

Improving Error Bounds for Multipole-Based Treecodes *

Ananth Grama, Vivek Sarin, and Ahmed Sameh
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907
{ayg, sarin, sameh}@cs.purdue.edu
Phone: (765) 494 6964, FAX: (765) 494 0739

Abstract

Rapid evaluation of potentials in particle systems is an important and time-consuming step in many physical simulations. Over the past decade, the development of treecodes such as the Fast Multipole Method (FMM) and the Barnes-Hut method has enabled large scale simulations in domains such as astrophysics, molecular dynamics, and material science. FMM and related methods rely on fixed degree polynomial (p) approximations of the potential of a set of points in a hierarchy. In this paper, we present a sequence of results to illustrate that keeping the multipole degree constant can lead to large aggregate errors. An alternate strategy based on a careful selection of the multipole degree leads to asymptotically lower errors; while incurring minimal computation overhead for practical problem sizes. The paper presents theoretical results for computing the degree of a particle-cluster interaction, the error associated with the interaction, the error associated with a particle for all of its interactions, and the computational complexity of the new method. These results show that it is possible to reduce the simulation error asymptotically while incurring minimal computational overhead. The paper also presents experimental validation of these results on a 32 processor Origin 2000 in the context of problems ranging from astrophysics to boundary element solvers. In addition to verifying theoretical results, we also show that it is possible to achieve excellent parallel speedup for the treecode.

1. Introduction

The problem of evaluating the potential due to a set of particles is an important and time-consuming one. The de-

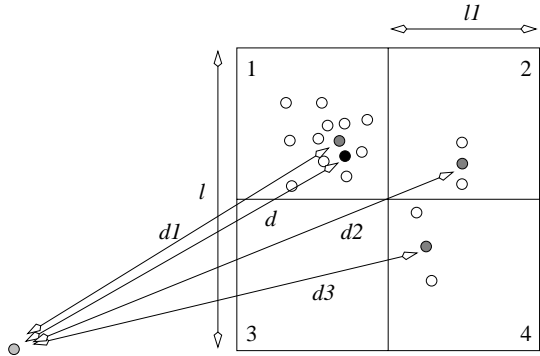
velopment of fast treecodes such as the Barnes-Hut and Fast Multipole Methods for n -body systems has enabled large scale simulations in astrophysics [11, 12, 15] and molecular dynamics [3]. Coupled with efficient parallel processing, these treecodes are capable of yielding several orders of magnitude improvement in performance [8, 17, 16]. In addition, treecodes also have applications in the solution of dense linear systems arising from boundary element methods [5, 6, 14, 7, 13].

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. At each step in this phase, if the domain contains more than s particles (for some preset constant s), it is recursively divided into four equal parts (eight parts in three dimensions). This process continues until each part has s elements or fewer. The resulting tree is an unstructured quad-tree (oct-tree in three dimensions). This tree is now traversed in post-order. Each internal node in the tree computes and stores an approximate representation of the particles contained in that sub-tree. This representation can either be a monopole or a multipole series. For astrophysical simulations, this can often be 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: the multipole acceptance criterion is applied to the root of the tree to determine if an interaction can be computed; if not, the node is ex-

*This research has been supported by NSF Grants #EIA9876014 and # ESC-95-27123. Computational resources for the project were provided by the National Computational Science Alliance and the Minnesota Supercomputing Institute.

panded 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 some constant α , an interaction can be computed. The Barnes-Hut method is illustrated in Figure 1.



- Center of mass of domain
- Centers of mass of subdomains
- Source particle

```

if ( $l/d < a$ )
    compute direct force interaction
    with the center of mass of domain.
else
    if ( $ll/d1 < a$ )
        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

```

Figure 1. Illustration of the serial Barnes-Hut method.

For a balanced tree, each of the n particles needs $O(\log n)$ interactions. Using a p -term multipole expansion, each interaction takes $O(p^2)$ time. This results in a total computational complexity of $O(p^2 n \log n)$. However, the tree size 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 [4, 1]. However, using box-collapsing techniques (the box is first collapsed to the smallest box that contains all the particles in the subdomain), this complexity can be reduced. There are some recent results demonstrating that it is beneficial to work with binary trees as opposed to 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, FMM computes potentials instead of forces. These potentials may be electrostatic, gravitational or others depending on the application. It is easy to see that force is equal to the gradient of potential, and therefore can be easily computed from the latter. Furthermore, since potential is a scalar quantity, it simplifies many computations. 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. FMM therefore uses cluster-cluster interactions in addition to particle-cluster interactions. The computational complexity of FMM was originally shown to be $O(n)$ for uniform distributions. Callahan and Kosaraju [4] show that the complexity of the potential estimation phase can be reduced to $O(n)$ for arbitrary distributions with preprocessing for computing well-separated sets.

The dominant cost in both of these treecodes is the potential (or force) estimation phase. This involves computing interactions between p term multipoles; which takes time $O(p^2)$. If the complexity of the potential estimation phase in a treecode is $O(R)$, then using p term multipoles results in a complexity of $O(p^2 R)$ in three dimensions. For uniform distributions, $R = n \log n$ for Barnes Hut and $R = n$ for FMM.

In this paper, we address the errors associated with these methods. We present a sequence of theoretical results which show that the error in these methods can be reduced significantly by selecting the multipole degree appropriately for different clusters. Furthermore, we show that for practical problem sizes, this adds minimal computational complexity. We illustrate these theoretical results in the context of both particle simulations as well as boundary element methods. These experiments are presented for a 32 processor SGI Origin 2000. The POSIX threads based treecode is also shown to yield excellent speedups even for relatively small problems.

2. Global Error Estimate for 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 1 *Suppose*

that k charges of strengths $\{q_j, j = 1, \dots, k\}$ 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 $P = (r, \theta, \psi) \in \mathbf{R}^3$ 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

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

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

Proof See page 54 of [9]. \square

We now present a sequence of theorems that bound the ratio r_s/r on both sides. This bound is used to quantify the error associated with a single interaction in the Barnes-Hut method. We bound the number and type of interactions for each particle and show that by holding the error associated with each interaction constant, we can reduce the overall error significantly. We provide theorems for determining the multipole degree to keep interaction error constant. Finally, we show that the computational complexity of varying the multipole degree in the prescribed manner adds little additional overhead for practical problem sizes.

Theorem 2 *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.

Proof Consider the scenario illustrated in Figure 2. Since a particle s interacts with a box b , but not with its parent box B ,

$$r \geq r_0$$

and

$$R < R_0$$

Here, $r_0 = r_s/\alpha$ and $R_0 = 2r_s/\alpha$. Using the triangle inequality $R + r_s/\sqrt{2} \geq r$, we can show that

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

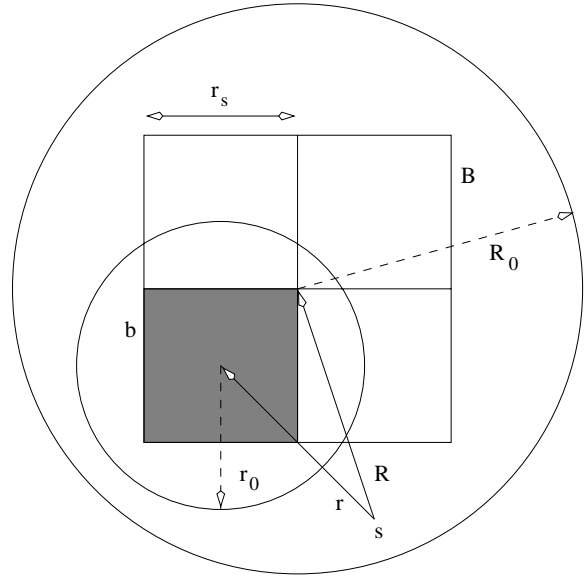


Figure 2. Establishing bounds on $\frac{r_s}{r}$ in a Barnes-Hut algorithm. An interaction with box b indicates that particle s could not interact with its parent box B . This is used to establish the lower bounds.

The bounds on r_s/r are tight. For smaller values of α , this bound tends to $\alpha/2 \leq r_s/r \leq \alpha$. For example, for $\alpha = 0.6$, the bounds are $0.25 \leq r_s/r \leq 0.6$. \square

Theorem 3 *In Barnes-Hut method, a particle interacts with a bounded number of boxes of a given size.*

Proof Since $\alpha' \leq r_s/r \leq \alpha$, the centers of all boxes of size r_s lie within an annular region defined by the following relation:

$$\frac{r_s}{\alpha} \leq r \leq \frac{r_s}{\alpha'}$$

and the boxes lie completely within the annular region defined by:

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

The ratio of the volume of this annular region and the volume of a single box gives the following upper bound in three dimensions on the number of boxes of size r_s :

$$n_{max} \leq \frac{4\pi}{3} \left[\left(\frac{1}{\alpha'} + \frac{1}{\sqrt{2}} \right)^3 - \left(\frac{1}{\alpha} - \frac{1}{\sqrt{2}} \right)^3 \right]$$

□

Theorem 4 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 for well-separatedness, the error in potential outside the sphere at a distance r from the center of the sphere due to these charges is bounded by

$$\epsilon \leq \frac{A}{r - r_s} \left(\frac{r_s}{r} \right)^{p+1} \leq \frac{A}{r_s} \times \frac{\alpha^{p+2}}{1 - \alpha}$$

where p is the degree of the truncated multipole expansion with $p > 1$ and

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

Proof The first part of this theorem

$$\epsilon \leq \frac{A}{r - r_s} \left(\frac{r_s}{r} \right)^{p+1} \quad (2)$$

is proved on page 54 of [9]. For proving the second part of this theorem, using the α -criterion of the Barnes-Hut method, we know that $r_s/r < \alpha$. Therefore,

$$\begin{aligned} r_s &\leq \alpha r \\ r - r_s &\geq r(1 - \alpha) \\ \frac{1}{r - r_s} &\leq \frac{1}{r} \times \frac{1}{1 - \alpha} \end{aligned}$$

We know that $1/r \leq \alpha/r_s$. Making this substitution, we have

$$\frac{1}{r - r_s} \leq \frac{\alpha}{r_s} \times \frac{1}{1 - \alpha}$$

Substituting this into Equation 2, we have

$$\epsilon \leq \frac{A}{r_s} \frac{\alpha^{p+2}}{1 - \alpha}$$

□

This theorem illustrates the problem as also the solution proposed in this paper. The associated error grows linearly with the net charge of the particle clusters. Moreover, the size of the largest cluster with which an interaction is computed in the Barnes-Hut method can be shown to be a constant factor of the total simulation domain. Therefore, the

associated errors can be large, and even unbounded for unstructured distributions. However, in applications such as protein simulations, the charge density is largely uniform across the domain of simulation. In such applications, the overall error in the Barnes-Hut method grows linearly with the magnitude of charge in the system. For larger systems, this error may become unacceptable.

Theorem 4 also provides an easy way for controlling error in particle-cluster interactions. By increasing the polynomial degree p with increasing A , the error in each interaction can be bounded by a constant. Since it has been shown that a single particle interacts with a fixed number of clusters of a given size, the overall error accumulated in all interactions of a particle can be logarithmically bounded. The next three theorems determine the variation of polynomial degree for constant error, the computational complexity of this new method, and its error bounds.

Theorem 5 The polynomial degree p_k required for a particle-cluster interaction for constant error is given by

$$p_k = p_j + \frac{k - j}{\log \alpha} + \frac{\log A_j - \log A_k}{\log \alpha}$$

Here, A_k is the current cluster at level k , and A_j is the smallest cluster at level j of the hierarchy for which the desired error criterion is satisfied.

Proof From Theorem 4, we know that

$$\epsilon \leq \frac{A}{r_s} \frac{\alpha^{p+2}}{1 - \alpha}$$

We assume the error associated with a single interaction to be a constant ϵ . Let d_j be the smallest cluster for which this error criterion is satisfied. In the original multipole method, for all other domains with higher aggregate charge, this error criteria will be violated. To compensate for this, assume another cluster d_k at a higher level in the tree. Since the error associated with these two domains is equal, we have

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

or

$$\frac{A_j}{A_k} \frac{r_{sk}}{r_{sj}} = \alpha^{p_k - p_j} \quad (3)$$

Here,

$$\frac{r_{sk}}{r_{sj}} = 2^{k-j}$$

Substituting in Equation 3 and simplifying, we have

$$p_k = p_j + \frac{k - j}{\log \alpha} + \frac{\log A_j - \log A_k}{\log \alpha}$$

□

In general, we select a minimum degree of interaction associated with the threshold value of A_j and increase multipole degree for larger cluster sizes. For structured domains, it is easy to control the polynomial degree in this manner since the increase in polynomial degree is not large. The multipole series are computed a-priori to the maximum required degree (this is possible since all parameters for the degree of an interaction are available at the time of tree-construction). This technique does not work very well for unstructured domains because it might result in very large degree multipoles. This difficulty can be overcome in two ways: (i) by altering the α criterion of the Barnes-Hut method; or (ii) by computing and storing the increased degree multipoles. In this paper, we will concentrate largely on uniform distributions; but will demonstrate empirically 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 6 *The error in the piecewise approximate Barnes-Hut method for structured distributions is $O(\alpha^{p+1} \log A)$.*

Proof This proof follows naturally from Theorems 2, 3, 4, 5; namely, the number of interactions with subdomains at any level are constant; the number of subdomain interactions is logarithmic in the number of particles; and the error associated with a single particle-subdomain interaction is constant. \square

The reader will note that this error is considerably less than the error bound on the original fixed-degree multipole based Barnes-Hut method. The only issue that remains to be resolved is the increased computation introduced by the additional multipole evaluations. The next theorem shows that this additional computation is minimal.

Theorem 7 *For a structured particle distribution, the computational complexity of the piecewise approximate Barnes-Hut method is given by $O(n(p+l)^3)$. Here, l is the number of levels of the hierarchical decomposition.*

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 between 256K - 2M particles. In order to optimize cache performance and for 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 8M and 64M. 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 7/3 of the original method. Clearly, the new method yields significant improvements in error while incurring minimum additional overhead.

3. Experimental Results

3.1. Experimental Setup

The improved and original Barnes-Hut methods are coded for an 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 the concurrency available in independent tree traversal of each particle. The particles are sorted in a proximity-preserving order (a Peano-Hilbert ordering) and force computation for sets of w particles are aggregated into a single thread. Here, w is a user specified parameter. Typically, in our experiments, we used 100 particles per thread. For a 40K particle simulation, this corresponds to 400 threads. This is adequate for balancing load across 32 processors. The hierarchical tree is also stored in a spatial order to optimize cache performance. We refer the reader to [7, 8, 17, 16] for a more detailed explanation of these schemes.

3.2. Application and Problem Instances

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 equally across the domain. Irregular distributions are generated using a Gaussian density function or overlapped Gaussian distributions (multiple Gaussians superimposed). Figure 3 illustrates examples of the three distributions. The number of particles in these distributions range from 20K to 100K. The simulation instances are named $ip\{n\}$ for irregular distributions with n particles and $uniform\{n\}$ for uniform distributions with n particles. The sizes of problem instances are selected to enable running various experiments on them. Errors in larger instances become more difficult to estimate since the accurate algorithm scales as $O(n^2)$. The notion of error in a simulation is formally defined as follows: let a be the vector corresponding to the accurate potentials at n particles; if the potentials computed from the treecode are represented by the vector a' then the error ϵ in the simulation is defined as:

$$\epsilon = \max |a_i - a'_i| \quad i = 1 \dots n$$

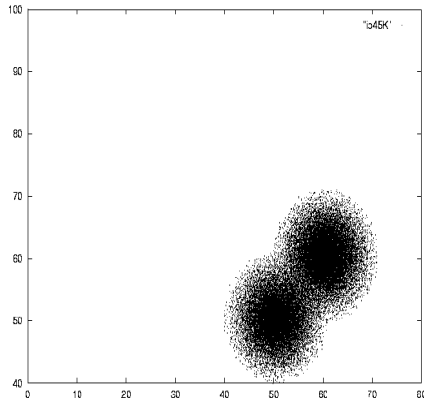
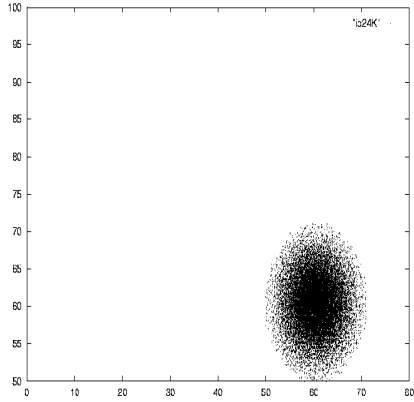


Figure 3. Sample distributions for experiments: (a) Gaussian (24K particles), and (b) Overlapped Gaussians (45K particles).

The treecode was also used to solve dense linear systems arising from boundary element methods for solving integral equations. The core of these iterative solvers is the application of the dense coefficient matrix to an intermediate vector. The coefficient matrix is generated by the Green's function of the Laplace's equation. The Green's function is $-\log r$ in two dimensions and $1/r$ in three dimensions. Problems of this nature 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 triangles. The integration over the surface is performed using Gaussian quadrature. A fixed number of Gauss-points are located inside the triangle and inserted into the hierarchical domain representation. Using this hierarchical domain, the potential is computed at the vertices of the triangles and matched to the boundary values. This process forms a single matrix-vector product that is required at

each step of the GMRES iterative solver.

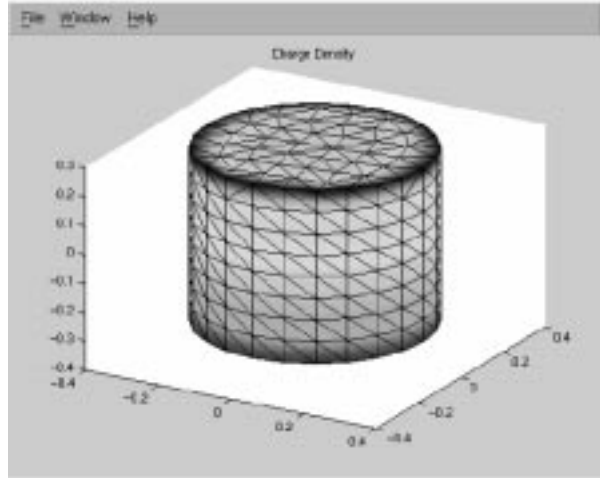


Figure 4. Charge density on the surface of a cylinder with one end fixed at a potential of 0V and the other end at 1V.

We use this technique to solve charge distribution problems on complex 3-D geometries. In Figure 4, we illustrate a simple problem instance with 572 triangles 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 (140800 triangles, 70439 nodes), gripper1 (142296 triangles, 71152 nodes), and gripper2 (185856 triangles, 92918 nodes). The first instance is a propeller from an airplane and the next two are surface discretizations of an industrial gripper. It is easy to see that these correspond to highly unstructured problem instances, since a bulk of the volume is empty and the nodes are concentrated on the surface.

3.3. Performance and Accuracy of the Improved Method

In this section, we examine the runtime and error properties of the two methods. The computation is expressed in terms of the number of multipole term expansions. This grows as p^2 for a p -term multipole expansion. The number of terms expanded is a very good indication of the serial complexity of the method. The reason we use this (instead of wall clock time) is because we are comparing serial complexities and errors; and as such would like to avoid efficiency factors introduced by parallelism; or time discrepancies introduced by other processes on the machine.

Table 1 illustrates the errors and the number of term expansions. From these tables, it is easy to see that the growth in error is much faster in the original method than in the improved method. Furthermore, the term expansions of the

n	Error(orig)	Error(new)	Terms(orig)	Terms(new)
Structured distributions				
10000	0.012027	0.012027	12293460	12293460
15000	0.017326	0.010399	24665616	25274289
20000	0.025982	0.016820	59631516	60443782
30000	0.036880	0.017386	94953456	96171621
80000	0.098395	0.019327	253741860	297318482
Unstructured distributions				
45000	2.479027	0.334691	70094952	102045637
82000	2.307508	0.268452	179039880	224057225

Table 1. Error and computation associated with the original and new methods. The table shows that error in the improved method grows much slower than the original method with minimal increase in computation.

two methods are not very different. This is also illustrated by the graphical representation in Figure 5 and is in good agreement with our theoretical results.

3.4. Parallel Performance of the Treecodes

In this section, we present the speedup achieved by the treecode for the original as well as the new methods. These experiments are presented in Table 2 for 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 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 dataset for the two simulations presented is roughly 140 MB. A single processor of the Origin has a 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 L2 cache and this contributes to the excellent performance. Nevertheless, the treecode yields excellent speedups on the Origin 2000.

The new algorithm yields slightly poorer speedups than the original algorithm. This is because the new 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.

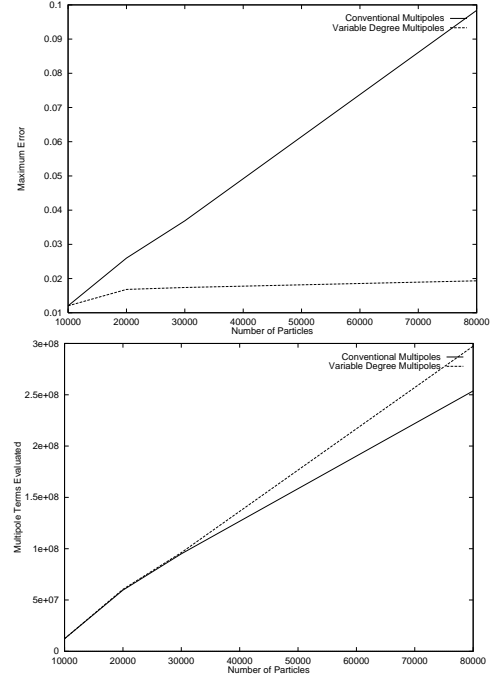


Figure 5. A comparison of the error and computational cost of the original and new methods illustrates the close agreement with theoretical results and advantages of the new scheme.

3.5. Performance of the Matrix-Vector Product

In Table 3, we present single iteration accuracy and runtimes for the new 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 new 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 di-

Problem	Serial		Parallel	
	Original	New	Original	New
uni40K	195.46	212.41	6.68 (29.26)	7.37 (28.82)
non-u46K	360.93	390.68	11.67 (30.92)	12.97 (27.83)

Table 2. Runtimes (in seconds) and speedups (in parenthesis) for single-thread and multi-threaded versions of a single iteration of the treecode on a 32 processor SGI Origin 2000.

Problem: Propeller; Num. triang: 140800 Num. nodes: 70439; Num. Gauss points: 6			
Algorithm	Degree	Time	Error
Original	4	31.83	0.000406
Improved	4*	33.60	0.000026
Reference	9	100.81	-
Exact		> 900.0	-

Problem: Gripper2; Num. triang: 185856 Num. nodes: 92918; Num. Gauss points: 6			
Algorithm	Degree	Time	Error
Original	4	46.40	0.000516
Improved	4*	50.13	0.000028
Reference	9	151.21	-
Exact		> 900.0	-

Table 3. Single iteration accuracy and run-times (seconds) on a 32 processor SGI Origin 2000 for the new and original methods. The new method yields significantly better errors while incurring minimal computational overhead.

agonal dominance of the kernel ($1/r$ generating function). Using this method, we were able to solve dense systems with over 100K unknowns in a few minutes.

4. Ongoing Work and Conclusions

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 computational overhead. We prove these bounds theoretically and demonstrate them experimentally for uniform as well as non-uniform distribution. Parallel formulations of these techniques are shown to yield excellent speedups on a 32 processor SGI Origin 2000. The treecode is also applied to solving large scale boundary element problems. The performance of the new matrix-vector product 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 exploring this and extending our theoretical results to unstructured distributions.

References

[1] S. Aluru. Greengard's n-body algorithm is not order n. *SIAM Journal on Scientific Computing*, 17(3):773–776, 1996.
[2] R. J. Anderson. Computer science problems in astrophysical 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. Let.*, 198:89, 1992.
[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. In *Proceedings of 24th Annual ACM Symp. on Theory of Computing*, pages 546–556, May 1992.
[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(3), June, 1993.
[6] N. Engheta, W. Murphy, V. Rokhlin, and M. Vassiliou. The fast multipole method for electromagnetic scattering problems. *IEEE Transactions on Antennas and Propagation*, 40(6), June, 1992.
[7] A. Grama, V. Kumar, and A. Sameh. Parallel hierarchical solvers and preconditioners for boundary element methods. Technical report, Purdue University, W. Lafayette, IN 47907, 1997. Accepted for publication in *SIAM Journal of Scientific Computing*.
[8] A. Grama, V. Kumar, and A. Sameh. Scalable parallel formulations of the Barnes-hut method for n-body simulations. Technical report, Purdue University, W. Lafayette, IN 47907, 1997. Accepted for publication in the *Parallel Computing Journal*.
[9] L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems*. MIT Press, 1987.
[10] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comp. Physics*, 73:325–348, 1987.
[11] N. Katz, T. Quinn, E. Bertschinger, and J. M. Gelb. Formation of quasars at high redshift. *Mon. Not. R. Astron. Soc.*, 270:L71–L74, 1994.
[12] N. Katz and S. White. Hierarchical galaxy formation: over-merging and the formation of an x-ray cluster. *Astrophysical Journal*, 412(2):412 – 455, 1993.
[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 Journal on Scientific Computing*, 15(3):713 – 735, May 1994.
[14] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *Journal of Computational Physics*, 60:187 – 207, Sept 1985.
[15] J. Salmon and M. S. Warren. Parallel out-of-core methods for n-body simulation. In *8th SIAM Conf. on Parallel Processing for Scientific Computing*, 1997.
[16] J. Singh, C. Holt, T. Totsuka, A. Gupta, and J. Hennessy. Load balancing and data locality in hierarchical n-body methods. *Journal of Parallel and Distributed Computing*, 1994.
[17] M. Warren and J. Salmon. A parallel hashed oct tree n-body algorithm. In *Proceedings of Supercomputing Conference*, 1993.