条件。我们先详细讲周期边界条件，再将Dirichlet条件，最后简单讨论一下其他条件。

1. 周期性边界条件

我们先考虑fig5中的计算区域。在upward pass 最后，我们求出了一个对整个计算区域的净多级展开

$$\Phi_{0,1}(z) = \sum_{k=1}^{p} \frac{a_k}{z^k} \tag{3.4.1}$$

这就是各个区域中各循环部分关于其中心的展开式。这些循环部分除了fig3中的都被计算区域自己well separated，并且它们诱导的场就是p阶局部多项式，因此$p \approx -\log_2(\epsilon)$就是满足精度$\epsilon$所需的条件。我们假设周期性粒子模型没有净电荷，于是根据lemma3.2.4获得的局部表示可以被写成

$$\Psi_{0,1} = \sum_{m=1}^{p} b_m \cdot z^m \tag{3.4.2}$$

其中

$$b_m = \frac{1}{z_0^m} \sum_{k=1}^{p} \frac{a_k}{z_0^k} \binom{m+k-1}{k-1} (-1)^k \quad with\ m = 0, 1, \ldots, p \tag{3.4.3}$$

其中z_0是先前循环区域的中心。

Remark. 在特定问题下（例如宇宙学），计算区域显然无法做到没有净电荷（质量）。这个条件对于电势能被良定义是必要的，因为对数部分在$z_0 \to \infty$是无界的。但是力的计算有可能仍有意义。实际上，用算法中的记号$\Phi_{l,i}, \Psi_{l,i}, \tilde{\Psi}_{l,i}$表示解析的电势展开式，因此它们的导数也是解析函数。更进一步，从Theorem3.2.1得知，导数$\Phi'_{l,i}$是纯逆幂级数。因此根据lemma3.2.1可以用相同的算法结构来计算各处的力场，而绕开对数部分带来的困难。唯一的区别就是初始的展开式是多极展开的导数而不是原来的展开式本身。

注意到，计算区域中well separated的部分是以z_0 为中心的且z_0有整的实部与虚部，其中$Re(z_0) \geq 2$或$Im \geq 2$。设S为这部分中心点的集合。为了计算所有well-separated部分产生的场,我们通过对所有$z_0 \in S$累加平移后的形如(3.4.3) 所示的局部展开系数来获得如下结果

$$b_m^{total} = \sum_{k=1}^{p} a_k \binom{m+k-1}{k-1} (-1)^k \left(\sum_S \frac{1}{z_0^{m+k}} \right) \tag{3.4.4}$$

关于S的各个z_0的逆幂部分可以被预先计算和存储。对于$(m+k) > 2$级数显然收敛。但对$(m+k) \leq 2$，级数不绝对收敛，计算的值取决于加法的顺序。选取一个合适的累加序列需要基于对物理模型的考虑。

首先假定模拟中仅有一个单位点电荷为于原点。那么周期性的模型就是一个均匀的电荷晶格并且牛顿第三定律要求每个粒子上净力为零。但是位于原点的粒子的净力对应于$1/z_0$在S上的累加，于是我们令

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

为了确定二阶项的值

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

我们假定模拟中唯一的粒子是沿x轴方向，位于原点的单位偶极子。于是该模型又是一个均匀晶格并且等价点$(\frac{1}{2}, 0)$和$(-\frac{1}{2}, 0)$见电势差一定为零，就是说

$$\Phi_{(\frac{1}{2},0)} - \Phi_{(-\frac{1}{2},0)} \equiv \delta\Phi = 0 \tag{3.4.5}$$

位于z_0的单个偶极子对电势差$\delta\Phi$造成的影响为

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

因此我们发现原点偶极子造成的电势差为-4。对一个z_0处的偶极子，$|z_0| \geq 1$,我们可以将上述影响扩张至$\delta\Phi$为

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

现在令S'为所有循环区域的中心，就是说S'表示所有实部和虚部为整数的点z_0，除了原点。于是

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

一个有点复杂的计算表明

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

因此，为了满足(3.4.5)，我们令

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

于是

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

而累加$\sum_{S' \setminus S}(1/z_0^2)$易得为零，因此我们有

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

于是在S上z_0的逆幂累加就被定义了。

将整个计算区域的多级展开$\Phi_{0,1}$转化为局部展开$\Psi_{0,1}$的过程描述了由所有well-separated 区域引起的电场，引用算法中的记号，可以被写作

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

其中T是一个p乘p的常矩阵，由如下方法定义

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

这可以被视为周期性条件的算法downward pass 中的第一步。就此，我们以及考虑了所有相互作用，除了与计算区域直接相邻的那一部分，如fig3所示。但位于计算区域内的展开式$\Phi_{l,i}$ 同样也是其近邻区域对应的展开式。通过将合适的区域加入interaction list，我们保持了原有的计算结构以及相关的计算复杂度。

2. *Dirichlet*边界条件

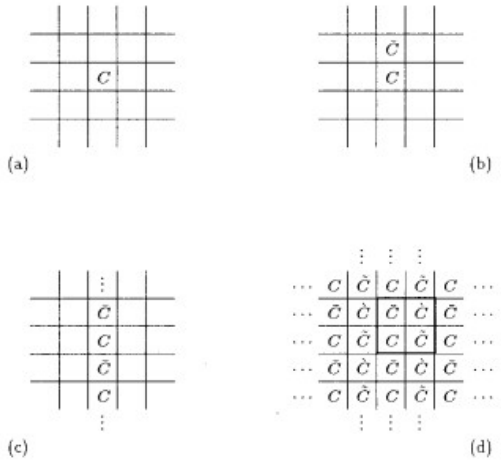

FIG. 6. The computational cell centered at the origin is represented by C. The cell $\overline{C}$ is the image of cell C reflected across the top boundary, with corresponding particles assigned charges of opposite sign. The cell $\tilde{C}$ is the image of cell C reflected across the left boundary, again with corresponding particles assigned charges of opposite sign. The cell $\hat{C}$ 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).

我们现在考虑齐次Dirichlet条件，也就是

$$\Phi(x, y) = 0 \quad (x, y) \in \partial D$$

其中∂D是计算区域的边界。从分析上讲，这可以通过镜像法来完成，之后将会详细讲。一般的，我们考虑电势差

由两部分构成，就是说

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

其中$\Phi_{sources}$表示计算区域内粒子产生的电势，Φ_{images}表示某些特定位于计算区域外的像区域电荷引起的电势。这些像点电荷与强度依照下面条件选取

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

对于我们考虑计算区域，合适的像点电荷可以通过迭代过程选取，见fig6.我们首先将每个带电量为σ_i 的电荷p_i通过上面的边界做一次反射，并在对应位置放一个带电量为$-\sigma_i$的电荷，形成一个记为$\bar{C}$的像区域(fig6b)。像点电荷集记为V_1，产生的对应电势记为Φ_{V_1}。将Φ_{V_1}加在$\Phi_{sources}$ 上显然可以使上边界满足条件。为了在下边界满足条件，我们需要向下反射所有电荷（包括原点和像点），产生两个新像区域（C和$\bar{C}$的复制）.经过这样的第二次反射后的电荷记为V_2。现在$\Phi_{sources} + \Phi_{V_2}$满足下边界的条件。我们因此再次向上反射一次，在加入两个新像区域以及新的像点电荷集记为V_3，于是$\Phi_{sources} + \Phi_{V_3}$重新满足上边界条件但又违反了下边界条件。通过上述迭代，我们构成了一个点集列$\{V_i\}$，满足

$$V_1 \subset V_2 \subset V_3 \ldots V$$

其中$V = \cup_{i=1}^{\infty} V_i$是包含无穷像区域的一个集合，见fig6c。易得，对应的电场序列$\{\Phi_{V_i}\}$在计算区域内收敛并且$\Phi_{sources} + \Phi_V$同时满足上下两个边界条件。

为了在另两边满足Dirichlet条件，我们进行一个类似地过程。首先，我们将所有电荷向左边界反射，这样显然不影响上下边界，同时使左边界满足了边界条件。现在，将所有像点电荷记为H_1.向右反射构造H_2，其中$\Phi_{sources} + \Phi_{H_2}$满足右边界的边界条件。重复上述反射过程得到一个无限集序列$\{H_i\}$满足

$$H_1 \subset H_2 \subset H_3 \subset \ldots H$$

其中$H = \cup_{i=1}^{\infty} H_i$ 是包含于二维像区域族中的点的集合，见fig6d。易得序列$\{\Phi_{H_i}\}$收敛于计算区域内，我们记其极限为Φ_H，最终观察到$\Phi_{sources} + \Phi_H = 0$满足所有边界条件。

从计算的角度，反射法的收敛率很不让人满意。但是结合了我们的算法，可以将它转化为极有效的数值工具。用前面的术语说，所有像区域除了计算区域的近邻部分外都well-sperated并且它们产生的场可以用一个局部展开$\Psi_{0,1}$表示。一旦计算了这种局部扩展的系数，我们只需要考虑计算单元本身最近邻内的相互作用。要做到这一点，就像在周期情况下一样，我们只需将适当的像区域添加到计算区域的Interaction list的中。

因此，仍只需计算$\Psi_{0,1}$。我们首先观察到像平面有以$(\frac{1}{2}, \frac{1}{2})$为中心的“supercell”,在fig6d中用粗线划出。但是通过上述周期问题的方法，我们可以得到一个关于$(\frac{1}{2}, \frac{1}{2})$的展开式，表示supercell的近邻外的所有相互作用。这个展开式可以用lemma3.2.5转化成关于原点的展开式，记为$\tilde{\Psi}_{0,1}$。现在还需要计算那些supercell近邻内的well-separated的区域。一共有27 个这样的区域，并且它们的多极展开可以被转化为关于原点的局部展开。将它与$\tilde{Psi}_{0,1}$相加得到$\Psi_{0,1}$

3. 其他边界条件

在特点应用下，我们用周期或Dirichlet条件，同时在另外情况时，Neumann或混合条件也被使用。一个混合边界条件的典型例子就是在计算区域两条相对的边上用到Neumann条件，在另外两条边上用Dirichlet条件。一些其他模型要求左右两边满足周期边界条件，上下满足Dirichlet或Neumann条件。这些条件的问题可以通过一种与上述过程一样的方法来解决。通过反射或周期延拓，先形成全屏幕的像。局部展开$\Psi_{0,1}$通过累加合适的well-separated部分获得，处理剩余的像点的相互作用如上文所示那般处理。

3.5 数值实验结果

电脑程序已能实现上述算法并能解决free-space问题以及包含周期性边界条件和齐次Dirichlet或Neumann边界条件的问题。

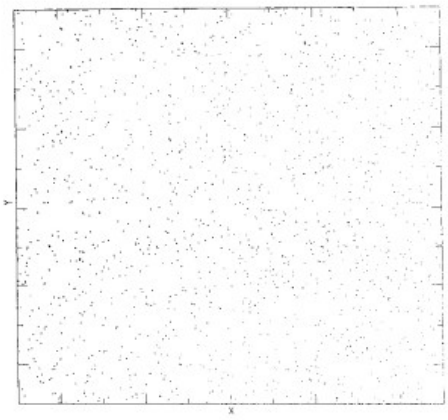

FIG. 7. 1600 randomly located charges in the computational cell.

为了测试目的，我们随机在计算区域内分配了一些点电荷（fig7）,带电量在0至1之间，数量在100至12800 之间。计算在VAX-8600上进行，展开式的阶数取为20，就是说有大约5位数精度。在各个例子中，我们通过三种算法进行计算：（1）通过上文中的算法用单精度运算；（2）直接用单精度计算(3.2.7)；（3）直接用双精度计算(3.2.7)。前两个计算用以比较我们算法与直接计算的计算速度与精度。双精度的直接计算用于作为前两个算法的相对误差的参照。在所有情况下都对周期模型进行了计算，周期性边界条件的方法在节3.4中提及。

数值实验的结果记录在Table 1中。第一列表示粒子数N，第二三列表示应用本文算法所需CPU时间T_{alg} 以及最大的相对误差δ_{alg}。第四五列表示直接计算所需的CPU时间T_{dir}与误差δ_{dir}

Remark. 对于12800个点的例子，本文算法耗时1分钟。而直接计算现得不切实际因为总共花费了约5小时也没得到什么有用的信息。因此我们仅直接计算了其中100个点，同时有单精度和双精度，并用以估计精度δ_{alg}和δ_{dir}。T_{dir}是通过缩放估得的。

TABLE I

Computational Results

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

通过Table 1可以有如下结论

1. 结果的误差如(3.2.4)、(3.2.11)和(3.2.15)预测的一致。没有发现什么截断误差带来的精度问题。

2. 尽管有些飘忽不定，但计算时间随粒子个数线性增长。

3. 对于1600个粒子的模型，直接计算的耗时约为本文算法的40倍。对12800个粒子的情况更是达到了将近300倍。

我们也进行了类似的关于齐次Dirichlet和Neumann边界的计算，结果与上述例子一致。

为了说明情况，我们在fig8中画出了包含10个随机电荷的Dirichlet边界条件的等点位线。总耗时15秒，其中半数时间用于计算超过10000个点的数值，而剩余一般用于画图。

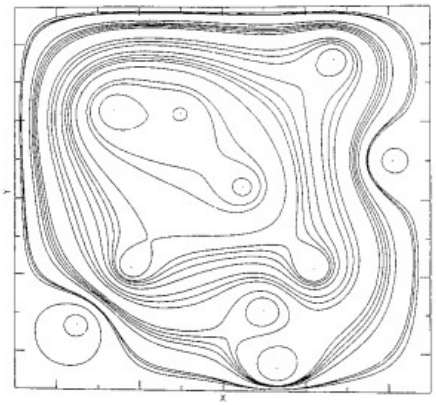

FIG. 8. The equipotential lines for the electrostatic field due to 10 randomly located charges in the computational cell, with homogeneous Dirichlet boundary conditions.

3.6 总结

我们构建了一种用于快速计算由粒子产生场的算法，类似问题在电磁物理学、分子动力学、流体力学（涡旋法）和天体力学中。该算法用动态模拟或Monte Carlo模拟都可行，只要所求场是类似自然界中库伦力的。本文算法的渐进CPU时间是$O(N)$阶的，其中N是模拟中的粒子数。并且节3.5中给出的数值例子表示即使对很大的粒子数，计算时间也是可以接受的。在本文在，我们呈现了二维的例子，可以很直接地将结果推广至三维，这将在将来汇报。

4 外文原文

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,

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

where m_i is the mass of the ith particle and the force is obtained from the gradient of a potential function Φ. 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 Φ 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 Φ_{near} (when present) is a rapidly decaying potential (e.g., Van der Waals), Φ_{external} (when present) is independent of the number of particles, and Φ_{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 Φ_{near} requires an amount of work proportional to N, as does the calculation of Φ_{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

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

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

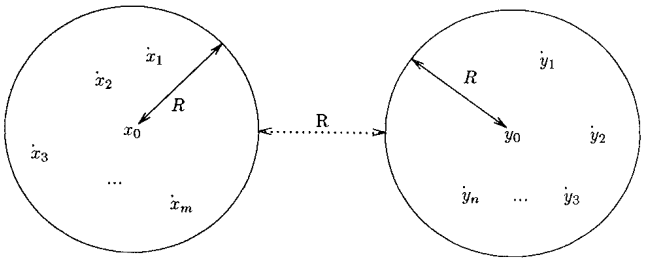

FIG. 1. Well-separated sets in the plane.

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

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

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 ε (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 $\{b_1\}$ 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.

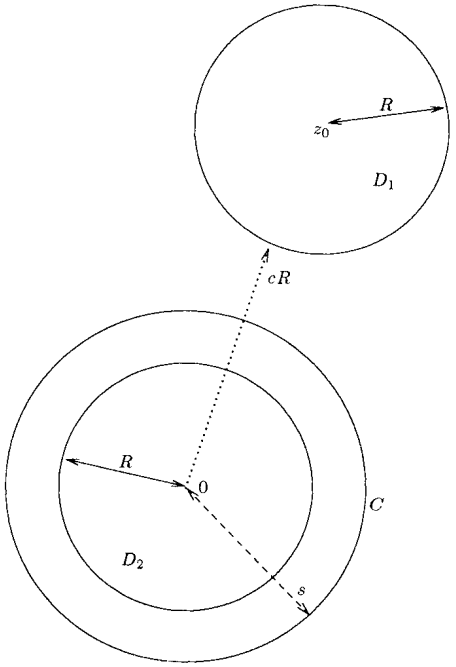

FIG. 2. Source charges $q_1, q_2, ..., q_1$ are contained in the circle D_1. The corresponding multipole expansion about z_0 converges inside D_2. C is a circle of radius s, with $s > R$.

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-

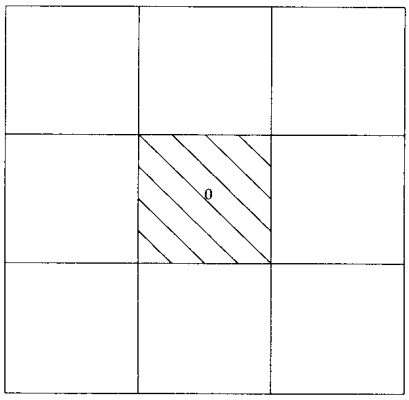

FIG. 3. The computational box (shaded) and its nearest periodic images. The box is centered at the origin "0" and has area one.

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 ε, 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

Φ_{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.

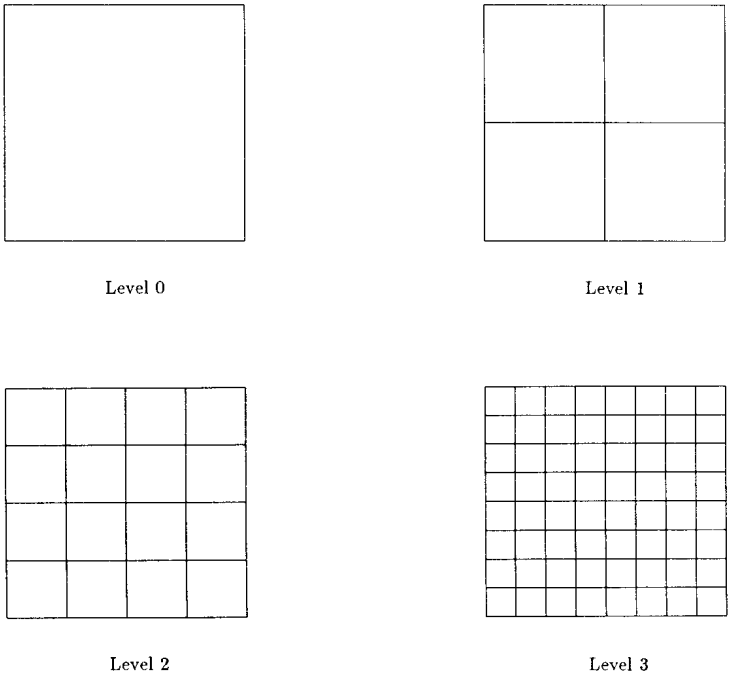

FIG. 4. The computational box and three levels of refinement.

$\tilde{\Psi}_{l,i}$ 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

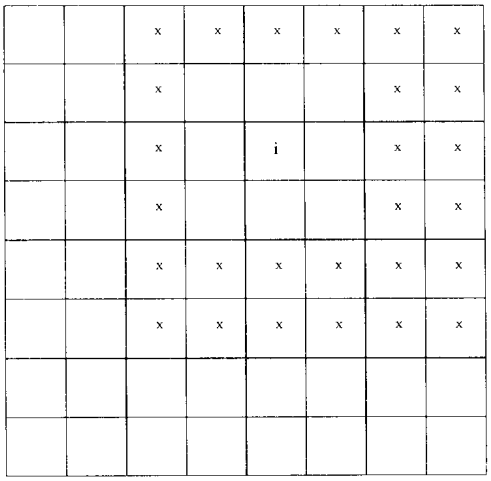

FIG. 5. Interaction list for box i. Thick lines correspond to mesh level 2 and thin lines to level 3. Boxes marked with an "x" are *well separated* from box i and contained within the nearest neighbors of box i's parent.

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 ε, 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 Φ', 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 lth 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, Φ and Ψ, 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 α and β 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 ε. 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 ∂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,

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

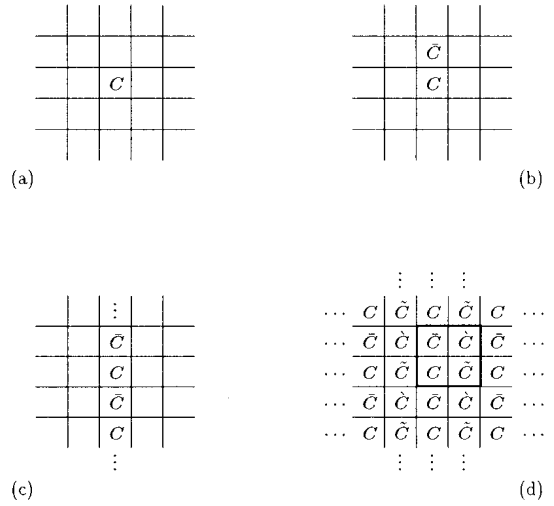

FIG. 6. The computational cell centered at the origin is represented by C. The cell $\bar{C}$ is the image of cell C reflected across the top boundary, with corresponding particles assigned charges of opposite sign. The cell $\tilde{C}$ is the image of cell C reflected across the left boundary, again with corresponding particles assigned charges of opposite sign. The cell $\grave{C}$ 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 Φ_{sources} is the field due to the particles inside the computational cell and Φ_{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 σ_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 $\bar{C}$ (Fig. 6b)). The set of image charges is denoted by V_1, and the field they induce is called Φ_{V_1}. Adding Φ_{V_1} to Φ_{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 $\bar{C}$). 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 Φ_H. Finally, we observe that $\Phi_{\text{sources}} + \Phi_H = 0$ on the entire boundary ∂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

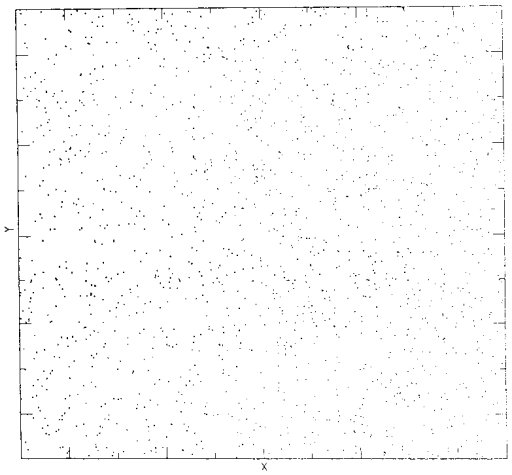

FIG. 7. 1600 randomly located charges in the computational cell.

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_{alg} that were required by the algorithm of the present paper to obtain the fields at all N particles, and the greatest relative error δ_{alg} obtained at any of the particles, respectively. Columns 4 and 5 contain the CPU times T_{dir} that were required by the direct algorithm (2.7) to obtain the fields at all N particles, and the greatest relative error δ_{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 δ_{alg} and δ_{dir}. The value for T_{dir} in this case was estimated by scaling.

The following observations can be made from Table I:

TABLE I

Computational Results

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

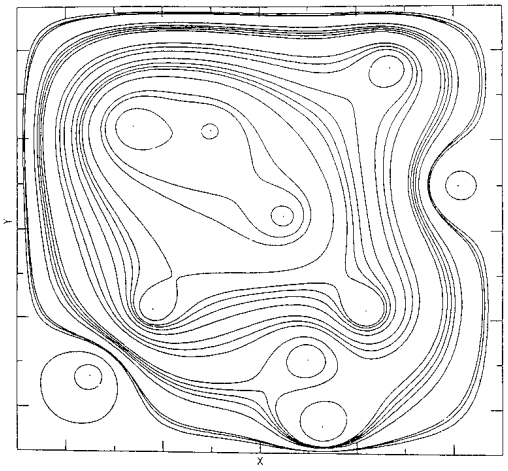

FIG. 8. The equipotential lines for the electrostatic field due to 10 randomly located charges in the computational cell, with homogeneous Dirichlet boundary conditions.

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).

参考文献

[1] Rick Beatson,Leslie Greengard.A short course on fast multipole methods.Wavelets, Multilevel Methods and Elliptic PDEs,1997,1-37.