SIAM J. SCI. COMPUT.
Vol. 21, No. 5, pp. 1790–1803

IMPROVING ERROR BOUNDS FOR MULTIPOLE-BASED TREECODES[*]

ANANTH GRAMA[†], VIVEK SARIN[‡], AND AHMED SAMEH[†]

Abstract. Rapid evaluation of potentials in particle systems is an important, time-consuming step in many physical simulations. Over the past decade, the development of treecodes, such as the fast multipole method and Barnes–Hut method, has enabled large scale simulations in astrophysics, molecular dynamics, material science, etc. These methods use fixed-degree polynomial approximations of the potential at a point, due to a set of particles in a hierarchical representation of the particle system. In this paper, we present analysis and experiments to illustrate that fixed-degree multipole approximations can lead to large aggregate errors. We describe an alternate strategy based on careful selection of the multipole degree that leads to asymptotically lower errors while incurring minimal computation overhead. First, we estimate the error associated with each particle-cluster interaction and the aggregate error for each particle. Then we describe a technique for computing the multipole degree of each interaction with a view to reducing aggregate error, and establish the computational complexity of the new method. Following this, we discuss numerical experiments that demonstrate favorable error properties of the new method at the expense of marginal increase in computation. Finally, we report a parallel implementation on the SGI Origin 2000 that achieves excellent speedup for problems from domains such as astrophysics and boundary element solvers.

Key words. Barnes–Hut, fast multipole method, integral equations, boundary elements, particle dynamics, parallel computation

AMS subject classifications. 65N38, 65R20, 65Y05, 65Y20, 70F10, 70-08

PII. S1064827598339128

1. Introduction.

The problem of evaluating the potential due to a set of particles is an important and time-consuming one. The development of fast treecodes such as the Barnes–Hut and fast multipole methods for n-body systems has enabled large scale simulations in astrophysics [11, 12, 15] and molecular dynamics [3]. Coupled with efficient parallel processing, these treecodes are capable of yielding several orders of magnitude improvement in performance [8, 16, 17]. In addition, treecodes have applications in the solution of dense linear systems arising from boundary element methods [5, 6, 7, 13, 14].

The all-to-all nature of interactions in typical particle systems implies that an accurate formulation of the n-body problem has an $O(n^2)$ complexity for an n particle system. This complexity can be reduced by exploiting the decaying nature of the interaction between bodies. For example, in astrophysical simulations, distant galaxies can be viewed as point masses placed at their centers of mass. Many fast algorithms use this principle to accelerate n-body simulations.

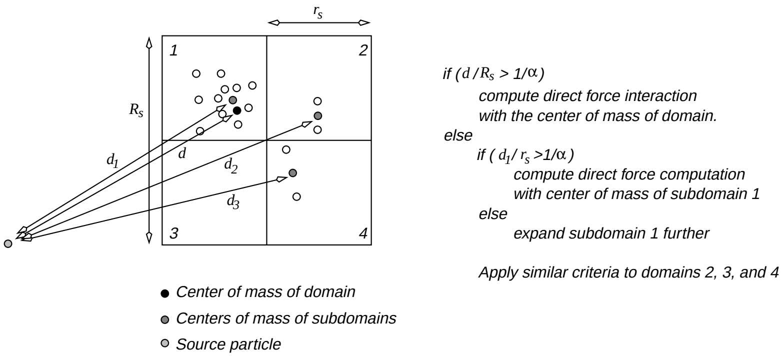
The Barnes–Hut method is one of the most popular methods due to its simplicity. It works in two phases: the tree construction phase and the force computation phase. In the tree construction phase, a spatial tree representation of the domain is derived.

[*] Received by the editors May 19, 1998; accepted for publication (in revised form) March 10, 1999; published electronically April 28, 2000. This work is supported in part by National Science Foundation grants ESC-9527123, EIA-9806741, ACI-9875899, and ACI-9872101. Computational resources were provided by NCSA at the University of Illinois and the Minnesota Supercomputing Center.

http://www.siam.org/journals/sisc/21-5/33912.html

[†] Department of Computer Sciences, Purdue University, West Lafayette, IN 47907-1398.

[‡] Department of Computer Science, Texas A&M University, College Station, TX 77843-3112 (sarin@cs.tamu.edu).



```
if ($d/R_s > 1/\alpha$)
      compute direct force interaction
      with the center of mass of domain.
else
      if ( $d_1/r_s > 1/\alpha$)
            compute direct force computation
            with center of mass of subdomain 1
      else
            expand subdomain 1 further

      Apply similar criteria to domains 2, 3, and 4
```

FIG. 1.1. *Illustration of the Barnes–Hut method in two dimensions.*

At each step in this phase, if the domain contains more than s particles (for some preset constant s), it is recursively divided into eight equal parts (four parts in two dimensions). This process continues until each part has at most s elements. The resulting tree is an unstructured oct-tree (quad-tree in two dimensions). Each internal node in the tree computes and stores an approximate multipole series representation of the particles contained in that subtree. For astrophysical simulations, this is often approximated by the center of mass of the particles contained in the tree. Once the tree has been constructed, the force or potential at each particle can be computed as follows: a multipole acceptance criterion is applied to the root of the tree to determine if an interaction can be computed; if not, the node is expanded and the process is repeated for each of the four (or eight) children. The multipole acceptance criterion for the Barnes–Hut method computes the ratio of the distance of the point from the center of mass of the box to the dimension of the box. If this ratio is greater than $1/\alpha$, where α is a constant less than unity, an interaction can be computed. The Barnes–Hut method is illustrated in Figure 1.1.

For a balanced tree, each of the n particles needs $O(\log n)$ interactions. However, the tree can be made arbitrarily large by bringing a pair of particles successively closer. The corresponding tree needs a large number of boxes to resolve the pair into separate boxes. Due to this, the worst case complexity of this technique is unbounded [1, 4]. This complexity can be reduced by using box-collapsing techniques (the box is first collapsed to the smallest box that contains all the particles in the subdomain). In addition, there are some recent results demonstrating that it is beneficial to work with binary trees instead of higher order trees [2]. Binary trees with controlled split allow better aspect ratios for partitions while reducing the number of nodes in the tree.

The fast multipole method (FMM) of Greengard and Rokhlin [10] is another hierarchical technique for computing n-body interactions. Unlike the Barnes–Hut method, the FMM computes potential instead of force (force can be computed as the gradient of the potential). Depending on the application, this potential may be electrostatic, gravitational, etc. The FMM computes the potential due to a cluster of particles at the center of other well-separated clusters. This can then be disseminated to individual particle positions to determine required potentials. In other words, the FMM computes cluster-cluster interactions in addition to particle-cluster interactions. For uniform distributions, the computational complexity of the FMM was originally shown to be $O(n)$. For arbitrary distributions, Callahan and Kosaraju [4] have shown that the complexity of the potential estimation phase can be reduced to $O(n)$ as well, after computing well-separated sets in a preprocessing step.

The dominant cost in both of these treecodes is the potential (or force) estimation phase. In the Barnes–Hut method, each particle-cluster interaction based on a p-term multipole requires $O(p^2)$ operations. For arbitrary distributions, this gives a complexity of $O(p^2W)$, where W is the total number of interactions. For the Barnes–Hut method on uniform distributions, $W = O(n \log n)$. The cluster-cluster interactions of the the FMM based on a p-term multipole take $O(p^4)$ operations. For a uniform distribution, the FMM has a complexity of $O(p^4n)$.

This paper is mainly concerned with the errors in these methods. We describe an algorithm that reduces the error significantly by selecting the multipole degree appropriately for different clusters. Furthermore, we show that, for practical problem sizes, this increases the computational complexity marginally. This is also illustrated via numerical experiments in the context of particle simulations as well as boundary element methods. Our IEEE portable operating system interface (POSIX) threads-based treecode has also shown excellent speedups on a 32 processor SGI Origin 2000, even for relatively small problems.

The paper is organized as follows. In section 2, we discuss error bounds on the Barnes–Hut method, outline the improved method, and analyze its error bounds. Experiments illustrating the accuracy and complexity of the new method are presented in section 3. Section 4 describes the parallel formulation of the methods, and section 5 presents the conclusions.

2. Improving error in the Barnes–Hut method.

The potential due to a set of charges located within a sphere of radius r_s at an observation point at distance r from the origin can be expressed as a multipole series. The error in a truncated multipole series of degree p was first derived by Greengard and Rokhlin [9, 10]. The following theorem from [9] describes the multipole expansion and the associated error.

THEOREM 2.1 (multipole error). *Suppose that k charges $\{q_j,\ j = 1, \dots, k\}$ are located at the points $\{P_j = (\rho_j, \theta_j, \psi_j)\}$ (in spherical coordinates), with $|\rho_j| < r_s$. Then for any point $P = (r, \theta, \psi) \in \mathbb{R}^3$ with $r > r_s$, the potential $\phi(P)$ is given by*

$$\phi(P) = \sum_{n=0}^{\infty} \sum_{m=-n}^{m=n} \frac{M_n^m}{r^{n+1}} \cdot Y_n^m(\theta, \psi),$$

where

$$M_n^m = \sum_{j=1}^{k} q_j \cdot \rho_j^n \cdot Y_n^{-m}(\theta_j, \psi_j)$$

and

$$Y_n^m(\theta, \psi) = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} \cdot P_n^{|m|}(\cos\theta) e^{im\psi},$$

in which $P_n^{|m|}(\cos\theta)$ are the associated Legendre functions. Furthermore, for any $p \geq 1$, the error in the truncated multipole series of degree p is given by

$$\epsilon = \left| \phi(P) - \sum_{n=0}^{p} \sum_{m=-n}^{m=n} \frac{M_n^m}{r^{n+1}} \cdot Y_n^m(\theta, \psi) \right|$$

and is bounded by

$$\epsilon \leq \frac{A}{r - r_s} \left(\frac{r_s}{r} \right)^{p+1}, \tag{2.1}$$

where $A = \sum_{j=1}^{k} |q_j|$.

Proof. See [9, p. 54]. ▯

In the Barnes–Hut method, the potential at a point is computed as a sum of the contributing potentials from clusters of particles. An interaction with a cluster is computed only if the point is *well-separated* from the cluster. This is enforced using a multipole acceptance criterion such as the α-criterion which requires that the ratio of the distance between the point and the center of mass of the cluster and the dimension of the box enclosing the cluster be greater than $1/\alpha$ for a constant $\alpha < 1$. Theorem 2.1 can be used to estimate the error in the Barnes–Hut method.

Theorem 2.2 (the Barnes–Hut multipole error). *Suppose that* k *charges of strengths* $\{q_j, j = 1, \dots, k\}$ *are located within a sphere of radius* r_s*. Then, for the Barnes–Hut method with* α*-criterion, the error in the truncated multipole series approximation of the potential at a distance* r *is bounded by*

$$\epsilon < \frac{A}{r_s} \cdot \frac{\alpha^{p+2}}{1 - \alpha},$$

where $p \geq 1$ *and* $A = \sum_{j=1}^{k} |q_j|$.

Proof. The α-criterion of the Barnes–Hut method ensures that $r/r_s > 1/\alpha > 1$. Subtracting unity and inverting, we obtain the following relation:

$$\frac{r_s}{r - r_s} < \frac{\alpha}{1 - \alpha}.$$

From Theorem 2.1,

$$\epsilon \leq \frac{A}{r_s} \cdot \frac{r_s}{r - r_s} \cdot \left(\frac{r_s}{r} \right)^{p+1} < \frac{A}{r_s} \cdot \frac{\alpha}{1 - \alpha} \cdot \alpha^{p+1},$$

which proves the theorem. ▯

This theorem highlights the main problem with aggregate error in the Barnes–Hut method. The error grows significantly with the net charge of the particle clusters. Moreover, the size of the largest cluster with which an interaction is computed can be shown to be within a constant factor of the total simulation domain. Thus, the aggregate error can be large, and even unbounded, for unstructured distributions. For instance, in applications such as protein simulations, the charge density is largely uniform across the domain of simulation; therefore, the overall error in the Barnes–Hut method grows significantly with the magnitude of charge in the system. In general, for large simulation domains, the aggregate error may become unacceptable.

Fortunately, this theorem also suggests an alternative strategy to control the error. By increasing the polynomial degree p for clusters with increased net charge A, the error in each interaction can be restricted to a constant value. The error in potential at each point would then be proportional to the total number of interactions. The multipole acceptance criterion along with the hierarchical decomposition of the domain can be used to establish the following:

- the number of interactions with clusters of a particular size are bounded by constant, and

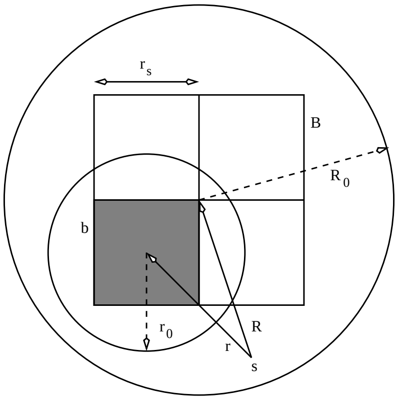


FIG. 2.1. *Establishing bounds on r_s/r in the Barnes–Hut method.*

- the number of distinct sizes of clusters is equal to the height of the decomposition tree.

For structured distributions with uniform charge density, this translates to $O(\log n)$ aggregate error. Next, we prove the first assertion and then outline an improved algorithm for selecting the polynomial degree p to restrict the error.

In order to bound the number of interactions for fixed size clusters, we first establish limits on the ratio r_s/r.

LEMMA 2.3. *In the Barnes–Hut method, the ratio r_s/r for particle-cluster interactions is bounded as follows:*

$$\alpha' < \frac{r_s}{r} < \alpha,$$

where α' and α are constants, such that

$$\alpha' = \left(\frac{2}{\alpha} + \frac{1}{\sqrt{2}}\right)^{-1}.$$

Proof. The interaction of particle s with box b indicates that s could not interact with its parent box B based on the α-criterion (see Figure 2.1). Therefore,

$$r > r_0 \text{ and } R < R_0,$$

where $r_0 = r_s/\alpha$ and $R_0 = 2r_s/\alpha$. Using the triangle inequality $R + r_s/\sqrt{2} \geq r$, it can be shown that

$$\left(\frac{2}{\alpha} + \frac{1}{\sqrt{2}}\right)^{-1} < \frac{r_s}{r} < \alpha,$$

which completes the proof. □

As α is reduced, this bound tends to $\alpha/2 < r_s/r < \alpha$, indicating a tight bound. It is now easy to show that the number of interactions with a box of size r_s is bounded by a constant.

LEMMA 2.4. *In the Barnes–Hut method, a particle interacts with a bounded number of boxes of a given size.*

Proof. Lemma 2.3 shows that $\alpha' < r_s/r < \alpha$; therefore, the centers of all boxes of size r_s lie within an annular region defined by the following relation:

$$\frac{r_s}{\alpha} < r < \frac{r_s}{\alpha'},$$

and the boxes themselves lie completely within the annular region:

$$\frac{r_s}{\alpha} - \frac{r_s}{\sqrt{2}} < r < \frac{r_s}{\alpha'} + \frac{r_s}{\sqrt{2}}.$$

The ratio of the volume of this annular region and the volume of a single box gives an upper bound on the number of boxes of size r_s. For a three-dimensional problem,

$$\begin{aligned} n_{max} &< \frac{4\pi}{3}\left[\left(\frac{1}{\alpha'} + \frac{1}{\sqrt{2}}\right)^3 - \left(\frac{1}{\alpha} - \frac{1}{\sqrt{2}}\right)^3\right] \\ &= \frac{4\pi}{3}\left[8\left(\frac{1}{\alpha} + \frac{1}{\sqrt{2}}\right)^3 - \left(\frac{1}{\alpha} - \frac{1}{\sqrt{2}}\right)^3\right], \end{aligned}$$

where n_{max} is the maximum number of boxes of a fixed size interacting with any particle. □

The polynomial degree p needs to be selected for each particle-cluster interaction in order to restrict the error. The next theorem shows how to determine the multipole degree to keep interaction error constant.

THEOREM 2.5 (variable degree multipoles). *The polynomial degree p_k required for a particle-cluster interaction for constant error is given by*

$$p_k = p_0 + k \log_\alpha 2 + \log_\alpha \frac{A_0}{A_k},$$

where A_k is the net charge on the cluster at level k and A_0 is the smallest net charge cluster at lowest level in the tree.

Proof. Let b_j be a cluster of particles in a box of size r_{sj} at the jth level of the tree. To fix the error for clusters at different levels in the tree, we force the bound on an error (Theorem 2.2) to be equal:

$$\frac{A_k}{r_{sk}} \cdot \frac{\alpha^{p_k+2}}{1-\alpha} = \frac{A_j}{r_{sj}} \cdot \frac{\alpha^{p_j+2}}{1-\alpha}$$

for the pair of clusters b_j and b_k. This simplifies to

$$\frac{A_j}{A_k} \cdot \frac{r_{sk}}{r_{sj}} = \alpha^{p_k - p_j},$$

where $r_{sk}/r_{sj} = 2^{k-j}$. The theorem follows from the choice of b_j as the smallest net charge cluster at lowest level. □

In general, we select a minimum degree of interaction p_0 associated with a threshold value A_0 and increase multipole degree for larger cluster sizes. For structured domains, it is easy to control the polynomial degree in this manner since increase in the polynomial degree is not large. The multipole series are computed a priori to the maximum required degree for each cluster. Clearly, this is possible since all parameters for the degree of an interaction are available at the time of tree construction.

However, this technique can result in very large degree multipoles for unstructured domains. This difficulty is overcome by

(1) altering the α-criterion of the Barnes–Hut method,
(2) computing and storing the increased degree multipoles, or
(3) using alternate height-balanced tree construction schemes.

This paper is primarily concerned with uniform distributions; however, our experiments demonstrate that the paradigm works for unstructured domains as well.

We now examine the error associated with the Barnes–Hut method with this improved multipole degree selection criteria.

Theorem 2.6 (improved Barnes–Hut error). *The error in the improved Barnes–Hut method for structured distributions with uniform charge density is $O(\alpha^{p+1}\log A)$.*

Proof. Observe that the number of particle-cluster interactions with fixed size clusters is bounded (Lemma 2.4), the number of distinct sizes of clusters equals the height of the decomposition tree ($O(\log n)$ for structured distributions), and the error associated with each interaction is constant (Theorems 2.2 and 2.5). From this it can be concluded that the error for uniform charge density is $O(\alpha^{p+1}\log n)$. The proof follows directly from the observation that for uniform charge density, $\log n$ is equivalent to $\log A$. □

The reader will note that this error is considerably less than the error bound on the original Barnes–Hut method with fixed-degree multipoles. The last issue to be resolved is the increase in computation introduced by the higher degree multipole evaluations. The final theorem shows that this additional computation is minimal.

Theorem 2.7 (complexity). *For structured particle distributions with uniform charge density, the computational complexity of the piecewise approximate Barnes–Hut method is $O(n\left(\lceil \frac{p}{c}\rceil + l\right)^3)$, where $l = \log_8 n$ is the number of levels of the hierarchical decomposition, p is the smallest degree, and $c = (1-d)\log_\alpha 2$ for the d-dimensional domain.*

Proof. For each particle, we need to compute at most n_{max} interactions with p_k degree multipole for levels $k = 0, \ldots, l$. The total computation is proportional to $n \cdot n_{max}\sum_{k=0}^{l} p_k^2$, where $p_k = p + ck$ for uniform charge density (Theorem 2.5). Let us define $q = \lceil \frac{p}{c} \rceil$. Now,

$$\sum_{k=0}^{l}(p+ck)^2 \le c^2\sum_{k=0}^{l}(q+k)^2.$$

The right-hand side of the above equation can be bounded as follows:

$$\sum_{k=0}^{l}(q+k)^2 = \sum_{j=q}^{q+l} j^2 \le \sum_{j=1}^{q+l} j^2 = \frac{1}{3}(q+l)^3 + \frac{1}{2}(q+l)^2 + \frac{1}{6}(q+l).$$

Since $q+l \ge 1$, $(q+l)^3 \ge (q+l)^2 \ge (q+l)$, and thus, the expression on the right-hand side of the above equation is bounded by $(q+l)^3$, i.e.,

$$\sum_{k=0}^{l}(q+k)^2 \le (q+l)^3.$$

Therefore,

$$\sum_{k=0}^{l}(p+ck)^2 \le c^2(q+l)^3.$$

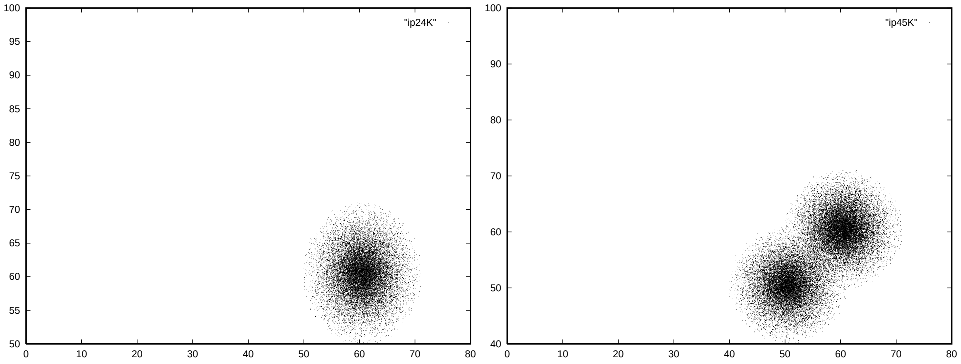


FIG. 3.1. *Sample distributions for experiments:* (a) *Gaussian (*24,000 *particles) and* (b) *Overlapped Gaussians (*45,000 *particles).*

The theorem follows from the observation that c is a constant depending only on the dimensions d and α. □

This result can be extended to unstructured distributions as well using the box-collapsing and flexible splitting techniques of Callahan and Kosaraju [4]. It is useful to note that the complexity of the original Barnes–Hut method grows as $O(p^2 n \log n)$. The number of levels in a uniform distribution l grows as $\log_8 n$, assuming a single particle per leaf cell. For typical values of p (6–7 degree approximations), this corresponds to 256,000–2,000,000 particles. In order to optimize cache performance and to obtain lower algorithmic constants, leaf nodes of the tree often represent clusters of up to 32 or 64 particles. This increases the number of particles to between 8 and 64 million. Thus, even for very large scale simulations, the improved method is within a small constant off the fixed-degree method. In general, for $l \leq p$, the complexity of the improved method is within a small factor of the original method. This is borne out by the experiments in section 3. As a result, the improved method yields significant improvements in error while incurring minimum additional overhead.

3. Experiments.

The treecode was tested in the context of particle simulations as well as boundary element solvers. Problem instances for particle simulations range from uniform to highly irregular distributions in three dimensions. Uniform distributions correspond to a random distribution of points distributed uniformly across the domain. Irregular distributions are generated using a Gaussian density function or overlapped Gaussian distributions (multiple Gaussians superimposed). Figure 3.1 illustrates examples of the two distributions. The number of particles in the test cases range from 20,000–100,000. The sizes of problem instances were selected to facilitate specific experiments.

The error in the multipole approximation is defined as

$$\epsilon = \max_{1 \leq i \leq n} |v_i - \tilde{v}_i|,$$

where v and $\tilde{v}$ are the vectors of accurate and approximate potentials, respectively.

3.1. Performance and accuracy of the improved method.

The comparison of the improved method to the original method is done on the basis of the number of multipole term evaluations. For a p degree approximation, this is proportional to p^2 for Barnes–Hut and has been found to be an excellent indication of the serial runtime of the method. In contrast, other metrics such as wall clock time often suffer

TABLE 3.1
Comparison of the improved method with the original method.

Structured distributions

n	ϵ_{orig}	$\epsilon_{improved}$	Levels (l)	Terms(orig)	Terms(improved)
$\alpha = 0.707, p = 6$					
10000	0.012065	0.010237	4	12×10^6	14×10^6
20000	0.025872	0.010121	5	60×10^6	68×10^6
40000	0.053976	0.011887	5	127×10^6	155×10^6
60000	0.078553	0.014987	5	190×10^6	241×10^6
80000	0.098279	0.014339	6	254×10^6	335×10^6
$\alpha = 0.77, p = 7$					
10000	0.006912	0.006567	4	14×10^6	17×10^6
20000	0.017104	0.006632	5	68×10^6	79×10^6
40000	0.027338	0.008215	5	144×10^6	180×10^6
60000	0.045432	0.010700	5	216×10^6	281×10^6
80000	0.058060	0.009193	6	289×10^6	393×10^6

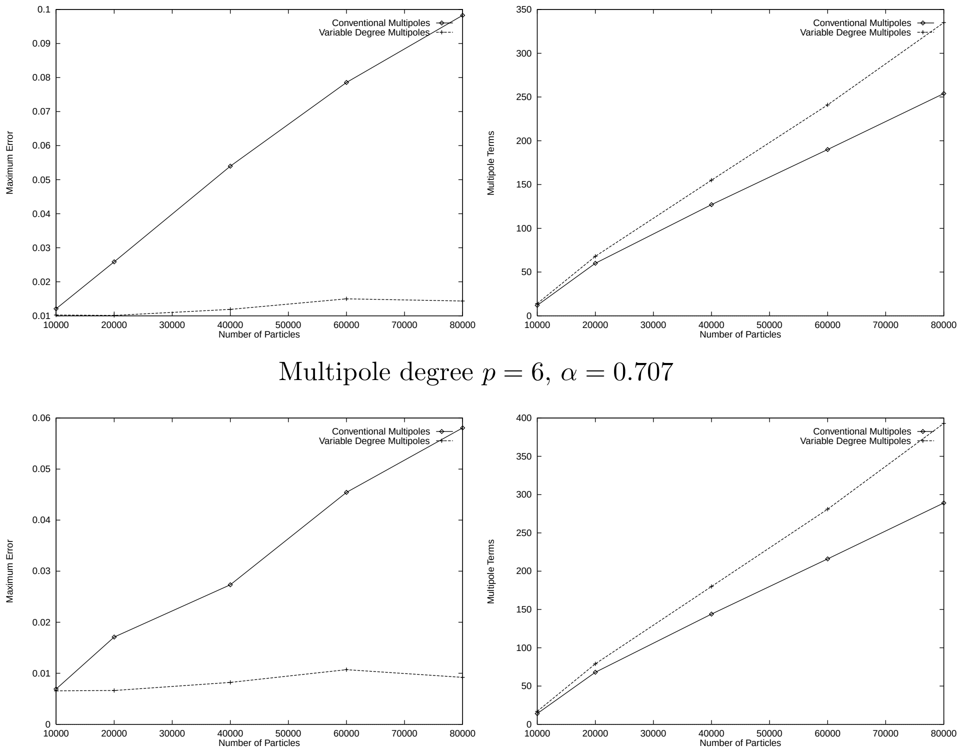
Unstructured distributions

n	ϵ_{orig}	$\epsilon_{improved}$	Terms(orig)	Terms(improved)
$\alpha = 0.707, p = 6$				
45000	2.479027	0.302205	70×10^6	108×10^6
82000	2.307508	0.221419	179×10^6	239×10^6

from discrepancies introduced by other processes on the machine, parallel efficiency, network congestion, etc.

Table 3.1 illustrates the error and the number of term expansions for two instances with $\alpha = 0.707, p = 6$ and $\alpha = 0.77, p = 7$ for the variable and fixed-degree multipoles (see also Figure 3.2). The following can be inferred from these experiments:

- The error in the variable degree multipole method grows much slower than the fixed-degree multipole method. This growth rate is expected to be logarithmic in the number of particles (Theorem 2.6). A careful look at the table reveals that for a fixed number of levels the error grows with the number of particles at a slightly faster rate. This is because in these experiments each leaf in the hierarchy has a maximum of 32 particles. The error associated with a single multipole interaction with the parent of a leaf node is considered the baseline error (all leaf node interactions are carried out as direct interactions as opposed to multipole interactions). Errors for higher level interaction are equated with this error in the variable degree multipole method. As the number of particles is increased for a fixed number of levels (for example from 20,000–60,000 particles), the number of particles in an average leaf node increases from 5 to 15. The corresponding increase in the baseline error node is from 40–120 particles. This contributes to an increase in the total error as the number of particles are increased for a fixed number of levels. When an additional level is added (at 80,000 particles), the error becomes lower again. Thus, local deviations from logarithmic growth are due to multiple particles per leaf node.
- The computation associated with variable degree multipoles is slightly higher than that of fixed-degree multipoles. For these computations, with $\alpha = 0.707$, the value of c in Theorem 2.7 is equal to 2.8. The excellent performance of



Multipole degree $p = 6$, $\alpha = 0.707$

Multipole degree $p = 7$, $\alpha = 0.769$

FIG. 3.2. *Comparison of the improved method with the original method for structured distributions.*

the variable-degree multipole method is due to the fact that the number of levels l is comparable to the degree of multipole approximations. With 32–64 particles per leaf node, this holds true for most realistic simulations with multipole approximations of 6 or 7 degrees.

Comparison to fixed-degree multipoles. To assess the merit of variable degree multipoles, we compare the number of multipole terms expanded by the new method with a higher fixed-degree multipole approximation. The objective is to demonstrate the savings in computation while holding the error nearly constant. In this experiment, we keep the value of α fixed so that the number of near field potential computations is identical. We present the results of this experiment in Table 3.2. The fixed degree of the original method is increased until the error is nearly identical to the error in the variable multipole method. It is clear from the table that as the number of particles is increased, the fixed-degree method needs much higher multipole approximations to match the error. Consequently, the computation associated with them is higher. For example, for an 80K particle system, fixed-degree multipoles take over 50% more computation and incur slightly higher errors than variable degree multipoles. This difference increases with the number of particles, clearly demonstrating that variable degree multipoles can lead to significant reduction in computation for a given error bound.

TABLE 3.2

Performance gains from variable degree multipoles over fixed-degree multipoles for similar error tolerance.

n	$\epsilon_{improved}$	ϵ_{orig}	deg(orig)	Terms(improved)	Terms(orig)
10000	0.006567	0.006912	7	17×10^6	14×10^6
20000	0.006632	0.013714	8	79×10^6	89×10^6
40000	0.008215	0.007069	9	180×10^6	238×10^6
60000	0.010700	0.010933	9	281×10^6	357×10^6
80000	0.009193	0.012614	10	393×10^6	589×10^6

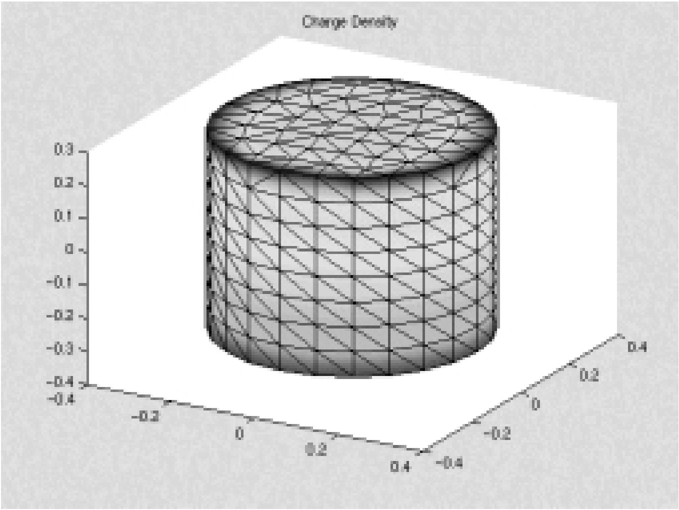


FIG. 3.3. *Charge density on the surface of a cylinder with one end fixed at zero potential and the other end at unit potential.*

3.2. Dense solvers. The treecode can be used to solve dense linear systems arising from boundary element methods for solving integral equations. In particular, the treecode was used to compute matrix-vector products with the approximation of the dense matrices in each iteration of the GMRES iterative solver. In general, the coefficient matrix is generated by the Green's function of the Laplace's equation and takes the form of $-\log r$ in two dimensions and $1/r$ in three dimensions. Such problems arise in the computation of charge density, given a potential distribution over a conductor, or for computing heat flows. The surface of the domain is discretized into triangular elements. Gaussian quadrature is used for integration over the surface. Typically, a fixed number of Gauss-points are located inside each element and inserted into the hierarchical domain representation. Using this hierarchical domain, the potential is computed at the vertices of the elements and matched to the boundary values. This process forms a single matrix-vector product that is required at each step of GMRES.

We use this technique to solve charge distribution problems on complex three dimensional geometries. In Figure 3.3, we illustrate a simple problem instance with 572 elements and 288 nodes. The corresponding dense linear system has 288 unknowns (one for each unknown charge density). The performance of our code was validated on three larger problems: propeller (140,800 elements, 70,439 nodes), gripper1 (142,296

Table 3.3

Single iteration errors and execution times (seconds) on a 32 *processor SGI Origin* 2000 *for the improved and original methods. Accuracy is compared with a reference using* 9 *degree multipole expansion (the exact computation takes over* 900 *seconds). For the improved method, the degree (indicated by $\star$) refers to the minimum degree.*

Propeller
140,800 elements, 70,439 nodes, 6 Gauss points per element

Algorithm	Degree	Time	ϵ
Original	4	19.73	0.000410
Improved	$4^\star$	22.91	0.000027
Reference	9	63.91	—

Gripper2
185,856 elements, 92,918 nodes, 6 Gauss points per element

Algorithm	Degree	Time	ϵ
Original	4	28.98	0.000530
Improved	$4^\star$	32.07	0.000026
Reference	9	95.13	—

elements, 71,152 nodes), and gripper2 (185,856 elements, 92,918 nodes). The first instance is a propeller from an airplane and the next two are surface discretizations of an industrial gripper. These correspond to highly unstructured problem instances since a bulk of the volume is empty and the nodes are concentrated on the surface.

In Table 3.3, we present single iteration errors and execution times for the improved and original methods. The errors are computed with respect to a 9 degree polynomial since the exact method took an inordinately large amount of time. From the table, once again it is evident that the improved method yields significantly better error properties while adding minimal computational overhead. The matrix-vector product was used in a GMRES solver with a restart of 10 and was observed to converge very well. This is consistent with the diagonal dominance of the kernel ($1/r$ generating function). Using this method, we were able to solve dense systems with over 100,000 unknowns within a few minutes.

4. Parallel treecodes.

The improved and original Barnes–Hut methods have been implemented for the SGI Origin 2000 and tested with up to 32 processors. The code is based on POSIX threads and optimized for single-processor cache performance, data-locality across processors, and false sharing. The parallel formulation exploits concurrency available in independent tree traversal of each particle. The particles are sorted in a proximity-preserving order (a Peano–Hilbert ordering) and potential computations for groups of consecutive particles are aggregated into a single thread. The size of each group, m, is a user specified parameter. In our experiments, we use 200–400 threads with 100–200 particles per thread. This is adequate for balancing the load across 32 processors. In addition, the hierarchical tree is stored in a spatial order to optimize cache performance. The reader is referred to [7, 8, 16, 17] for details of these parallel schemes.

In Table 4.1, we present speedup achieved by the improved and original methods on a 32 processor SGI Origin 2000. The speedup is computed as the ratio of the runtime of the threaded version with multiple kernel threads to that of the single

Table 4.1

Execution time (in seconds) and speedups (in parentheses) for single-thread and multithreaded versions of a single iteration of the treecode on a 32 *processor SGI Origin* 2000 *($\alpha = 0.707$, $p = 6$).*

		Serial		Parallel	
n	Distribution	Original	New	Original	New
40,000	uniform	139.73	145.61	4.95 (28.23)	5.32 (27.37)
46,000	nonuniform	257.35	290.51	8.82 (29.18)	10.95 (26.67)

thread version. Since it is difficult to mask processors from the thread scheduler, the results are only available for 32 processors. It is evident from the table that the performance of the treecode is extremely good, with parallel efficiencies in the range of 80–90%. This must be tempered by the observation that the data-set for the two simulations presented is approximately 140 MB. A single processor of the Origin has an L2 cache of 4 MB; and across 32 processors, a total cache of 128 MB. This is indeed very close to the data-set size. Consequently, at this level, the program works almost entirely out of the L2 cache and this contributes to the excellent performance. Nevertheless, the treecode yields excellent speedups on the Origin 2000.

The improved algorithm yields slightly poorer speedups than the original algorithm. This is because the improved algorithm fetches longer multipole series. However, the effect of this increased communication is not very significant because a large fraction of the data is local to the processor. The increased communication volume can also be estimated in a manner similar to the computation and shown to be bounded.

5. Conclusions and ongoing work. Hierarchical treecodes have proven to be a critical component of large scale n-body computations. In this paper, we have presented an improved treecode that yields considerably better error bounds while incurring minimal additional computational overhead. We prove these bounds theoretically for uniform distributions and demonstrate them experimentally for uniform as well as nonuniform distributions. Parallel formulations of these techniques are shown to yield excellent speedups on a 32 processor SGI Origin 2000. The treecode is also used to solve large scale boundary element problems, and the performance of the matrix-vector product based on the improved method is shown to be superior to the original method. The results presented in this paper can easily be extended to the the fast multipole method as well. We are currently in the process of extending our technique to unstructured distributions, and applying it to fast multipole method.

Acknowledgments. The authors wish to thank the reviewers whose comments helped in improving the quality of this manuscript.

REFERENCES

[1] S. Aluru, *Greengard's n-body algorithm is not $O(N)$*, SIAM J. Sci. Comput., 17 (1996), pp. 773–776.

[2] R. J. Anderson, *Computer science problems in astrophyiscal simulation*, in Silver Jubilee Workshop on Computing and Intelligent Systems, Indian Institute of Science, Tata-McGraw-Hill Publishing Company Limited, New Delhi, 1993.

[3] J. A. Board, J. W. Causey, J. F. Leathrum, A. Windemuth, and K. Schulten, *Accelerated molecular dynamics with the fast multipole algorithm*, Chem. Phys. Lett., 198 (1992), pp. 82–94.

[4] P. B. Callahan and S. R. Kosaraju, *A decomposition of multi-dimensional point-sets with applications to k-nearest-neighbors and n-body potential fields*, J. ACM, 42 (1995), pp. 67–90.

[5] R. Coifman, V. Rokhlin, and S. Wandzura, *The fast multipole algorithm for the wave equation: A pedestrian prescription*, IEEE Antennas and Propagation Magazine, 35 (1993), pp. 7–12.

[6] N. Engheta, W. Murphy, V. Rokhlin, and M. Vassiliou, *The fast multipole method for electromagnetic scattering problems*, IEEE Trans. Antennas and Propagation, 40 (1992), pp. 634–641.

[7] A. Grama, V. Kumar, and A. Sameh, *Parallel hierarchical solvers and preconditioners for boundary element methods*, SIAM J. Sci. Comput., 20 (1998), pp. 337–358.

[8] A. Grama, V. Kumar, and A. Sameh, *Scalable parallel formulations of the Barnes-Hut method for n-body simulations*, Parallel Comput. J., 24 (1998), pp. 797–822.

[9] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, MIT Press, Cambridge, MA, 1987.

[10] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.

[11] N. Katz, T. Quinn, E. Bertschinger, and J. M. Gelb, *Formation of quasars at high redshift*, Mon. Not. R. Astron. Soc., 270 (1994), pp. L71–L74.

[12] N. Katz and S. White, *Hierarchical galaxy formation: Overmerging and the formation of an x-ray cluster*, Astrophys. J., 412 (1993), pp. 412–455.

[13] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White, *Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory*, SIAM J. Sci. Comput., 15 (1994), pp. 713–735.

[14] V. Rokhlin, *Rapid solution of integral equations of classical potential theory*, J. Comput. Phys., 60 (1985), pp. 187–207.

[15] J. Salmon and M. S. Warren, *Parallel out-of-core methods for n-body simulation*, in 8th SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, MN, 1997.

[16] J. Singh, C. Holt, T. Totsuka, A. Gupta, and J. Hennessy, *Load balancing and data locality in hierarchical n-body methods*, J. Parallel Distributed Comput., 27 (1995), pp. 118–141.

[17] M. Warren and J. Salmon, *A parallel hashed oct tree n-body algorithm*, in Proceedings of Supercomputing Conference, Portland, OR, November 15–19, 1993, pp. 12–21.