On the Accuracy of Poisson's Formula Based N-Body Algorithms

On the Accuracy of Poisson's Formula Based N–Body Algorithms

Y. Charlie Hu
S. Lennart Johnsson

TR-06-96

January 1996

Parallel Computing Research Group

Center for Research in Computing Technology
Harvard University
Cambridge, Massachusetts

Submitted to *SIAM Journal on Scientific Computing*.

ON THE ACCURACY OF POISSON'S FORMULA BASED N–BODY ALGORITHMS*

Y. CHARLIE HU† AND S. LENNART JOHNSSON‡

Abstract. We study the accuracy–cost tradeoffs of a Poisson's formula based hierarchical N–body method. The parameters that control the degree of approximation of the computational elements and the separateness of interacting elements, govern both the arithmetic complexity and the accuracy of the method. Empirical models for predicting the execution time and the accuracy of the potential and force evaluations for three-dimensional problems are presented. We demonstrate how these models can be used to minimize the execution time for a prescribed error and verify the predictions through simulations on particle systems with up to one million particles. An interesting observation is that for a given error, defining the near–field to consist of only nearest neighbor elements yields a lower computational complexity for a given error than the two–element separation recommended in the literature. We also show that the particle distribution may have a significant impact on the error.

Key words. N–body simulation, multipole algorithms, hierarchical N–body methods, numerical simulation

AMS subject classifications. 65C99, 65G99, 70–08, 70F10

1. Introduction.

Although $O(N)$ N–body methods [1, 8, 7, 6, 21, 20] are superior to the direct $O(N^2)$ methods in terms of arithmetic complexity for large scale simulations, they have yet to gain widespread acceptance. There are several reasons for this. First, though the break–even point compared to the direct method occurs at a few tens of thousands of particles in three dimensions, significant gains is only achieved for systems of a hundred thousand particles or more when good accuracy is required. For large particle systems, efficient parallel implementations are required both with respect to memory and processing speed even if the arithmetic complexity grows in proportion to the number of particles. Thus, one issue that is limiting their rapid acceptance is their complex computational structures relative to the direct method amplified by the need for efficient parallel computer implementations. Another important reason for the slow acceptance of the $O(N)$ methods is the limited understanding of the error behavior of such methods, especially of the accuracy–cost tradeoffs with respect to the parameters that control the errors and the arithmetic complexity. A third limiting factor is that further algorithmic development is necessary to cover the range of boundary conditions encountered in applications. Most algorithmic developments and parallel computer implementations have focused on applications with free space boundary conditions.

Like the multi–grid methods [3], the multipole–based N–body methods, or their variations, are approximation methods. For all these methods higher accuracy can be achieved at increased computational effort. Though the computational time is directly proportional to the number of particles, the constant of proportionality depends upon the desired accuracy.

*This work was supported Supported by the Air Force Office of Scientific Research through grants F49620-93-1-0480 and F49620-96-1-0289.

†Aiken Computation Lab, Harvard University, 33 Oxford Street, Cambridge, MA 02138 (hu@das.harvard.edu).

‡Department of Computer Science, University of Houston, Houston, Texas 77204-3475 (johnsson@cs.uh.edu)

There are four sources of errors common to all the multi–pole like $O(N)$ N–body methods:

1. Approximation errors in representing computational elements that represent the aggregate effect of clusters of particles by finite series expansion. Greengard and Rokhlin (GR) [6, 7], and Zhao [21] give error bounds on truncated multipole expansions for individual computational elements as a function of the number of terms in the expansions. Similarly, Anderson [1] provides some insights into the error characteristics of methods based on Poisson's formula, but no rigorous bounds. Clearly, the degree of approximation directly affects the arithmetic complexity in evaluating element interactions, and therefore that of the complete methods.
2. The definition of the near–field (and hence the far–field), i.e., the separateness of interacting computational elements, determines the numbers of near–field and interactive–field computational elements. The hierarchical evaluation of interactive–field computational elements and the direct evaluation in the near–field at the leaf–level are two computation dominating stages of $O(N)$ N–body methods. The degree of separateness also determines the base constant of the exponentially growing error bound.
3. With near–fields consisting of two or more computational elements in each coordinate direction, the computational effort can be reduced through the use of supernodes, i.e., the interactions of all child elements of a node with a given element is replaced by the interaction of the parent node with that element, or conversely, the interaction of a given element with all child elements of a node is replaced by the interaction of the given element with the parent element. Supernodes introduces additional errors.
4. Finite machine precision leads to round–off error. The sensitivity to the round–off error in the direct evaluation in the near–field and in the hierarchy traversal is not necessarily the same.

Note that the direct method only suffers from round–off errors, while the contributions to the total error in the hierarchical methods are quite complex. For instance, the evaluation of the near–field potentials and/or forces via the direct method is subject only to round–off error, while the error in the evaluation of the far–field potentials and/or force via hierarchy traversal is due to all issues raised above. Thus, varying the hierarchy depth changes the proportion of particles in the near–field and in the far–field, and therefore may affect the error of the hierarchical methods. The hierarchy depth also directly affects the total number of arithmetic operations.

In using the Poisson's formula based method introduced by Anderson [1], sphere integration introduces one more source of error.

The main contribution of this paper are empirical models for the error as a function of the approximation parameters of the Poisson's formula based nonadaptive method by Anderson. We demonstrate how to use these models to minimize the execution time for a desired accuracy and illustrate the sensitivity of the models to particle distribution. Our study shows that using near–fields consisting of only nearest neighbor computational elements minimizes the number of arithmetic operations (and execution time) for a given accuracy.

We devised some simple model distributions in studying the impact of the particle distributions on the error. Our experiments show that the accuracy may degrade significantly for nonuniform distributions. We observed an accuracy degradation by a factor of six for one variation of the uniform distribution. This range is an illus-

tration of the impact of the particle distribution, not a bound on the range. The more nonuniform the particle distribution is with respect to the center of the spheres, the less accurate the method will be. Enlarging the outer spheres, which intuitively will smooth out the integrations, improves the accuracy. With respect to the radii of spheres used in the integration required in using Poisson's formula for the computational elements, we numerically verify the optimum choices given by Anderson for uniform particle distributions.

The empirical study reported in this paper was performed on the Connection Machine system CM–5E using a data–parallel implementation of Anderson's method [10]. Using an efficient implementation of the direct method, we are able to verify the accuracy and the accuracy–cost tradeoffs of Anderson's method via simulations of systems of up to one million particles within reasonable execution time. To our knowledge, this is the first numerical verification of the accuracy of Anderson's method.

The paper is organized as follows. Section 2 describes the parameters common to all multipole–like $O(N)$ N–body methods. Section 3 details the computational elements of Anderson's multipole method. Section 4 describes the simulation environment and defines how the simulation errors are determined. Section 5 discusses the accuracy–cost tradeoffs of Anderson's method, and Section 6 presents experimental results on the choices of sphere radii in Anderson's method and their impact on the error for small variations of the uniform particle distribution. Section 7 summarizes the paper.

Previous work. Table 1.1 lists the parameter choices used in some previous implementations of multipole–like methods. All simulations but the ones by Greengard–Rokhlin used uniform distributions. Greengard and Rokhlin [6], and Zhao [21] numerically examined the accuracy of their respective methods as a function of the number of terms retained in the multipole expansions, and the number of particles. Schmidt and Lee [16] examined the accuracy and the execution time for a few combinations of hierarchy depth and number of terms in the multipole expansions for a 3–D GR method with one–separation (i.e., the near–field consists of only nearest neighbor elements). Shimada et al. [17] examined a few combinations of hierarchy depth, one–separation and two–separation (i.e., the near–field consists of nearest neighbor elements and their nearest neighbors as well). Using a shared–memory implementation of the GR method, Leathrum [13] numerically examined the tradeoffs between the accuracy and the execution time for six combinations of the degree of separation and the use of supernodes, namely, one–separation, two–separation, one–two–separation (two–separation for the leaf–level and one–separation for the rest of the hierarchy), and the same simulations using supernodes. The simulations were performed for only 10,000 particles with a hierarchy depth of three for all simulations. The simulations showed that two–separation with supernodes gives the smallest execution time for any specified error less than 0.0002, while one–separation without supernodes gives the smallest execution time for errors between 0.0002 and 0.0005. One–separation with supernodes gave significantly worse execution time than the other approximations for a wide range of errors. Esselink [5] studied the arithmetic complexity, accuracy and execution times on uniprocessors for the GR method, Appel's method, a combinations of the two, and Ewald's summation method. Limited by the memory, he performed simulations for up to 92,762 particles with hierarchy depths two, three, and four.

2. Parameters in multipole–like hierarchical N–body methods. Multipole–like hierarchical methods [1, 2, 6] for the N–body problem partition the potentials

TABLE 1.1
Characteristics of some previous implementations of multipole–like $O(N)$ N-body methods.

Author	Method	Degree of separation	Use of supernodes	Hierarchy depth
Greengard–Rokhlin [7, 9]	GR	2	No	$\lfloor \log_8^N \rfloor$
Zhao [21]	Zhao	2	Yes	$\lfloor \log_8^N \rfloor$
Schmidt–Lee [16]	GR	1	No	3,4,5
Shimada et al. [17]	GR	1,2	No	$\log_8^N$
Leathrum [13]	GR	1,2,mixed 1 and 2	Yes and No	$\lfloor \log_8^N \rfloor$
Esselink [5]	GR	2	Yes	2,3,4

into two parts:

$$\phi_{total} = \phi_{near-field} + \phi_{far-field}, \tag{2.1}$$

where $\phi_{near-field}$ is the potential due to nearby particles and $\phi_{far-field}$ is the potential due to faraway particles. The near–field is evaluated through the classical N–body technique of pairwise interactions, while the far–field is evaluated hierarchically.

The multipole–like $O(N)$ methods share the same computational structure, but differ in the computational elements they use. There are four parameters in these methods that govern both the errors and the arithmetic complexity: the truncation error in representing computational elements by finite series expansions, the degree of separation of the near–field, the use of supernodes, and the hierarchy depth. The later three controls the computational structure. In the typical error bounds $O(c^{-p})$ of $O(N)$ methods, p is the *error decay rate* (related to the truncation error), and the base constant c is controlled by the separateness of interacting computational elements.

2.1. Parameters in computational elements. There are two key ideas in hierarchical methods that lead to reduced arithmetic complexity. The first is to represent a cluster of particles sufficiently far away from an evaluation point by a single computational element, called *far–field potential representation*. The exact representation of computational elements would require an infinite number of terms in multipole expansions or in Poisson's formula (Anderson), and hence, in practice must be truncated to a finite number of terms. The $O(N)$ methods also introduces a *local–field potential representation* – a second kind of computational element. This element approximates the potential field in a "local" domain due to particles in the far domain.

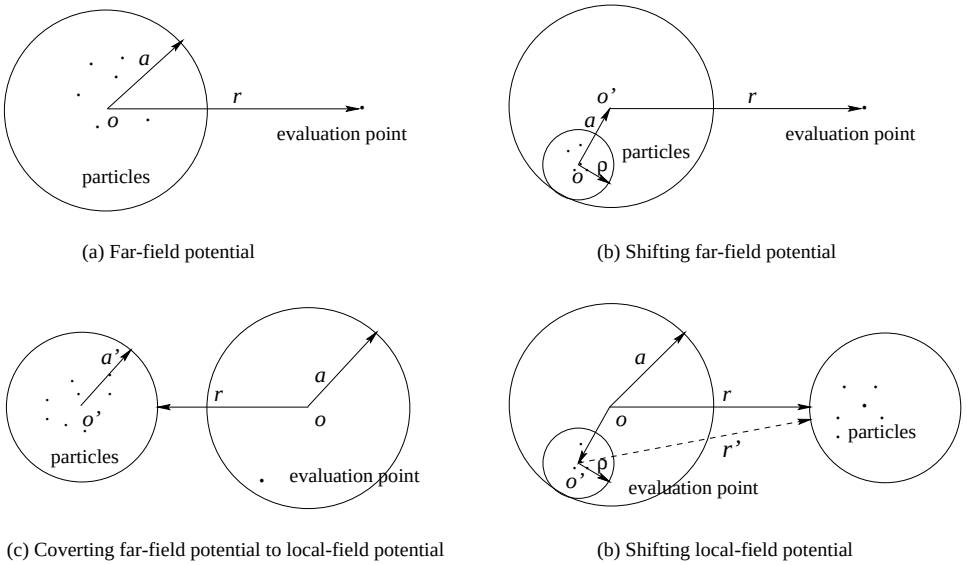
Table 2.1 summarizes the computational elements in Barnes–Hut's [2] $O(N \log_2 N)$ method, and the $O(N)$ methods by GR [6], Zhao [21], and Anderson [1], and the coordinate systems used. The relationship between the truncation error, or error decay rate, p in the methods by GR and by Zhao, and the integration order and number of integration points K in Anderson's method is discussed in Section 2.4.

In $O(N)$ methods, three kinds of translations on the computational elements are performed during the hierarchy traversal, as shown in Figure 2.1(b)–(d). In all implementations so far, the translations are between far–field potentials or local–field potentials with the same truncation order. The error bounds of the three translations given below for multipole–based methods are proved by Greengard–Rokhlin [8, 7], and Zhao [21], and are summarized in Table 2.2. The errors of the translations in Anderson's method are expected to have similar form [1], and is described in detail in Section 3.

- T_{F2F}: shifting a far–field potential representation centered at o with radius ρ to a new center o' at a distance a away, as shown in Figure 2.1(b). In

TABLE 2.1
Computational elements of four hierarchical methods. p denotes the number of terms retained in the GR and Zhao's methods, while K denotes the number of integration points in Anderson's method.

Method	Dim	Computational Element	Coordinate	Translation Cost $f(p)$
Barnes–Hut	2–D/3–D	center of mass	Cartesian	
Greengard–Rokhlin	2–D	multipole/local expansions	Complex	$O(p^2)$, $O(p\log p)$ with FFT
	3–D	multipole/local expansions	Spherical	$O(p^4)$, $O(p^2\log p)$ with FFT
Zhao	3–D	multipole/local expansions	Cartesian	$O(p^6)$
Anderson	2–D	outer/inner ring approximations	Cartesian	$O(K^2)$
	3–D	outer/inner sphere approximations	Cartesian	$O(K^2)$



(a) Far-field potential

(b) Shifting far-field potential

(c) Coverting far-field potential to local-field potential

(b) Shifting local-field potential

FIG. 2.1. *The three translations in $O(N)$ hierarchical N–body methods.*

multipole–based methods, the coefficients of the new p–term multipole expansion can be computed *exactly* in exact arithmetic from the coefficients of the original p–term multipole expansion. Thus, no approximation error is introduced, only round–off errors. The relative error in evaluating the new multipole expansion at an evaluation point at distance r from the new center is bounded by $(\frac{\rho+a}{r})^p$.

- T_{F2L}: converting a far–field potential representation centered o' to a local–field potential representation centered at o at a distance r away from o', as shown in Figure 2.1(c). Each coefficient in the local expansion depends on all coefficients of the far–field multipole expansion, and therefore cannot be computed exactly from a finite term far–field expansion, even in exact arithmetic. The truncation of the far–field representation induces an error in local–field expansion. If the coefficients of the p–term local expansion were computed exactly, i.e., computed from an infinite far–field expansion with no

TABLE 2.2
Three translations and their error bounds in multipole–based methods.

Translation	Coefficients in p–term to p–term translation	Relative error
T_{F2F}	exact	$(\frac{\rho+a}{r})^p$
T_{F2L}	not exact	$(\frac{a}{r})^p$ if coefficients are exact
T_{L2L}	exact	$(\frac{\rho}{r'})^p$ if coefficients in T_{F2L} are exact

round–off error, then the relative error from evaluating the local–field at any evaluation point inside a sphere of radius a would be bounded by $(\frac{a}{r})^p$.

- T_{L2L}: shifting a local–field potential representation centered at o to be centered at o', as shown in Figure 2.1(d). The coefficients of the new p–term local expansions can be computed *exactly* from the coefficients of the original p–term local expansions in the absence of round–off errors. No approximation error is introduced in the shifting operation. The error in evaluating the new local expansion at an evaluation point at a distance ρ from the new center is bounded by $(\frac{\rho}{r'})^p$, if the original p–term local–field potential were a correct p–term expansion. However, since the original local expansion is not exact due to its derivation from a truncated far–field potential representation, the error bound is not guaranteed to be an upper bound.

Using the two types of computational elements and the three types of translations, the multipole–like hierarchical methods transform the interactions among particles that are well separated into interactions among particles and computational elements (in the BH method) [2] or among computational elements (in the multipole–like methods) [1, 7].

2.2. Parameters in computational structure. The second key idea in the hierarchical methods is hierarchical formation and evaluation of the computational elements.

Hierarchical methods divides the computational domain into a hierarchy of subdomains (meshes) (see Figure 2.2). Mesh level 0 represents the entire domain (*box*). Mesh level $l+1$ is obtained from level l by subdividing each subdomain at level l (*parent box*) into four (in two dimensions) or eight (in three dimensions) equally sized subdomains (*child boxes*). In an adaptive method, only subdomains with sufficiently many particles are further subdivided. Boxes that are not further subdivided are *leaves*. Hierarchical methods can be easily extended to rectangular domains in two dimensions and parallelepipedic domains in three dimensions [1].

The multipole–like $O(N)$ methods distinguish between three regions with respect to each subdomain (box) in the hierarchy. The definition of the three regions has a significant impact on the constant in the asymptotic arithmetic complexity. In the original formulation of multipole–based methods [8, 7], the *near–field* is defined as those subdomains that share a boundary point with the considered subdomain in two dimensions, and those subdomains which share a boundary point with the considered subdomain and *second nearest neighbor* subdomains which share a boundary point with the nearest neighbor subdomains in three dimensions. We denote these two kinds of near–fields as *one–separation* (1–Sep) and *two–separation* (2–Sep) neighborhoods, respectively. In general, the *d–separation* near–field in two and three dimensions contain $(2d+1)^2-1$ and $(2d+1)^3-1$ subdomains, respectively. The *far–field* of a subdomain is the entire domain excluding the subdomain and its near–field subdomains. The *interactive–field* of a subdomain at level l is the part of the far–field

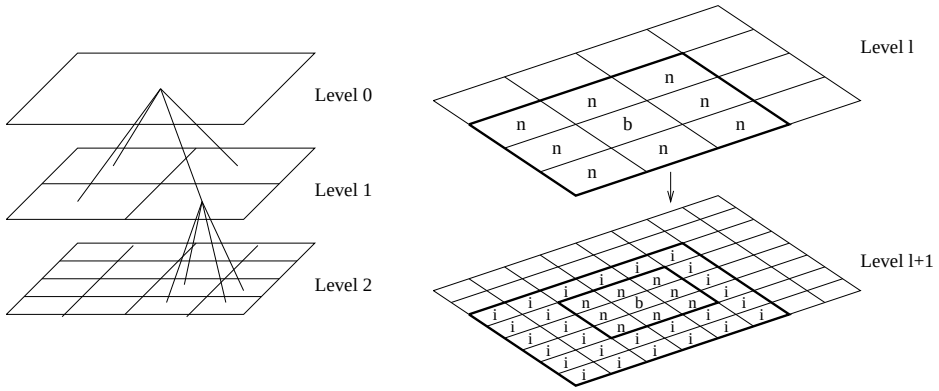


FIG. 2.2. *Recursive domain decompositions, the near–field, and the interactive–field in two dimensions.*

that is contained in its parent's near–field. In three dimensions, these definitions yield $7(2d+1)^3$ interactive–field subdomains (a $2(2d+1)\times 2(2d+1)\times 2(2d+1)$ subdomain excluding the near–field and the box itself).

In [21], Zhao defines the near–field according to the distance between individual boxes and the box under consideration. The interactive–field and far–field are defined accordingly. Under this definition, there are only 81 near–field and 567 interactive–field boxes in three dimensions.

Hierarchical methods can be abstracted in terms of three functions G, Φ, Ψ, the three aforementioned translation operators T_{F2F}, T_{F2L} and T_{L2L}, and a set of recursive equations. G is the potential function in an explicit Newtonian formulation, Φ_i^l is the contribution of subdomain i at level l to the potential field in domains in its far–field. Ψ_i^l represents the contribution to the potential field in subdomain i at level l due to particles in subdomain i's far–field region, i.e., the local–field potential in subdomain i at level l. We assume all Φ_i^l and Ψ_i^l are computed using computational elements of the same approximation order. The computational structure is described as follows by Katzenelson [12]:

Algorithm: (A generic hierarchical method)

1. Compute Φ_i^h for all boxes i at the leaf level h.
2. Upward pass: for $l = h-1, h-2, ..., 2$, compute
$$\Phi_n^l = \sum_{i \in \{children(n)\}} T_{F2F}(\Phi_i^{l+1}).$$
3. Downward pass: for $l = 2, 3, ..., h$, compute
$$\Psi_i^l = T_{L2L}(\Psi_{parent(i)}^{l-1}) + \sum_{j \in \{interactive-field(i)\}} T_{F2L}(\Phi_j^l).$$
4. Far–field: evaluate local–field potential at particle k inside every leaf–level subdomain
$$\phi_{k,\ far-field} = \Psi_{box(k)}^h(k).$$
5. Near–field: evaluate the potential field due to the particles in the near–field of leaf–level subdomains, using a direct evaluation of the Newtonian interactions

with nearby particles,

$$\phi_{k,\ near-field} = \sum_{j \in \{near-field(box(k))\}} G_j(k).$$

2.2.1. Errors. In multipole–based methods, Step 1 computes p–term expansions at the leaf–level thereby introducing a truncation error and is subject to round–off errors as well. Step 2 computes p–term expansions at all the nonleaf levels and is subject only to roundoff errors. Step 3 evaluates the local expansions at all the levels; the goal is to compute the local expansions at the leaf–level. The coefficients of the local expansions are subject to errors induced by the truncation of the far–field representation, which are accumulated during T_{L2L} translations. The local–field representation introduces new truncation errors by being themselves finite expansions, and are subject to round–off errors. Like T_{F2F}, T_{L2L} is only subject to round–off errors. If the p–term local expansions obtained from the far–field expansions were exact p–term expansions, then the relative error in the evaluation of local expansions at particles inside the leaf–level subdomains would be bounded by $((2d+2)/\sqrt{3}-1)^{-p}$ using d–separation near–fields. But, since the p–term expansions are not exact due to truncation errors affecting each coefficient and round–off errors, the upper bound for the error is at least $((2d+2)/\sqrt{3}-1)^{-p}$. Therefore, in the rest of the paper, we simply refer to this bound as the error estimate. Step 5 evaluates the potential due to near–field particles using the direct method, incurring only round–off errors.

Although varying the hierarchy depth does not change the separateness of interacting computational elements, it changes the portion of the particles in the near–field and far–field. Since the contribution from the particles in the two fields are computed differently, the overall error may differ with the hierarchy depth.

Note that in hierarchical methods, it may be beneficial to use higher order approximations for the far–field potentials than for the local–field potentials, since accuracy is lost in the far–field to local–field potential conversion.

2.2.2. Arithmetic operations. For N uniformly distributed particles and a hierarchy of depth h having 8^h leaf–level boxes, the numbers of operations required for the five stages of the above generic hierarchical method are listed in Table 2.3. It can be easily derived that the total number of arithmetic operations is minimized when

$$8^h \approx \sqrt{N_{near}/N_{int} f(p)} \cdot N, \tag{2.2}$$

i.e., number of particles in each leaf–level box is independent of N. We term the hierarchy depth that minimizes the total number of arithmetic operations the *optimal hierarchy depth*. Since the hierarchy depth can only be integer, the optimal hierarchy depth is either $\lfloor log_8 \sqrt{N_{near}/N_{int} f(p)} \cdot N \rfloor$ or $\lceil log_8 \sqrt{N_{near}/N_{int} f(p)} \cdot N \rceil$, whichever yields fewer total arithmetic operations. Clearly the optimal hierarchy depth balances the arithmetic operations at the two computation–dominant stages, namely, the downward pass and the direct evaluation in the near–field, since their arithmetic operation counts are proportional to 8^h and $N^2/8^h$, respectively. Ignoring the insignificant terms at the other three stages, the total arithmetic operation count at the optimal hierarchy depth is $O(\sqrt{N_{near} \cdot N_{int} \cdot f(p)} \cdot N)$, i.e., linear in the square root of translation costs.

Lowering the degree of separation of the near–field can result in significantly reduced arithmetic complexity. For example, the constant $\sqrt{N_{near} \cdot N_{int}}$ is reduced

TABLE 2.3

Arithmetic operations for the five stages of multipole–like hierarchical methods. $g(p)$ represent the computational cost as a function of the truncation order in establishing the computational elements. $f(p)$ represents the translation cost. N_{int} and N_{near} are the numbers of interactive–field boxes and near–field boxes for interior nodes, respectively. Without using supernodes, $N_{near} = (2d+1)^3$ and $N_{int} = 7(2d+1)^3$ for d–separation near–fields in three dimensions.

Stage	Arithmetic operations
Init–potential	$O(g(p) \cdot N)$
Upward pass	$O(f(p) \cdot 8^h)$
Downward pass	$O((N_{int} \cdot f(p) + f(p)) \cdot 8^h)$
Far–field	$O(g(p) \cdot N)$
Near–field	$O(N_{near} \cdot N^2/8^h)$
Minimized total	$O((g(p) + \sqrt{N_{near} \cdot N_{int} \cdot f(p)}) \cdot N)$

TABLE 2.4

The impact of using supernodes on arithmetic complexities and analytic error bounds of multipole–like $O(N)$ methods in three dimensions.

	N_{near}	N_{int}	$\sqrt{N_{near} \cdot N_{int}}$	Error bound
1–Sep	27	189	71	1.3^{-p}
1–Sep/supernode	27	56	39	1^{-p}
2–Sep	125	875	331	2.5^{-p}
2–Sep/supernode	125	189	154	2.1^{-p}
3–Sep	343	2401	907	3.6^{-p}
3–Sep/supernode	343	315	329	2.8^{-p}

by a factor of 4.6 by changing the separateness from two–separation to one–separation. The reduced separation will lower the base in the error estimate $((2d+2)/\sqrt{3}-1)^{-p}$ and therefore increase the error for a fixed p.

2.3. Supernodes. The use of supernodes [9, 21] reduces the effective value of N_{int} for a given separation, which brings about a dramatic improvement in the overall performance.

In [21], Zhao observes that for two–separations in three dimensions, there are many groups of eight sibling nodes of common parents in the the 567 interactive–field boxes of a destination box for the far–field to local–field conversion. Zhao then suggests converting the far–field of the parent node instead of the far–field of all eight sibling nodes, as shown in Figure 2.3. In this case, 488 parent nodes have all their children in the interactive–field, and using the concept of supernodes reduces the number of far–field to local–field conversions per destination node from 567 to 140. In GR's method in three dimensions, using supernodes reduces the number of interactive–field interactions per box from 875 to 189.

The idea of using supernodes can be generalized to any degree of separation. Table 2.4 lists the impact of using supernodes on arithmetic complexities and error bounds for one–, two–, and three–separations. If good error estimates were known, then for any given method that uses an optimal hierarchy depth and whose translation cost $f(p)$ is known, the tradeoff between arithmetic operations (cost) accuracy could easily be derived. Our simulations shows that for Anderson's method the error bounds are not useful as predictors of the actual error, neither with respect to the dependence on the truncation order nor with respect to the base constant. A discussion of our empirical models is given in Section 5.

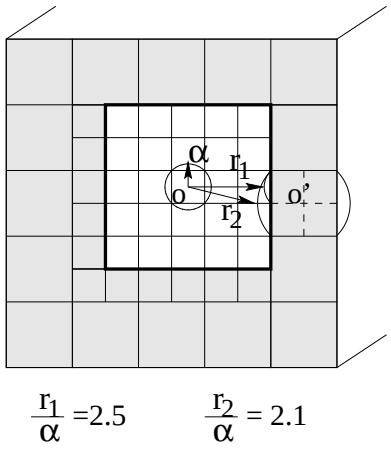


FIG. 2.3. *A supernode in the fast multipole method with two–separation.*

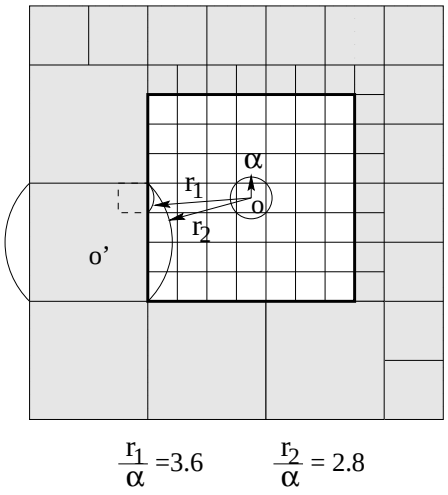


FIG. 2.4. *A supernode in the fast multipole method with three–separation.*

2.4. A comparison of multipole–like $O(N)$ hierarchical methods. Table 2.5 summarizes the arithmetic complexities of GR's and Anderson's methods in three dimensions using the optimal hierarchy depth for the uniform particle distribution. In GR's method, multiplications of translation coefficients which are independent of multipoles can be precomputed. Analogously, for Anderson's method, translation coefficients can be precomputed. Note that precomputation requires extra storage for precomputed data.

A direct comparison of the computational effort required by the different hierarchical methods for a given level of accuracy is clearly of interest. The lack of good error estimates for all considered methods unfortunately prevents such a comparison at this time. For uniform particle distributions, we develop in Section 5 empirical models for the errors in Anderson's method. Due to the unavailability of codes for GR's or Zhao's methods for 3–D problems we have not been able at this time to make a direct comparison of how the parameters of the different methods relate to each other. The few results we fond in the literature are given in Table 2.4. These results suggest that for a given desired error, GR's method requires a higher order approximation than Anderson's method, and therefore potentially more arithmetic operations than Anderson's method. Assuming the same base constant and $p = D/2 + 2$ as suggested by Anderson does not seem to be sensible.

3. Computational elements in Anderson's method. Anderson [1] uses Poisson's formula for representing solutions of Laplace equation, which results in computational elements that are simple to translate. Another advantage is that the computations in two and three dimensions are very similar. Therefore, a code for three dimensions is easily obtained from a code for two dimensions, or vice versa.

Let $g(x, y, z)$ denote potential values on a sphere of radius a, and Ψ denote the harmonic function external to the sphere with these boundary values. Given a sphere of radius a and a point $\vec{x}$ with spherical coordinates (r, θ, ϕ) outside the sphere, let $\vec{x_p} = (cos(\theta)sin(\phi), sin(\theta)sin(\phi), cos(\phi))$ be the point on the unit sphere along the vector from the origin to the point $\vec{x}$. The potential value at $\vec{x}$ is (equation (14) of

TABLE 2.5

The optimum number of arithmetic operations per particle for 3–D hierarchical N–body methods for uniform particle distributions. `SQRT` *and* `DIV` *are normalized as four floating–point operations [16]. m is the number of particles per leaf–level box that minimizes the total number of arithmetic operations. The optimal hierarchy depth h is either* $\lfloor log_8 \frac{N}{m} \rfloor$ *or* $\lceil log_8 \frac{N}{m} \rceil$*, whichever yields a smaller arithmetic operation count. Only high order terms are included. Thus, the formula are valid for large N, e.g.* $N > 100{,}000$*. Zhao's method requires much more arithmetic operations than GR's method and is not listed here. Note that for the same method, different separations yields different optimal number of particles per leaf–level box, and hence a different optimal number of floating–point operations per particle.*

FFT	Precomputation	optimal m	FLOPs/particle at $\frac{N}{8^d} = m$
GR : 2–Sep			
N	Y	$2.523(p+1)^2$	$6370(p+1)^2$
N	N	$2.958(p+1)^2$	$7463(p+1)^2$
Y	Y	$7.174p$	$18105p$
Y	N	$12.5p\sqrt{1+0.41\log_2 p}$	$31425p\sqrt{1+0.41\log_2 p}$
GR : 2–Sep/superd3			
N	Y	$1.172(p+1)^2$	$2973(p+1)^2$
N	N	$1.375(p+1)^2$	$3484(p+1)^2$
Y	Y	$3.398p$	$8613p$
Y	N	$5.84p\sqrt{1+0.41\log_2 p}$	$14682p\sqrt{1+0.41\log_2 p}$
Anderson : 2–Sep			
	Y	1.26K	$3280K$
	N	$0.893K\sqrt{6M+8}$	$2245K\sqrt{6M+8}$
Anderson : 2–Sep/superd3			
	Y	0.589K	$1590K$
	N	$0.416K\sqrt{6M+8}$	$1046K\sqrt{6M+8}$
Anderson : 1–Sep			
	Y	1.26K	$810K$
	N	$0.887K\sqrt{6M+8}$	$491K\sqrt{6M+8}$

TABLE 2.6

Comparison of published accuracy results. For $p = D/2 + 2$ *as suggested by Anderson, his method yields a more accurate result than the multipole–based methods. 2–Sep/superdi denotes that supernodes are being invoked at level i and successive levels towards the leaves.*

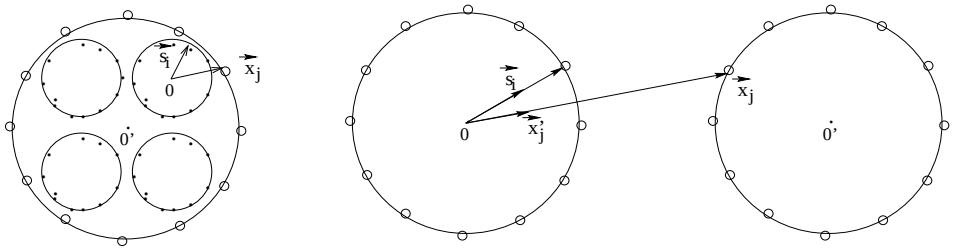
Distribution	Anderson, $D = 14$		GR, $p = 8$	Zhao, $p = 9$
	1-Sep	2-Sep/superd4	1-Sep	2-Sep/superd3
N=1000, uniformly dist. & charged, RMS force [21]	1.2e-05	6.1e-06		9.7e-05
N=40000, uniformly dist., random charges in [-1,1], RMS potential [16]	5.4e-5	1.6e-05	1.2e-04	

[1])

$$\Psi(\vec{x}) = \frac{1}{4\pi} \int_{S^2} \left[\sum_{n=0}^{\infty} (2n+1)(\frac{a}{r})^{n+1} P_n(\vec{s} \cdot \vec{x_p}) \right] g(a\vec{s}) ds, \tag{3.1}$$

where the integration is carried out over S^2, the surface of the unit sphere, and P_n is the nth Legendre function.

Given a numerical formula for integrating functions on the surface of the sphere with K integration points $\vec{s_i}$ and weights w_i, the following formula (equation (15) of



(a) Translations T_{F2F} and T_{L2L} (b) Translation T_{F2L}

FIG. 3.1. *Translations as evaluations of the approximations*

[1]) is used to approximate the potential at $\vec{x}$:

$$\Psi(\vec{x}) \approx \sum_{i=1}^{K} \left[\sum_{n=0}^{M} (2n+1)(\frac{a}{r})^{n+1} P_n(\vec{s_i} \cdot \vec{x_p}) \right] g(a\vec{s_i}) w_i \tag{3.2}$$

This approximation is called an *outer–sphere approximation*. Note that in this approximation, two approximations are made compared to Equation (3.1): the series is truncated, and the integral is evaluated with a finite number of terms.

The approximation used to represent potentials inside a given region is (equation (16) of [1])

$$\Psi(\vec{x}) \approx \sum_{i=1}^{K} \left[\sum_{n=0}^{M} (2n+1)(\frac{r}{a})^{n+1} P_n(\vec{s_i} \cdot \vec{x_p}) \right] g(a\vec{s_i}) w_i \tag{3.3}$$

and is called an *inner–sphere approximation*.

The outer–sphere and the inner–sphere approximations define the computational elements in Anderson's hierarchical method. Clustering is first done at the leaf–level, as in the generic method; outer–sphere approximations are constructed for clusters of particles in the leaf–level boxes. During the upward pass, outer–sphere approximations of child boxes are combined into a single outer–sphere approximation of their parent box (T_{F2F}). This combining operation is particularly simple to implement; one simply evaluates the potential induced by the component outer–sphere approximations at the integration points of the parent outer–sphere approximation, as shown in Figure 3.1. The situation is similar for the other two translations used in the method, which are shifting a parent box's inner–sphere approximation to add to its children's inner–sphere approximations (T_{L2L}) and converting the outer–sphere approximations of a box's interactive–field boxes to add to the box's inner–sphere approximation (T_{F2L}).

3.1. Parameters in sphere approximations. The integration order for the sphere approximations determines the choices of K, M, and a, i.e., the integration order affects the truncation of the series expansion. Table 3.1 lists the relationships among the parameters and the expected error decay rate in the approximation of a single element in Anderson's method, as given in [1].

A suitable value for K depends upon the choice of integration method, and required accuracy. The traditional integration methods based on a product grid, which typically use trapezoidal integration in the θ direction and Gaussian quadrature in the ϕ direction, are inefficient since the integration points are crowded near the poles.

TABLE 3.1
Parameter selections and expected and verified errors of outer/inner sphere approximations in Anderson's method. α is the side length of a box.

Order of integration D	K	M ($\leq$ D/2)	$\beta_0 = \frac{a_{outer}}{\alpha}$	$\gamma_0 = \frac{\alpha}{a_{inner}}$	Expected error decay rate (M+2)	Avg. error decay rate
5	12	2	10	2	4	3.95
7	24	3	7	2	5	5.68
9	32	4	5	2	6	8.85
11	50	5	2.5	2	7	7.32
14	72	7	2	2	9	8.90

Anderson chose to use integration formulas from a non–product family, as described in [15]. The 5th, 7th, 9th, 11th, and 14th order integration formulas require 12, 24, 32, 50, 72 integration points, respectively.

A suitable choice of M is related to the choice of the order of the integration method. In fact, aliasing occurs if there are more terms in the series expansion than can be resolved by the sampling points used in the integration. Anderson showed that if an integration formula for the sphere of degree D is used, then an appropriate choice for M is $M \leq \frac{D}{2}$. The reason is that for a function on the sphere comprised of spherical harmonics of degree m, the spherical expansion coefficients of this function can be determined exactly by an integration formula with accuracy of degree $2m$.

Finally, a, the radius of the sphere of integration for each computational element must be determined. If the particles are centered within a sphere of radius α, we can choose any radius $a \geq \alpha$ for the sphere of integration for computational elements representing outer–sphere approximations. For the outer–sphere potential approximation of the fields of a collection of particles inside a sphere of radius α, the error at distance $r > \alpha$ from the center of the sphere is $O(\frac{\alpha}{r})^{M+2}$. The expected error decay rate $M + 2$ and the actual error decay rate of single sphere approximation observed by Anderson in his sequential code are listed in Table 3.1. The larger the value of a, the smoother the potential values at the integration points, and the smaller the error in the evaluation of outer–sphere approximations. On the other hand, for very large values of a the fact that floating–point accuracy is of limited precision could result in loss of numerical accuracy in accumulating terms. Anderson suggested that the proper choice of a for the outer–sphere approximations is $a = 2\alpha$, and that for the inner–sphere approximations $a = \alpha/2$ shall be used. In our simulations, unless stated otherwise, we use the outer sphere radii listed in Table 3.1.

3.2. Translations as matrix–vector multiplications. The translations in Anderson's method can be performed as matrix–vector multiplications. A field approximation (equation (3.2) or (3.3)) can be rewritten as

$$\Phi(\vec{x_j}) \approx \sum_{i=1}^{K} f(\vec{s_i}, \vec{x_j}) \cdot g(a\vec{s_i}), \quad j = 1, K, \tag{3.4}$$

where $f(\vec{s_i}, \vec{x_j})$ represents the inner summation in the original approximation. $f(\vec{s_i}, \vec{x_j})$ is a function of the vector $\vec{s_i}$ from the origin of the source sphere to its ith integration point and the vector $\vec{x_j}$ from the origin of the source sphere to the jth integration point on the destination sphere. The evaluation of the field at an integration point on the destination sphere due to the field values at all integration points on the source spheres is an inner–product computation. Hence, the evaluation of the field at all

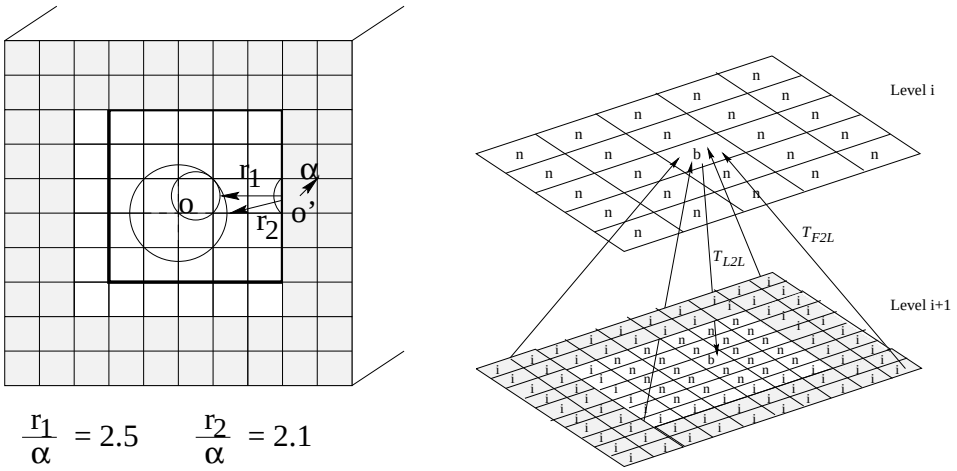


FIG. 3.2. *Supernodes in Anderson's method.*

integration points on the destination sphere due to the field values at all the integration points of the source sphere constitutes a matrix–vector multiplication, where the matrix is of shape $K \times K$. We refer to this matrix as a *translation* matrix. The entries of the translation matrix only depends on the relative locations of the source and destination spheres. Thus, for each translation operator, the set of translation matrices at different levels of the hierarchy are all the same. Many translation matrices within a given level are also identical. Hence, they can be precomputed. Each translation can be performed as a matrix–vector multiplication requiring $2K^2$ operations. Furthermore, translations for boxes at the same level can be aggregated and performed as collections of matrix–matrix multiplications (multiple–instance matrix–matrix multiplication) for higher efficiency (see [10] for details.)

3.3. Supernodes in Anderson's method. The fact that translations in Anderson's method, e.g., T_{F2L}, are evaluations of the sphere approximations leads to a different way of using supernodes. The error in evaluating an outer–sphere approximation (equation (3.2)) with the series expansion truncated after M terms is $O(\frac{\alpha}{r})^{M+2}$, where α is the radius of the sphere containing the particles and r is the distance of the evaluation point from the center of the sphere [1]. Therefore, it is desirable to perform the conversion from a small sphere to a large sphere. Specifically, if a box in the interactive–field is common to all sibling nodes of a parent, then the conversion is made with the parent box as a target instead of the sibling boxes, as shown in Figure 3.2. The converted local–field is added to the local–field of the parent. When all such conversions have been made, the local–field is passed to each of the children through the eight translation operations T_{L2L}, and the local–field conversions from the remaining interactive–field boxes are added. This way of using supernodes is dual to the supernodes in multipole–based methods, and the reduction in N_{int} remains the same.

4. Simulation environment and error definitions. The empirical evaluation in this paper were performed on the Connection Machine system CM–5E using our data–parallel implementation of a nonadaptive version of Anderson's method [10]. The code is written in Connection Machine Fortran (CMF) [18] with subroutine calls to the Connection Machine Scientific Software Library (CMSSL) [19] for nodal BLAS

operations. Detailed descriptions of the performance of the implementation can be found in [10]. We measure the error of Anderson's method relative to a parallel implementation of the direct method. This code is also written in CMF and uses cyclic shift operations to perform all–to–all communication. It exploits symmetry in the potential/force evaluation from Newton's third law. Both codes perform floating–point operations in 64–bit precision. All the simulation results presented are collected on a 32 node CM–5E configuration unless otherwise stated.

Characteristics common to many simulations were as follows:

- Uniform particle distribution: there are an equal number of particles uniformly and randomly distributed in each leaf–level box, unless otherwise specified. The particle coordinates are $\mathbf{r}_i = (x_i, y_i, z_i), 1 \leq i \leq N$.
- Uniform charges: the particles are assigned equal charges of the same sign, and the total charge is made 1, i.e. $q_i = 1.0/N, 1 \leq i \leq N$.

The code computes the Coulombic potential and the force field at each particle location.

$$\phi_i = \sum_{j=1,N}^{j \neq i} \frac{q_j}{\|\mathbf{r}_i - \mathbf{r}_j\|}$$

$$\mathbf{f}_i = \sum_{j=1,N}^{j \neq i} \frac{q_j(\mathbf{r}_i - \mathbf{r}_j)}{\|\mathbf{r}_i - \mathbf{r}_j\|^3}$$

The actual *potential energy* and *force* for particle i are $\phi_i \cdot q_i$ and $\mathbf{f}_i \cdot q_i$, respectively. The relative errors in computing the potential and the force field are the same as those in computing the potential energy and the force.

Let $\phi_i^c, \mathbf{f}_i^c$ denote the computed values using Anderson's method, and $\phi_i^r, \mathbf{f}_i^r$ denote the computed values using the direct method.

We use Newton's third law to assess the error in the force calculation using the direct method. This error gives a rough measure of the round–off errors. For the entire range of number of particles simulated, the sum of all the force fields is smaller than 5×10^{-14}.

The error in the potential and force fields in the simulations are determined using the following quantities, where $||\ ||$ denotes the l_2–norm,

$$\beta_i = |\frac{\phi_i^r - \phi_i^c}{\phi_i^r}|$$

$$\gamma_i = \frac{\|\mathbf{f}_i^r - \mathbf{f}_i^c\|}{\|\mathbf{f}_i^r\|}$$

$$\epsilon_{rms\_rel\_pot} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \beta_i^2}$$

$$\epsilon_{rms\_rel\_for} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \gamma_i^2}$$

$\epsilon_{rms\_rel\_pot}$ and $\epsilon_{rms\_rel\_for}$ are the root–mean–square relative errors of potential and force fields, respectively.

5. Accuracy–cost tradeoffs. As discussed in Section 2, in addition to the parameters of a given particle system (the number of particles, the particle coordinates

and charge distributions), the parameters of a multipole–like hierarchical $O(N)$ N–body method (the approximation order, the degree of separation in the near–field, the use of supernodes, and the hierarchy depth) also affect the accuracy (ϵ) and the execution time per particle (T/N), i.e.,

$$\begin{aligned}\epsilon = f(&N, \text{charge}, \text{distribution}, \text{approximation order},\\ &\text{hierarchy depth } h, \text{separation}, \text{supernode})\\ \text{T}/N = g(&N, \text{charge}, \text{distribution}, \text{approximation order},\\ &\text{hierarchy depth } h, \text{separation}, \text{supernode})\end{aligned}$$

An important issue in using these approximate N–body methods is how to choose the parameters of the methods so that the code performs the potential or force evaluation in the shortest time while satisfying a prescribed accuracy requirement.

We first consider uniform distributions for which the accuracy and execution time per particle can be expressed as

$$\begin{aligned}\epsilon &= f_1(N, \text{approximation order}, \text{hierarchy depth } h, \text{separation}, \text{supernode})\\ \text{T}/N &= g_1(N, \text{approximation order}, \text{hierarchy depth } h, \text{separation}, \text{supernode})\end{aligned}$$

The numerical simulations discussed in Section 5.2 show that for systems of up to one million uniformly distributed particles, the accuracy is independent of the number of particles and the hierarchy depth, i.e.,

$$\epsilon = f_2(\text{approximation order}, \text{separation}, \text{supernode})$$

Since the error is independent of the hierarchy depth, we choose the hierarchy depth such that it minimizes the total number of arithmetic operations, i.e., the FLOP count. Using an optimal hierarchy depth, the FLOP count per particle is independent of the number of particles (see Table 2.3), i.e.,

$$T/N = g_2(\text{approximation order}, \text{separation}, \text{supernode})$$

Since the optimal hierarchy depth balances the two computation–dominant stages – the hierarchy traversal and the direct evaluation in the near–field, and the efficiency of the two stages are similar in our implementation of Anderson's method, the FLOP count per particle is an accurate estimate of the execution time. Model equations for the prediction of the execution time is discussed in Section 5.3.

Based on our error and execution time models, we present in Section 5.4 a procedure for selecting parameters that minimizes the execution time for a given accuracy for uniform particle distributions.

5.1. Controlled use of supernodes. Our simulations show that during the downward traversal of the hierarchy, delaying the use of supernodes until further away from the root improves the accuracy of the code significantly, compared to using supernodes at all possible levels. Figure 5.1 shows that using supernodes starting at level four (with root being level zero) improves the accuracy by up to a factor of five compare to starting at level three for a seven–level hierarchy, and another factor of five is observed if the use of supernodes is delayed until level five.

To understand why delaying the use of supernodes to levels four or five improves the accuracy of the method dramatically, we consider the relative contributions and accuracies of the potentials from the interactive–fields at different levels. For any

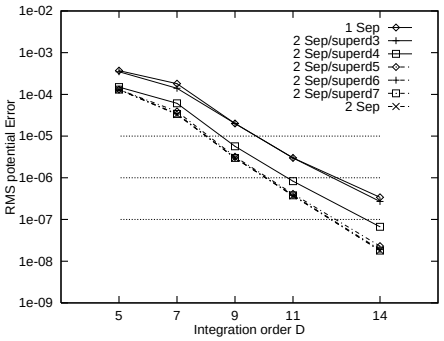


FIG. 5.1. *Delaying the use of supernodes improves the accuracy for 1M particles with depth seven hierarchy.*

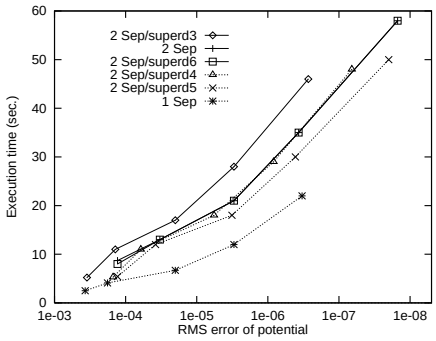


FIG. 5.2. *Execution time vs. accuracy on a 256 node CM–5E for 1M particles with optimal hierarchy depths.*

multipole–like hierarchical $O(N)$ method, the potential evaluation at each particle using a hierarchy of depth h can be decomposed as

$$\phi = \phi_I^{d=2} + \phi_I^{d=3} + ... + \phi_I^{d=h} + \phi_N^{d=h},$$

where $\phi_N^{d=i}$, $\phi_I^{d=i}$ denote the potential due to particles in the near–field and interactive–field at level i, respectively. The error from every $\phi_I^{d=2}$ contributes to the total error. $\phi_N^{d=h}$ is computed using the direct method. Table 5.1 lists the total potential and the different contributions for 2–Sep, 2–Sep/superd3 and 1–Sep. The weights are defined as the ratio obtained by dividing the potential (computed by the direct method) contributed from a subregion by the total potential, averaged over all particles. For a separateness of two, the potential contributions from the interactive–fields at level two, three, and four, i.e. $\phi_I^{d=2}, \phi_I^{d=3}, \phi_I^{d=4}$, carry large weights. Therefore, their accuracies dominate the accuracy of the total potential. This explains the improvement of the error offered by delaying the use of supernodes till level four and five, as seen in Figure 5.1.

Delaying the use of supernodes until after a few levels close to the root adds little to the total execution time, compared to the use of supernodes at all levels. For simulations with a large number of particles, the optimal hierarchy depth is deep. For such simulations, the amount of computation close to the root is insignificant compared to the computations close to the leaf–level. Second, the use of supernodes in parallel implementations of Anderson's method offers little gain in execution time close–to–root. This fact can be understood through the following consideration. In Anderson's method, for a separateness of two, using supernodes at level i turns 875 interactive–field neighbor interactions per box at level i into 91 neighbor interactions per box at level i and 784 neighbor interactions per box at level $i-1$. Since our code only exploits parallelism among same–pair boxes at the same level, and sequentialize the interactive–field neighbor interactions [11], each neighbor interaction at levels close to the root where there are fewer boxes than processing nodes take about the same time. Thus, e.g., 875 neighbor interactions at level i takes about the same time as 91 neighbor interactions at level i and 784 interactions at level $i-1$, if levels $i-1$ and i are sufficiently close to the root that the computations are fully parallel. Therefore, neighbor interactions with or without supernodes at these levels take about the same time. For example, on a 32 node CM–5, at level three, the neighbor interactions using supernodes is only 24% faster than without supernodes. In addition, the cost

TABLE 5.1
Potential contributions from the interactive–fields at different levels for 1M particles and a depth seven hierarchy.

Potential	Weight	RMS potential error				
		$D = 5$	$D = 7$	$D = 9$	$D = 11$	$D = 14$
2–Sep, ϕ	1.00e+00	1.3e-04	3.3e-05	3.0e-06	3.7e-07	1.8e-08
$\phi_{int}^{d=2}$	2.11e-01	5.7e-04	9.3e-05	8.7e-06	1.1e-06	3.5e-08
$\phi_{int}^{d=3}$	4.74e-01	1.0e-04	3.0e-05	2.2e-06	2.3e-07	1.7e-08
$\phi_{int}^{d=4}$	2.16e-01	5.5e-05	2.1e-05	1.2e-06	1.2e-07	1.5e-08
$\phi_{int}^{d=5}$	7.12e-02	3.4e-05	1.6e-05	6.7e-07	9.0e-08	1.3e-08
$\phi_{int}^{d=6}$	2.04e-02	4.5e-05	1.5e-05	8.4e-07	1.5e-07	1.4e-08
$\phi_{int}^{d=7}$	4.68e-03	3.9e-05	1.1e-05	7.5e-07	1.2e-07	8.8e-09
2–Sep/superd3, ϕ	1.00e+00	3.5e-04	1.4e-04	2.0e-05	3.0e-06	2.7e-07
$\phi_{int}^{d=2}$	2.11e-01	5.7e-04	9.3e-05	8.7e-06	1.1e-06	3.5e-08
$\phi_{int}^{d=3}$	4.74e-01	6.2e-04	2.3e-04	3.8e-05	6.3e-06	5.9e-07
$\phi_{int}^{d=4}$	2.16e-01	3.5e-04	1.7e-04	2.1e-05	3.7e-06	3.5e-07
$\phi_{int}^{d=5}$	7.12e-02	2.2e-04	1.3e-04	1.2e-05	2.4e-06	2.1e-07
$\phi_{int}^{d=6}$	2.04e-02	1.5e-04	1.1e-04	8.3e-06	1.9e-06	2.2e-07
$\phi_{int}^{d=7}$	4.68e-03	2.6e-04	1.3e-04	1.7e-05	6.8e-06	1.0e-06
1–Sep, ϕ	1.00e+00	3.7e-04	1.8e-04	2.0e-05	3.0e-06	3.4e-07
$\phi_{int}^{d=2}$	5.76e-01	6.1e-04	2.5e-04	2.8e-05	4.3e-06	5.2e-07
$\phi_{int}^{d=3}$	2.87e-01	3.4e-04	1.8e-04	1.6e-05	2.5e-06	3.8e-07
$\phi_{int}^{d=4}$	9.77e-02	2.2e-04	1.4e-04	9.3e-06	1.8e-06	2.9e-07
$\phi_{int}^{d=5}$	2.84e-02	1.6e-04	1.1e-04	8.0e-06	2.9e-06	3.8e-07
$\phi_{int}^{d=6}$	7.67e-03	3.4e-04	1.2e-04	2.0e-05	7.3e-06	1.0e-06
$\phi_{int}^{d=7}$	2.43e-03	1.1e-03	7.0e-04	1.2e-04	5.0e-05	8.6e-06

for neighbor interactions at level three for 2–Sep/superd3 is less than 5% of the total execution time for the hierarchy traversal for a depth six hierarchy. Since the 2–Sep/superd4 improves the accuracy by a factor of five compared to 2–Sep/superd3, it is more accuracy–cost efficient than the later one.

Since 2–Sep/superd3 yields errors similar to 1–Sep, but requires more arithmetic operations, and similarly, 2–Sep/superd5 yields errors similar to 2–Sep and 2–Sep/superd$i$ for $i \geq 5$, but requires fewer arithmetic operations, we only consider 1–Sep, 2–Sep/superd4, and 2–Sep/superd5 in this Section. Figure 5.2 plots the execution time on a 256 node CM–5E as a function of accuracy for the six methods using optimal hierarchy depths. 1–Sep and 2–Sep/superd5 yields the lowest execution time for a given accuracy, while 2–Sep/superd4 is more cost efficient than 2–Sep and 2–Sep/superd6 for low accuracy and about the same as the latter two for high accuracy.

5.2. Error models. Our simulations show that the accuracy is independent of the number of particles and the hierarchy depth. Figures 5.3– 5.6 plot the RMS error in the potential as a function of the number of particles for different hierarchy depths, integration order and different separateness and use of supernodes. Note that for a small number of particles, the accuracy in all cases become worse as the hierarchy depth increases. The reason is that as the hierarchy depth increases, for a relatively small number of particles, there is an increased number of empty leaf–level boxes. Thus, the particle distribution relative to the nonleaf boxes is less balanced, which results in larger integration errors in traversing the hierarchy.

Similarly, the RMS error in the force fields is also independent of the number of particles and the hierarchy depth.

Figures 5.7–5.8 plot the the RMS error of potential and force field as a function

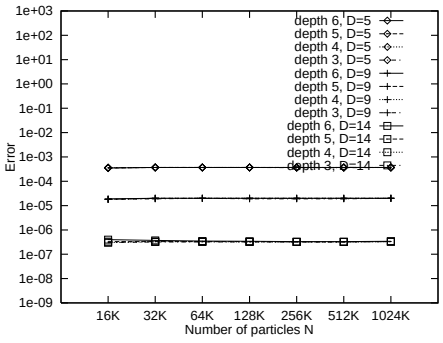


FIG. 5.3. *The RMS error of the potential for 1–Sep at depth 3,4,5,6*

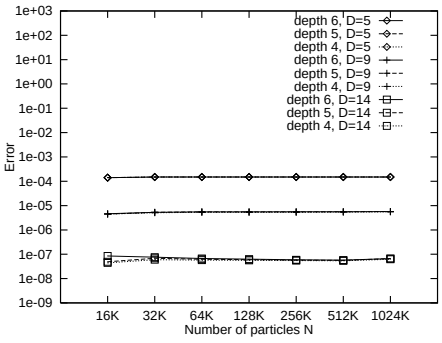


FIG. 5.4. *The RMS error of the potential for 2–Sep/superd4 at depth 4,5,6*

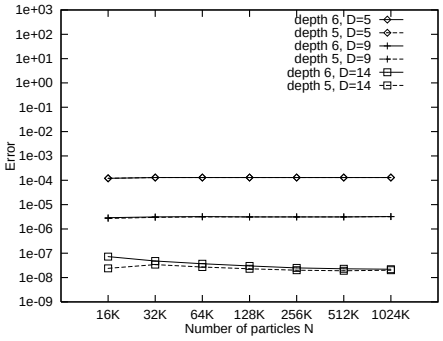


FIG. 5.5. *The RMS error of the potential for 2–Sep/superd5 at depth 5,6*

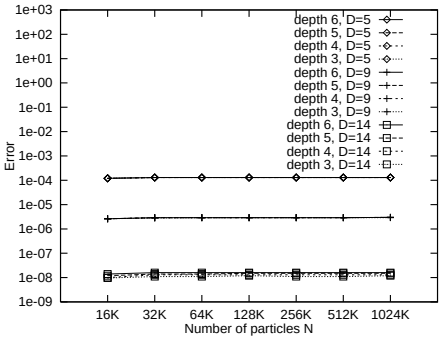


FIG. 5.6. *The RMS error of the potential for 2–Sep at depth 3,4,5,6*

of the integration order D. The RMS error of the force field is about two orders of magnitude larger than that of the potential field. We focus on the RMS potential error in the rest of the paper since the code always computes the potential (the sphere approximations evaluate the potentials), while the forces are computed either via numerical differentiation based on the potential or by calling additional subroutines for analytical differentiation. The error in the force field is very sensitive to the normalization used. We use a local normalization whereas several other studies reporting smaller force field errors use a global normalization (for example, see [14]).

Assuming the error in the potential can be modeled by $c_1 \cdot c^{-p}$, we can determine either the decay rate or the base constant from our simulations assuming the other is known. If we assume the error decay $p = D/2 + 2$, then the constants c and c_1 for 1–Sep, 2–Sep/superd4, 2–Sep/superd5 can be computed using a first–order least–squares–fit (LSF) as in Table 5.2. If we instead assume the base constant c is known and has the value given in Table 2.4, then the decay rate p and the constant c_1 for the five integration orders can be computed using a first–order LSF as in Table 5.3. Both the decay rate and the base constant thus determined differ significantly from the values from single outer–sphere approximations verified by Anderson [1]. The rather large base constants in Table 5.2 suggests that for computational elements of similar decay rates, Anderson's method is more accurate than GR's or Zhao's methods, for which the base constants are about two [21].

Attempting to verify the base constant and error decay rate from our simulations

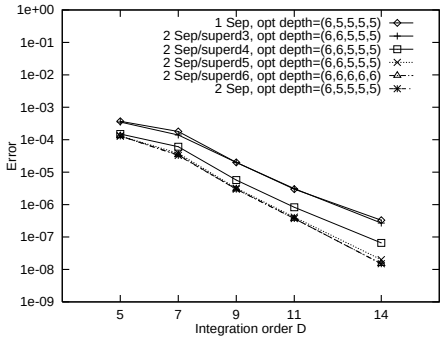


FIG. 5.7. *The RMS error of the potential for 1M particles and optimal hierarchy depth for each of the five integration orders.*

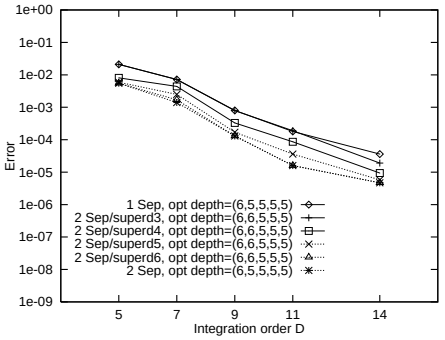


FIG. 5.8. *The RMS error for forces for 1M particles and optimal hierarchy depth.*

TABLE 5.2
Base constant and proportionality factor determined from simulations assuming an error decay rate $p = D/2 + 2$.

Separation	c	c_1	LSE $= \sum_i \epsilon_i^2$
1–Sep	4.44	0.173	1.4e-1
2–Sep/superd4	5.08	0.117	1.3e-1
2–Sep/superd5	6.19	0.214	9.1e-2

TABLE 5.3
Error decay rate and proportionality factor determined from simulations assuming a base constant $c = ((2d+2)/3^{\frac{1}{2}} - 1)^{-p}$.

Integration order D	p	c_1	LSE $= \sum_i \epsilon_i^2$
5	1.66	5.59e-04	2.2e-3
7	2.52	3.58e-04	2.5e-3
9	2.84	4.31e-05	2.2e-3
11	3.09	7.03e-06	7.3e-3
14	4.44	1.18e-06	4.8e-2

as above gives rise to some concern over the accuracy of the analytic models. Since the accuracy is independent of the number of particles and the hierarchy depth, we only need to model the error as a function of integration order for each combination of separation and use of supernodes. Using second–order LSF, let

$$log_{10}(\epsilon) = C_1 \cdot D^2 + C_2 \cdot D + C_3 \tag{5.1}$$

The fitted constants are given in Table 5.4.

Given an error ϵ, we can compute the required integration order D using equation (5.1). Since the integration orders are discrete and take values in {5,7,9,11,14}, the computed D is rounded up to the nearest available integration order. As verified through simulations shown in Table 5.5, predicting the integration order for a given accuracy using equation (5.1) is quite accurate.

5.3. Timing models. In the early papers on the fast multipole method, the hierarchy depth is chosen such that there is roughly one particle in each leaf–level box (for example, see [6, 7, 9, 16, 21]), or there are a predetermined number of particles per leaf–level box [4]. Later, the issue of using an optimal hierarchy depth to minimize the total number of arithmetic operations and hopefully also the execution time of the code was raised [22]. In [1], Anderson proposed a performance model based on the FLOP count for predicting the absolute execution time of different stages of the method on uniprocessors for any given particle distribution and hierarchy depth. Based on the performance model, he proposed a parameter selection procedure that selects the optimal hierarchy depth through a "dry run" of the method with variable hierarchy depth.

TABLE 5.4
Second–order LSF of the RMS potential error as a function of the integration order (see equation (13).)

Method	C_1	C_2	C_3	LSE = $\sum_i \epsilon_i^2$
1–Sep	-4.42e-03	-.274	-1.84	8.8e-02
2–Sep/superd4	-4.69e-03	-.301	-2.10	7.6e-02
2–Sep/superd5	-5.88e-03	-.325	-2.04	4.7e-02

TABLE 5.5
Predicted and measured integration orders required for a given RMS error of the potential. For the mispredicted case $^{\uparrow}$*, using the mispredicted integration order would give an RMS error of 1.8e-04.*

RMS	Predicted D			Measured D		
error	1–Sep	2–Sep/superd4	2–Sep/superd5	1–Sep	2–Sep/superd4	2–Sep/superd5
1.0e-02	5	5	5	5	5	5
5.0e-03	5	5	5	5	5	5
1.0e-03	5	5	5	5	5	5
5.0e-04	5	5	5	5	5	5
1.0e-04	$7^{\uparrow}$	7	5	9	7	5
5.0e-05	9	7	7	9	7	7
1.0e-05	11	9	9	11	9	9
5.0e-06	11	9	9	11	9	9
1.0e-06	14	11	11	14	11	11
5.0e-07	14	14	11	14	14	11

For a parallel implementation of a hierarchical method, the optimal hierarchy depth has to take into account the communication needs of the implementation and the communication efficiency of the underlying parallel machine. For our implementation of Anderson's method on the CM–5/5E, the FLOP count gives rather accurate prediction on the relative execution time. First, the optimal hierarchy depth balances the two computation–dominant stages – the downward pass and the direct evaluation in the near–field (see Table 2.3.) As reported in [10], for problems with reasonably large sizes relative to the machine size, the two time–dominant stages achieve similar efficiencies, and therefore the FLOP count is a good predictor of the optimal hierarchy depth with respect to execution time for both methods. Second, the efficiencies for different separations and use of supernodes, e.g. 1–Sep, 2–Sep/superd4 and 2–Sep/superd5, are almost the same for the same hierarchy depth. Thus, it is expected that the FLOP count for Anderson's method predicts the parallel execution time well for all combinations of separations, use of supernodes, and hierarchy depths.

Since all simulations were performed on the CM–5/5E, all FLOP counts used in this section uses CM–5 normalized FLOP counts, i.e., `DIV` and `SQRT` are counted as five and eight FLOPs, respectively. This count accurately reflects the actual number of (floating–point) cycles they take on the Vector Units of CM–5/5E.

Table 5.6 lists the predicted and measured optimal hierarchy depths for the three methods. The prediction is based solely on the FLOP counts. For the miss–predicted cases, the running time using the miss–predicted depth and the measured optimal depth differ by at most 20%.

Figures 5.9–5.12 plot the FLOP ratio as a function of the number of particles for different degrees of separations and different use of supernodes. In general, since the hierarchy depth is always an integer, the FLOP counts and consequently the ratios between the different computational approximations is expected to oscillate

TABLE 5.6
Predicted and measured optimal hierarchy depth.

N	1–Sep					2–Sep/superd4					2–Sep/superd5				
D	5	7	9	11	14	5	7	9	11	14	5	7	9	11	14
measured															
16k	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
32k	4	3	3	3	3	4	4	3	3	3	4	3	3	3	3
64k	4	4	4	3	3	4	4	4	4	3	4	4	4	3	3
128k	4	4	4	4	4	5	5	4	4	4	5	5	4	4	4
256k	5	4	4	4	4	5	5	5	5	4	5	5	5	5	4
512k	5	5	5	4	4	5	5	5	5	5	5	5	5	5	5
1024k	6	5	5	5	5	6	6	5	5	5	6	6	5	5	5
predicted - measured															
16k	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
32k	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
64k	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
128k	0	0	0	0	0	0	-1	0	0	0	0	-1	0	0	0
256k	0	0	0	0	0	0	0	0	-1	0	0	0	0	-1	0
512k	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1024k	-1	0	0	0	0	0	-1	0	0	0	0	-1	0	0	0

since the optimal depth is not necessarily the same for the different approximations. For each approximation, the FLOP count per particle exhibits a period for each factor of eight increase in the number of particles. The use of supernodes, and delayed use of such nodes, alters this simple explanation. For instance, for 2–Sep/superd4 and 2–Sep/superd5 the FLOP counts per particle differ before and after starting the use of supernodes. In Figure 5.9, the optimal depth is less than four for a small number of particles, and supernodes are not used for 2–Sep/superd4. The ratio therefore is about four (see Table 2.4.) For large number of particles, the optimal hierarchy depth is beyond four, and supernodes are used. The ratio then decreases to around two. Similar trends are seen in Figure 5.10, except that the decrease in the ratio occurs for a larger number of particles since the level at which supernodes are used is delayed to five. For the ratio $FC_{2-Sep/superd5}/FC_{2-Sep/superd4}$ in Figure 5.11, the ratios are around one for small and large numbers of particles, since both approximations plotted use or do not use supernodes at levels close to the leaf–level, where the arithmetic operations dominate the hierarchy traversal. For a number of particles for which the optimal hierarchy depth is four, the ratio is around two as 2–Sep/superd4 uses supernodes at level four and 2–Sep/superd5 does not. Figure 5.12 plots the FLOP count ratios as a function of the integration order D.

To verify the accuracy of using the FLOP count to predict the relative execution times of the different approximation methods, we plot the ratios of the measured execution times of the three methods in Figures 5.13–5.16. The plots differ somewhat from the FC ratio plots in Figures 5.9–5.12. In general, the execution time ratios are lower than the FLOP count ratios, since for the same integration order, 2–Sep/superd4 and 2–Sep/superd5 often has an optimal hierarchy depth that is one larger than that of 1–Sep, and therefore achieve a better efficiency. Also note that for a small number of particles, there is little computation and the relatively high overhead (e.g. precomputation of translation matrices) hides the FLOP count difference among the approximation methods.

5.4. Accuracy–cost tradeoffs for Anderson's method. Based on the error models and timing models developed above, the selection of the degree of separation for the near–field, the use of supernodes, and the hierarchy depth that minimizes

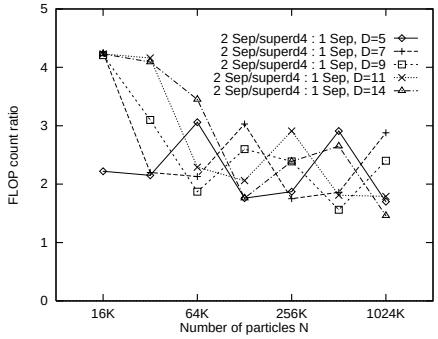


FIG. 5.9. *The FLOP count ratio* $FC_{2-Sep/superd4}/FC_{1-Sep}$ *at optimal depth.*

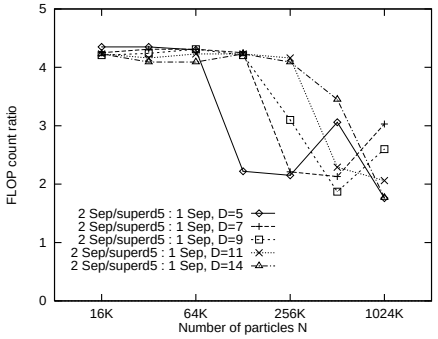


FIG. 5.10. *The FLOP count ratio* $FC_{2-Sep/superd5}/FC_{1-Sep}$ *at optimal depth.*

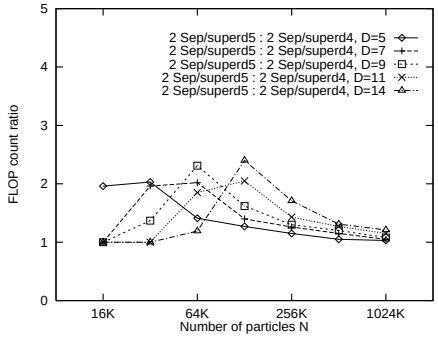


FIG. 5.11. *The FLOP count ratio* $FC_{2-Sep/superd5}/FC_{2-Sep/superd4}$ *at optimal depth.*

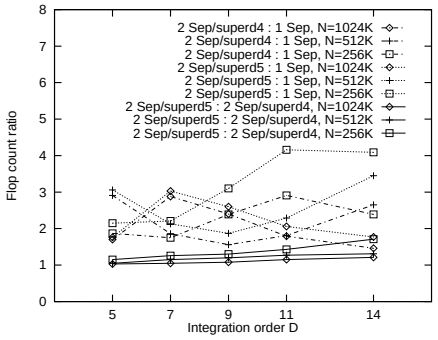


FIG. 5.12. *The FLOP count ratios* $FC_{2-Sep/superd4}/FC_{1-Sep}$, $FC_{2-Sep/superd5}/FC_1$, *and* $FC_{2-Sep/superd5}/FC_{2-Sep/superd4}$ *at optimal depth.*

the execution time of the method for uniformly distributed particles for a specified accuracy can be determined as follows:

1. For each combination of degree of separation and use of supernodes, find the required integration order D_i using equation (5.1). Look up the number of integration points K required.
2. Use the FLOP count to predict the optimal hierarchy depth, and calculate the total FLOP count at the predicted optimal hierarchy depth. The total FLOP count is a good predictor of the execution time.
3. Choose the method that has the smallest total FLOP count.

Using the above procedure, we compare 1–Sep with 2–Sep/superd4 and 2–Sep/superd5. Figure 5.17 plots the ratios of the FLOP counts of the three candidate methods as a function of desired error. Figure 5.18 plots the ratios of the execution times of the three candidate methods as a function of desired error. The two plots conform reasonably well, and suggest that the proposed parameter selection procedure works well. Similar to the FLOP count and time ratio comparison in Figures 5.9–5.16, the execution time ratio is again closer to one than the FLOP count ratio.

A surprising outcome of our simulations is that 1–Sep is almost always more cost–efficient than 2–Sep/super for any prespecified error, as shown in the two plots

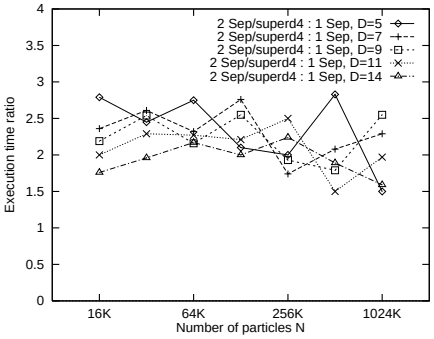


FIG. 5.13. *The execution time ratio* $T_{2-Sep/superd4}/T_{1-Sep}$ *at optimal depth.*

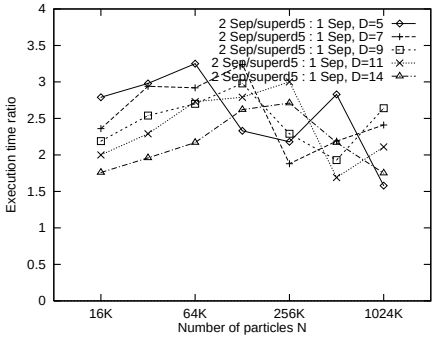


FIG. 5.14. *The execution time ratio* $T_{2-Sep/superd5}/T_{1-Sep}$ *at optimal depth.*

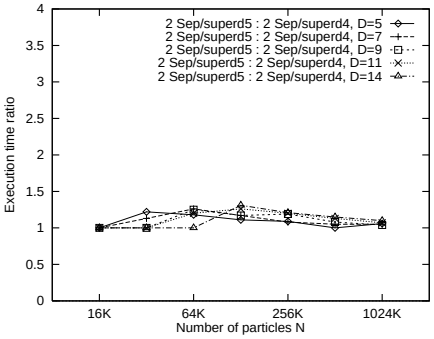


FIG. 5.15. *The execution time ratio* $T_{2-Sep/superd5}/T_{2-Sep/superd4}$ *at optimal depth.*

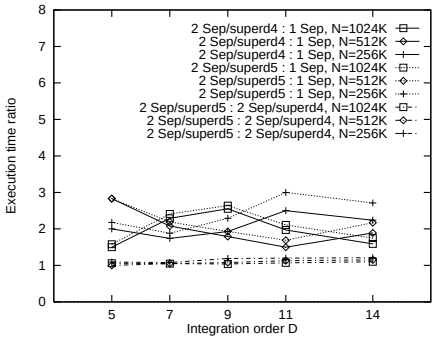


FIG. 5.16. *The execution time ratios* $T_{2-Sep/superd4}/T_{1-Sep}$, $T_{2-Sep/superd5}/T_{1-Sep}$ *and* $T_{2-Sep/superd5}/T_{2-Sep/superd4}$ *at optimal depth.*

above. The few cases when the ratios are larger than one is when 1–Sep has $D = 7$ and depth four, and 2–Sep/superd4 and 2–Sep/superd5 have $D = 5$ and depth five, and the former is more costly. A close look at the two methods tells us that the tradeoffs between them is mainly determined by the cost–accuracy tradeoffs of the integration formulas used in the outer/inner approximations in Anderson's method. As described in [1], the integration formulas, taken from [15], are highly efficient in the sense that they use minimum number of integration points to achieve a certain degree of integration accuracy. (The exact definition of the efficiency of an integration formula is defined in [15].) To achieve a given error, 1–Sep often requires one order higher integration order than that required by 2–Sep/superd4 and 2–Sep/superd5, i.e.

$$FC_{1-Sep}(D_{i+1}) = C_1 \cdot K_{i+1} \cdot N,$$
$$FC_{2-Sep/superd5}(D_i) = FC_{2-Sep/superd4}(D_i) = C_2 \cdot K_i \cdot N$$

where C_1 and C_2 are the constants of proportionality of the three methods, as listed in the last column of Table 2.5 (2–Sep/superd4 and 2–Sep/superd5 have about the same FLOP count as 2–Sep/superd3 for a hierarchy with more than four levels.) We observe that C_2/C_1 is close to two while K_{i+1}/K_i is often less than two. Therefore 1–Sep often has a smaller execution time than 2–Sep/superd4 or 2–Sep/superd5 for

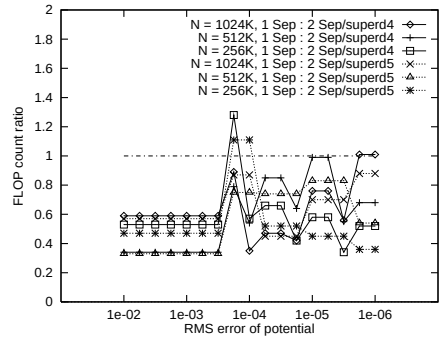


FIG. 5.17. *The FLOP count ratios* $FC_{1-Sep}/FC_{2-Sep/superd4}$ *and* $FC_{1-Sep}/FC_{2-Sep/superd5}$ *as a function of the RMS error in the potential.*

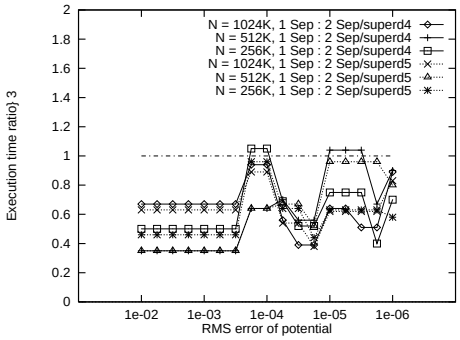


FIG. 5.18. *The execution time ratios* $T_{1-Sep}/T_{2-Sep/superd4}$ *and* $T_{1-Sep}/T_{2-Sep/superd5}$ *as a function of the RMS error in the potential.*

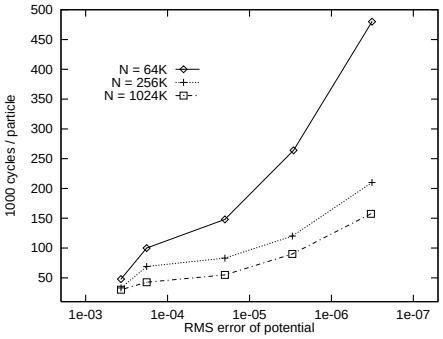


FIG. 5.19. *Cost vs. accuracy on a 32 node CM–5E for 1–Sep.*

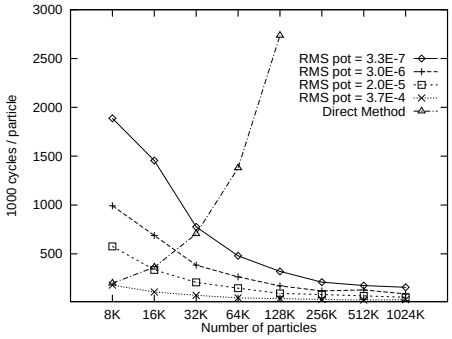


FIG. 5.20. *The breakeven points between Anderson's method with 1–Sep and the direct method for potential evaluation on a 32 node CM–5E.*

the same prespecified error.

Figures 5.19–5.20 show the tradeoffs between computational cost and the accuracy achieved using optimal parameters. The computation cost is represented by cycles/particle, and therefore relatively independent of the type and the size of the parallel machine. Figure 5.19 shows that for Anderson's method, higher accuracies can be achieved at increased computational cost. For small numbers of particles, the hierarchies are shallower and there are fewer boxes in the hierarchy. Therefore, the degree of parallelism is lower, the code becomes less efficient, and the cost per particle becomes higher. Figure 5.20 shows that for three digits of accuracy the method is competitive with direct N–body solvers at about 8,000 particles for three–dimensional problems, while for six digits of accuracy the break–even point is at about 35,000 particles.

6. On the choice of sphere radii in Anderson's method and impact of nonuniform distributions. In this section we investigate the impact of the choice of sphere radii on the accuracy in Anderson's method. We also study the impact of the particle distributions on the accuracy. All particle charges are assumed the same.

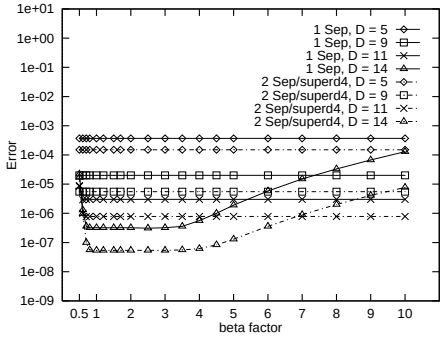


FIG. 6.1. *The RMS error of the potential for different outer sphere radii, N=256K.*

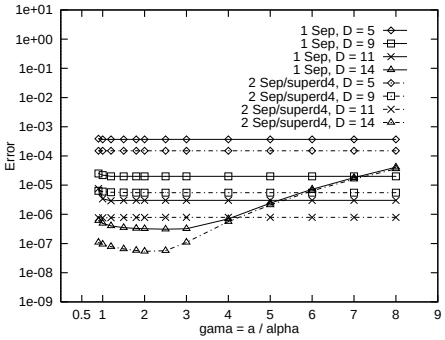


FIG. 6.2. *The RMS error of the potential for different inner sphere radii, N=256K.*

6.1. Uniform particle distribution. Anderson [1] suggested the radii of outer sphere and inner sphere listed in Table 3.1, and verified the parameters through evaluating a single sphere approximation. We verify the optimality of the parameters with respect to the accuracy of the complete method. All the simulations in this section are performed on a 64 node CM–5E.

Figure 6.1 plots the RMS error of the potential as a function of the scaling factor β_0 (defined in Table 3.1) for one–separation and two–separation with supernodes. It can be seen that for integration orders 5, 7, 9, and 11, the accuracy is invariant with respect to the outer sphere radius for the scaling factor ranging from one to ten. For integration order 14, the accuracy stays high when the scaling factor ranges from 0.8 to 3, but decreases outside this range. Figure 6.2 shows the RMS error of the potential as a function of the scaling factor γ_0 (defined in Table 3.1) for one–separation and two–separation with supernodes. Again, for integration orders 5, 7, 9, and 11, the accuracy of the method stays almost the same for γ_0 ranging from 1.2 to 8.0, while for integration order 14, the error is minimized for γ_0 between 2.0 and 3.0 for one–separation, and 2.0 and 2.5 for two–separation with supernodes. The range of optimal outer and inner sphere radii cover the values suggested in [1]. The unique behavior for integration order 14 could be attributed to limited floating–point precision. However, the same behavior is exhibited in 128–bit precision, which was experimentally verified using a Fortran–90 version of the code on DEC alphastations.

6.2. Error sensitivity to particle distributions. The accuracy of the hierarchical methods depend on the distributions of particles. For instance, if in Anderson's method the particles inside an outer sphere is clustered in one corner, then the magnitude of their projections to the integration points may vary significantly. In evaluating the outer sphere approximations, it is expected that the integration be less smooth and a relatively large numerical error may occur in summing up values of different magnitudes. On the other hand, the more clustered in the center of a leaf–level box the particles are, the more accurate the approximation offered by the computational element will be. The same arguments are true for GR' and Zhao's methods. In [4], Carrier et. al. found that with the same number of terms in the multipole expansions, the accuracy for different nonuniform distributions may vary up to an order of magnitude.

In this section, we investigate how the accuracy of Anderson's method with one–separation varies with small variations in the particle distributions. Specifically, for

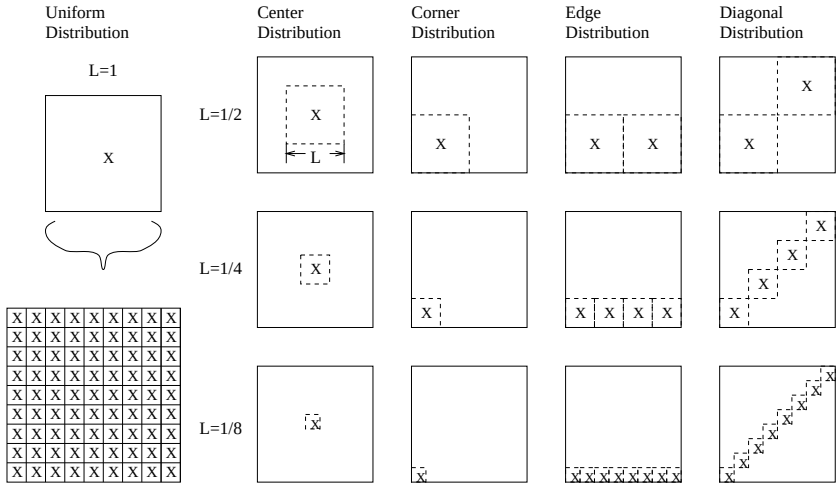


FIG. 6.3. *Particle distributions used to study the impact of particle distribution on the accuracy of Anderson's method.*

a given number of particles we first compute the optimal hierarchy depth as if the particle distribution was uniform. We then vary the relative locations of the particles inside each leaf–level box and study the error behavior. Since the number of particles in the leaf–level boxes remains unchanged, the optimal hierarchy depth stays the same. The four distributions we considered are shown in Figure 6.3:

- Center distribution, where the particles are uniformly distributed within a box of side length L centered at the center of the original leaf–level box;
- Corner distribution, where the particles are uniformly distributed within a box of side length L at one of the eight corners of the original leaf–level box;
- Edge distribution, where the particles are uniformly distributed within boxes of side length L located along one edge of the original leaf–level box;
- Diagonal distribution, where the particles are uniformly distributed within boxes of side length L located along one diagonal of the original leaf–level box.

Note that the smaller the imbalance factor L, the more imbalanced the distribution appears with respect to the uniform distribution. For $L = 1$, all four distributions degenerate to the uniform distribution. For any L and each of the four distributions, the distributions within leaf–level boxes are the same. Though the four distributions clearly are synthetic, they offer a controlled way of studying the impact of the particle distribution on the accuracy.

Figure 6.4 shows that the accuracy of the Center distribution is improved slightly if the particles are clustered to the center of the boxes. Figure 6.5 shows that changing the outer sphere radius has no impact for integration orders 5, 7, 9, and 11. But for integration order 14, it affects the accuracy of the center distribution in the same way it affects that of the uniform distribution. The same is true in changing the inner sphere radius as shown in Figure 6.6. Therefore, varying the outer and inner sphere radii does not affect the accuracy for the Center distribution in any other way than for the uniform distribution.

Figure 6.7 shows that the accuracy of the Corner distribution becomes worse as the particles are more and more clustered to a corner, especially for high integration

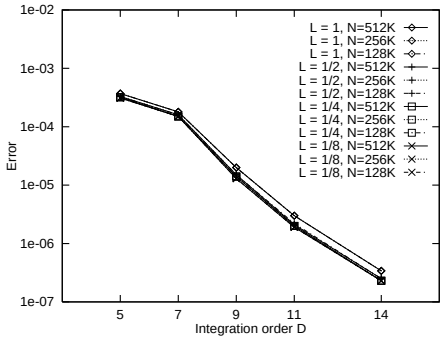


FIG. 6.4. *The RMS error of the potential as a function of the integration order D, Center distribution, 1–Sep.*

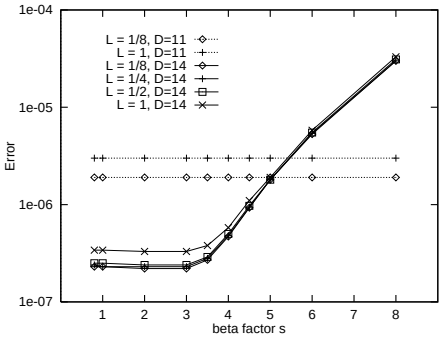


FIG. 6.5. *The RMS error of the potential as a function of the integration sphere radius (scaling factor β_0), Center distribution, 1–Sep, N=512K.*

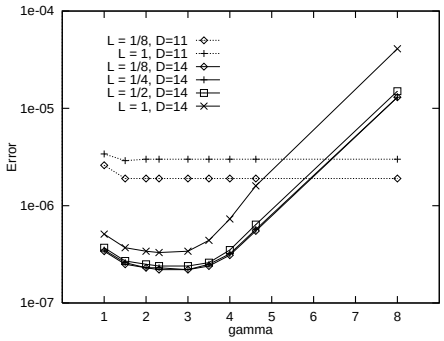


FIG. 6.6. *The RMS error of the potential as a function of the inner sphere radius, Center distribution, 1–Sep, N=512K.*

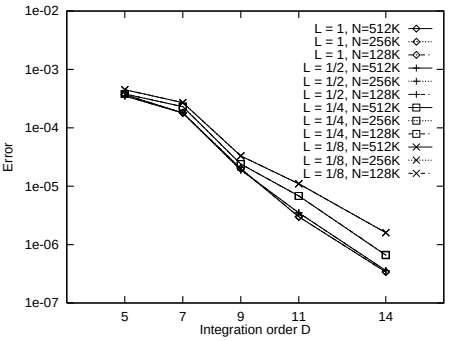


FIG. 6.7. *The RMS error of the potential as a function of the integration order D, Corner distribution, 1–Sep.*

orders. Again, changing the outer and inner sphere radii has almost no impact on the accuracy for integration orders 5, 7, 9, and 11. But for integration order 14, increasing the outer sphere radius improves the accuracy for L=1/8, as shown in Figure 6.8. In general, a large outer sphere smoothes the integration, but makes the coefficients of integration small in magnitude, which also may lead to accuracy problems due to round–off error. A good balance between the two effects are required for high accuracy. Increasing the outer sphere radius improves this balance for the Corner distribution. Similarly, decreasing inner sphere radius has the effect of smoothing the integration as the inner sphere is drawn away from the conner of particles, and therefore improves the accuracy for $\gamma_0 \approx 3.5$, as shown in Figure 6.9.

The Edge and Diagonal distributions are more balanced than the Corner distribution. It can be seen from Figures 6.10–6.11 that their accuracies are better than that for the Corner distribution for small L's. Figures 6.12–6.13 show the accuracies of the distributions as a function of the imbalance factor L. As expected, the Center distribution gives the best accuracy, followed by the Diagonal and Edge distributions. The Conner distribution yields the lowest accuracy. Only the errors for the 11th and 14th integration orders are shown, since the difference is much smaller for small integration orders.

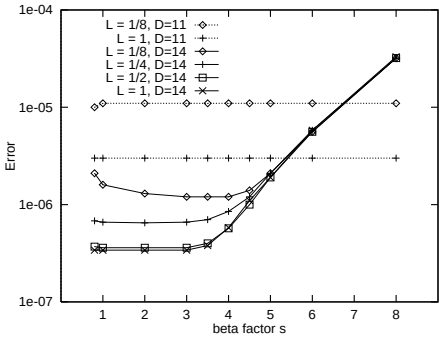


FIG. 6.8. *The RMS error of the potential as a function of the outer sphere radius, Corner distribution, 1–Sep, N=512K.*

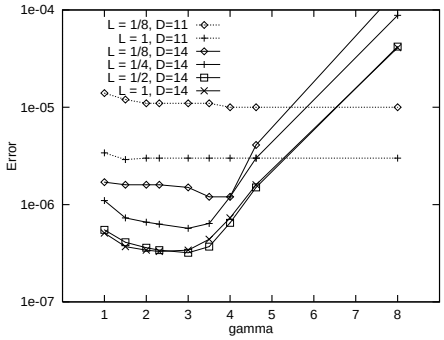


FIG. 6.9. *The RMS error of the potential as a function of the inner sphere radius, Corner distribution, 1–Sep, N=512K.*

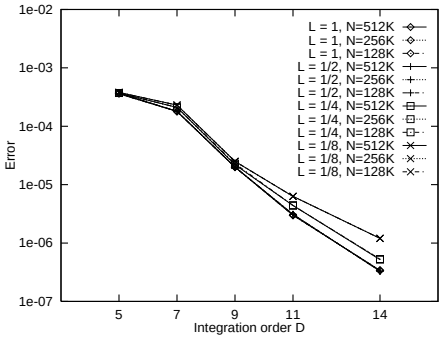


FIG. 6.10. *RMS error of potential varying D, Edge, 1–Sep*

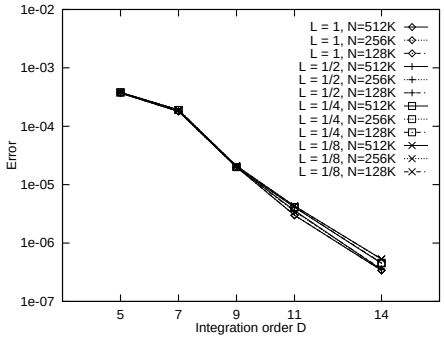


FIG. 6.11. *The RMS error of the potential as a function of the integration order D, Diagonal distribution, 1–Sep.*

7. Conclusions. Using our nonadaptive parallel implementation of Anderson's method, we have studied its accuracy–cost tradeoffs. Our studies show that in three–dimensions, using a near–field consisting of only nearest–neighbor boxes almost always minimizes the FLOP count required for a given accuracy, when the optimal hierarchy is used.

Our empirical study of the error sensitivity with respect to particle distributions shows that significant impact can be expected. For instance, by concentrating the particles uniformly into subdomain of size $(\frac{1}{8})^3$ in a corner of the leaf–level boxes, the error may increase by a factor of six or more. For highly nonuniform distributions enlarging the outer spheres and shrinking inner spheres may improve the accuracy for high integration orders. Such changes have the potential of better balancing the integration error (smoother function) with roundoff errors in small in magnitude coefficients.

Our study of the accuracy–cost tradeoff for hierarchical $O(N)$ N–body methods focused on Anderson's method. Empirical studies of Greengard–Rokhlin's method similar to ours would allow a more direct comparison of the computational efforts of the two methods for a given accuracy. We believe it to be very feasible to develop error and execution time models similar to ours for Greengard–Rokhlin's method. A direct comparison would be most interesting.

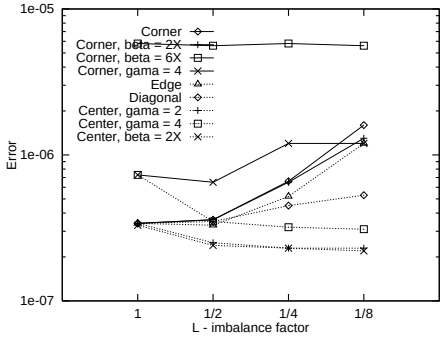


FIG. 6.12. *The RMS error of the potential as a function of L, K = 14, 1–Sep, N=512K.*

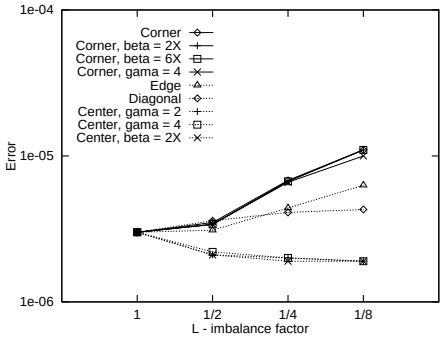


FIG. 6.13. *The RMS error of the potential as a function of L, K = 11, 1–Sep, N=512K.*

REFERENCES

[1] C. R. Anderson. An implementation of the fast multipole method without multipoles. *SIAM J. Sci. Stat. Comput.*, 13(4):923–947, July 1992.

[2] J. Barnes and P. Hut. A hierarchical $O(N \log N)$ force calculation algorithm. *Nature*, 324:446–449, 1986.

[3] A. Brandt. Multilevel computations: Review and recent developments. In S. McCormick, editor, *Proc. Third Copper Mountain Conference on Multigrid Method*, Copper Mountain, Colorado, April 1987.

[4] J. Carrier, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm for particle simulations. *SIAM J. Sci. Statist. Comput.*, July 1988.

[5] K. Esselink. A comparison of algorithms for long range interactions. Technical Report 500.90.200GRA204-IN73916 AMER.94.006, Koninklijke/Shell-Laboratorium, Amsterdam, August 1994.

[6] L. Greengard and W. D. Gropp. A parallel version of the fast multipole method. In *Parallel Processing for Scientific Computing*, pages 213–222. SIAM, Philadelphia, 1989.

[7] L. Greengard and V. Rokhlin. A fast algorithm for particle simulation. *Journal of Computational Physics*, 73:325–348, 1987.

[8] L. F. Greengard and V. Rokhlin. Rapid evaluation of potential fields in three dimensions. Technical Report YALEU/DCS/RR–515, Dept. of Computer Science, Yale Univ., February 1987.

[9] L. F. Greengard and V. Rokhlin. On the efficient implementation of the fast multipole algorithm. Technical Report YALEU/DCS/RR–602, Dept. of Computer Science, Yale Univ., February 1988.

[10] Y. Hu and S. L. Johnsson. A data parallel implementation of hierarchical N–body methods. *International Journal of Supercomputing Applications and High Performance Computing*, 10(1):3 – 40, 1996.

[11] Y. Hu and S. L. Johnsson. Implementing $O(N)$ N–body algorithms efficiently in data–parallel languages. *Journal of Scientific Programming*, 5(4):337 – 364, 1996.

[12] J. Katzenelson. Computational structure of the N–body problem. *SIAM J. Sci. Statist. Comput.*, 10(4):787–815, 1989.

[13] J. F. Leathrum. *The parallel fast multipole method in three dimensions.* PhD thesis, Duke University, 1992.

[14] A. McKenney. Implementation issues for fast multipole implementations for molecular dynamics simulations. manuscript, 1996.

[15] A. D. McLaren. Optimal numerical integration on a sphere. *Math. Comput.*, 17:361 – 383, 1963.

[16] K. E. Schmidt and M. A. Lee. Implementing the fast multipole method in three dimensions. *J. Stat. Phy.*, 63(5/6), 1991.

[17] J. Shimada, H. Kaneko, and T. Takada. Performance of fast multipole methods for calculating electrostatic interactions in biomacromolecular simulations. *Journal of Computational Chemistry*, 15(1):28–43, 1994.

[18] Thinking Machines Corp. *CM Fortran Reference Manual, Version 2.1*, 1993.

[19] Thinking Machines Corp. *CMSSL for CM Fortran, Version 3.1*, 1993.

[20] M. S. Warren and J. K. Salmon. A portable parallel particle program. *Computer Physics Communications*, 87, 1995.

[21] F. Zhao. An $O(N)$ algorithm for 3–dimensional N–body simulations. Technical Report AI–TR–995, AI Lab, MIT, 1987.

[22] F. Zhao and S. L. Johnsson. The parallel multipole method on the Connection Machine. *SIAM J. Sci. Statist. Comput.*, 12(6):1420–1437, Nov. 1991.