1987ApJS...64..715H

THE ASTROPHYSICAL JOURNAL SUPPLEMENT SERIES, 64:715–734, 1987 August
© 1987. The American Astronomical Society. All rights reserved. Printed in U.S.A.

PERFORMANCE CHARACTERISTICS OF TREE CODES

LARS HERNQUIST
Department of Astronomy, University of California, Berkeley; and Institute of Geophysics and Planetary Physics, Lawrence Livermore National Laboratory
Received 1986 November 19; accepted 1987 February 10

ABSTRACT

A FORTRAN implementation of the Barnes-Hut hierarchical tree algorithm is presented and analyzed in the context of the astrophysical N-body problem. In particular, timing tests performed on a CRAY X-MP demonstrate that the current version is a factor $\sim$150–200 times faster than the Barnes-Hut C code running on a VAX 11-780 with floating-point accelerator. It is verified that this technique scales as $O(N \log N)$ for useful values of the clumping parameter θ ($0.3 \lesssim \theta \lesssim 1.0$). Given this behavior, a comparison with a traditional $O(N^2)$ direct force computation indicates that the tree method is more efficient for N greater than about a few thousand. The typical error introduced into the force calculation by the clustering of particles at large distances is shown to be small ($\lesssim 1\%$ relative to a direct sum over particles) for $\theta \lesssim 1$. The error decays weakly with increasing N and can be minimized if the softening length, ε, is chosen to be $\varepsilon \lesssim \lambda$, where λ is the mean interparticle separation. If quadrupole terms are included in the multipole expansion of the cluster potentials, then the error can be reduced further, and the efficiency of the code can be improved in some situations. It is shown that the errors in the force computation contribute to the violation of the conservation laws, and that this effect places further constraints on θ. Finally, the influence of the approximations on the relaxation time is considered. Simple tests indicate that the tree method will be no more collisional than a direct $O(N^2)$ calculation. In view of this conclusion, and the lack of restrictions on either the spatial resolution or global geometry, tree algorithms should be suitable for studying a wide range of astrophysical phenomena, including potentially both collisional and collisionless systems. As an example, a simulation of the decay of a satellite orbit around a self-gravitating disk with $N = 32{,}768$, performed with the tree method, is compared with previous calculations.

Subject headings: numerical methods

I. INTRODUCTION

Discrete N-body techniques have been used to investigate a wide range of astrophysical phenomena during the past 25 years. The discrepancy between the number of objects present in a large system such as a galaxy ($N \sim 10^{11}$), and the number of particles that can be treated in a computer simulation ($N \lesssim 10^5$) has motivated vigorous searches for increasingly efficient algorithms. The earliest studies (e.g., von Hoerner 1960; Aarseth 1963; Hénon 1964) relied on the straightforward particle-particle (PP) method, in which the force on a particle is computed as a direct sum over all the individual two-body interactions. Since no approximations are made in the force computation, these codes have the advantage of offering a large, controllable dynamic range in spatial resolution and are not limited to specific geometrical configurations. Unfortunately, the CPU time required per time step scales as $\sim O(N^2)$ in the most naive implementations. Refinements, such as the introduction of a higher order integrator (e.g., Aarseth 1971) and multiple time scales (e.g., Ahmad and Cohen 1973) have reduced the order of this method somewhat (for a review see Aarseth 1985). Nevertheless, the computational expense has remained prohibitive for $N \gtrsim 10{,}000$ even on the most powerful supercomputers.

With the development of the fast Fourier transform (FFT; Cooley and Tukey 1965), efficient Eulerian algorithms became feasible (e.g., Hockney 1967; Hohl and Hockney 1969). In the particle-mesh (PM) method mass density is assigned to a fixed grid and forces are obtained through the solution of Poisson's equation via Fourier transforms and the convolution theorem (for a review see Hockney and Eastwood 1981; Birdsall and Langdon 1985). The use of the FFT reduces the operations count to $\sim O(M \log M)$, where M, the number of grid cells, is typically approximately equal to the number of particles. As a result it has been possible to perform simulations with a large number of particles ($N \sim 10^5$). However, the spatial resolution is constrained by the mesh spacing and is further degraded by numerical errors (Bouchet, Adam, and Pellat 1985; Bouchet and Kandrup 1985). In addition, the introduction of a fixed grid limits the flexibility of this method because of geometrical restrictions and difficulties in handling individual nearby binary encounters. Recent refinements to the PM method have included the use of multiple meshes to provide local enhancements in the spatial resolution. Although such a procedure may be well suited for interacting galaxies (e.g., James and Weeks 1986), it is unclear whether or not it could be adapted to systems with many high-density peaks whose locations are not known *a priori*.

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

Some of the advantages of both PP and PM techniques can be retained in hybrid schemes such as the particle-particle–particle-mesh (P^3M) method (e.g., Eastwood and Hockney 1974). The contribution to the force from nearby particles is determined by a direct sum, while a PM computation is used for the distant interactions. As a result P^3M codes are significantly faster than those based on the PP technique alone and have a much larger dynamic range in spatial resolution than PM codes. This method has proved to be most advantageous for inhomogeneous systems with a high density contrast, such as the large-scale structure formed early in the universe (Efstathiou *et al.* 1985; Davis *et al.* 1985). However, P^3M codes tend to be slow in comparison with PM routines, since the primary bottleneck is in the short-range force computation (the method scales as $\sim O(N_n N)$, where N_n is the average number of nearby particles). Finally, the requirements imposed by a fixed grid limit the flexibility of this method.

An independent class of algorithms, loosely referred to as the expansion method, was first proposed in the astrophysical setting by Aarseth (1967). The potential is obtained through the use of multipole expansions to solve Poisson's equation (e.g., van Albada and van Gorkum 1977; van Albada 1982; Villumsen 1982; for a review see van Albada 1986). If the expansions are truncated at low order, then this technique is highly efficient, the force computation per step scaling as $O(nN)$, where n is the number of terms in the expansion. In addition, owing to the mean field nature of the solution to Poisson's equation, expansion codes are less collisional than those based on other techniques. However, this procedure is somewhat limited in its range of applicability. A low-order expansion will not be an accurate representation of the global potential unless the system and basis functions chosen for the expansion share the same symmetry properties. Thus a given code will be useful only for certain geometries. The effective resolution of this technique is limited by the truncated expansion and is, furthermore, highly anisotropic (e.g., McGlynn 1984). It may not be trivial, therefore, to perform a rigorous error analysis for this approach. Individual two-body effects are completely suppressed and, hence, collisional systems cannot be modeled. Finally, it is awkward to generalize codes of this type to handle multiple expansion centers (e.g., Piran and Villumsen 1986).

Recently a new class of hybrid N-body algorithms has been devised which physically incorporates the advantages of the PP and mean field expansion methods (Appel 1981,1985; Jernigan 1985; Porter 1985; Barnes and Hut 1986; Barnes 1986; Greengard and Rokhlin 1986; Press 1986). The short-range force on a particle is, on average, computed as a direct sum over nearby particles. Remote bodies are organized into progressively larger groups with distance from the particle, and then a multipole expansion of the potential of each cluster about its center of mass is performed. The long-range contribution to the acceleration is then given by the sum of the particle-cluster interactions. To date, all implementations of this scheme have used a tree data structure to perform the clustering operation efficiently, and, as a result, have become known as tree codes. It should be emphasized, however, that the tree has been introduced merely as a computational device and may not ultimately represent the best data structure for this purpose. (For example, it may be advantageous to link internal nodes together to allow for more efficient communication between clusters. The resulting configuration should properly be referred to as a graph.)

Thus far, two distinct variations of this technique have been proposed, which differ primarily in their organization of the particle information. The philosophy of the approach adopted by Appel, Jernigan, and Porter (AJP) (Appel 1981, 1985; Jernigan 1985; Porter 1985) is to apply a recursive center of mass coordinate transformation to the particle positions, leading to an arrangement of the particles in a hierarchy of clusters. The data structure used to express this transformation is a binary tree, with particles at the leaves and internal nodes representing clusters of increasing complexity toward the root. Each node is labeled with the mass and center-of-mass coordinates of the cluster (i.e., all the particles below the node) which it represents. The optimal tree can be constructed unambiguously as follows (Jernigan 1985; Porter 1985): Given a list of N particles, the two which are closest together in physical space are chosen and joined together to form a node. The two particles are removed from the list and replaced by the center-of-mass node, leaving a list of $N-1$ entities. The above procedure is repeated, which may result in the joining of either two other particles or a single particle with the center-of-mass node. This process is repeated indefinitely until only one object remains, identified as the root node, which has a mass equal to the sum of the masses of all the particles and is located at the center of mass of the system. The tree will have $\sim O(\log_2 N)$ levels and can be updated in a time $\sim O(N \log N)$ (Porter 1985). Since the transformation preserves the spatial relationship of particles and clusters (i.e., entities which are physically close together will be simply situated with respect to one another in the tree), the data structure generated in this way should be referred to as a mutually nearest neighbor tree (Appel 1985; Press 1986).

The technique introduced by Barnes and Hut (1986, hereafter BH) relies on a hierarchical subdivision of space into cubic cells and is, in a sense, an Eulerian analog of the AJP algorithm. An oct tree (8 descendants per node, in three dimensions) is used to organize the particle data, each node of which represents a physical volume of space. The total mass of all the particles within a given volume and their center-of-mass coordinates are stored at the corresponding node. Thus, the root represents a cubic volume which is sufficiently large to contain all the particles in the system. This cell is subdivided into 8 cubic cells of equal volume which form the basis for the 8 immediate descendants of the root node. Each subvolume is, in turn, subdivided into smaller units, and this procedure is repeated until each cell at the lowest level in the hierarchy (i.e., most finely subdivided) contains either one or zero particles. Information about empty cells is not stored explicitly in the tree; thus the leaves always represent a volume containing precisely one particle. The tree which results from this prescription will have $\sim O(\log_8 N)$ levels and can be constructed in a time $\sim O(N \log N)$.

The significant improvement in efficiency represented by tree codes relative to direct methods [$O(N \log N)$ versus $O(N^2)$] results from approximations made in the force calculation. The interaction between a given particle and clusters

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

of particles is computed to a specified level of accuracy by neglecting the detailed structure of the clusters. This can be performed by comparing the separation d between a given particle and a cluster to the size s of the cluster. If

$$\frac{s}{d} < \theta, \tag{1.1}$$

where θ is a fixed tolerance parameter (BH), then the internal distribution of particles can be neglected and the interaction can be computed using a low-order expansion of the cluster potential about its center of mass. (Note that s/d is the relevant quantity to consider, since the multipole expansion of the cluster potential is essentially a power series in this ratio.) The force on a particle can be obtained for a given θ by walking through the tree and comparing the size of each node to the distance from the node to the particle. If the tolerance criterion (1.1) is satisfied, then the interactions between the particle in question and all particles below the node can be replaced by a single term in the force evaluation. For $\theta > 0$ the result of this tree pruning is typically that the number of terms in the force computation is proportional to $\log N$, rather than to N, for a direct sum over all particles, and hence the CPU time required to advance the state of the system scales as $N \log N$ (BH).

The AJP and BH techniques are similar in that both use trees to describe physical systems, with particles located at the leaves and internal nodes representing compound objects. However, there are several differences that may determine which algorithm will be more efficient in a particular circumstance. The primary advantage of the BH method is that the tree structure is cleaner and simpler than that in the AJP scheme. The sizes and shapes of the nodes are uniform across a given level in the BH tree, but are completely arbitrary in the AJP algorithm. As a result, a rigorous error analysis should be more feasible for the BH method. In addition, tree construction will probably be more efficient than in the AJP method, since the BH tree can be built through the use of bit-oriented arithmetic. The primary advantage of the AJP technique lies in the fact that the spatial relationship between particles and clusters is preserved by the center-of-mass reduction, which is not necessarily true of the BH tree structure. Tightly bound particles could find themselves on widely separated branches of the BH tree if the spatial subdivision introduced a cell boundary between them at a high level in the hierarchy. In addition, the quasi-Eulerian nature of the BH technique implies that the tree must be reconstructed after each step, since particles can cross cell boundaries. It would be possible, though perhaps not desirable, to update the AJP tree only occasionally, after clusters become significantly distorted.

The differences in the BH and AJP approaches will be closely tied to issues relating to the relative efficiency of the two methods, but probably not to the reliability of the results which they produce. (For example, although the cell boundaries introduced by the subdivision of space in the BH method are somewhat artificial, they should not affect the accuracy of the force computation as long as cells which contain nearby particles are opened during the tree walk.) A possible exception might involve situations in which the accurate modeling of the dynamics of short-range binary encounters is critical, such as in a globular cluster. Since binaries would exist naturally as isolated substructures in the AJP tree, it would appear to be far easier to implement techniques such as regularization (Jernigan 1985; Porter 1985) in this method. Thus, it is likely that the two algorithms will be most well suited for investigating different types of astrophysical systems. The AJP technique would appear to be preferred if the global evolution of the system is sensitive to the dynamics of isolated binaries, while the BH method might be more generally useful as a result of its inherent simplicity.

All implementations of the tree method offer a number of significant advantages over previous N-body techniques. In particular, tree algorithms are most closely related to direct N-body codes but are significantly faster because of the approximate nature of the force computation. The lack of a fixed grid or preferred geometry suggests that tree codes will be much more flexible than other efficient N-body techniques. No *a priori* restrictions are imposed on the global geometry of the system. In addition, the dynamic range in spatial resolution is, in principle, unlimited and can be locally fine-tuned through the use of a smoothed two-body interaction potential with variable softening parameter. Tree codes are naturally suited for simulating isolated systems, since there are no complications with aliases as in Fourier methods or ambiguities over particles exiting from a grid, but can be modified to handle periodic boundary conditions (e.g., Appel 1981). Tree codes should be useful, therefore, for studying a wide range of astrophysical problems, including highly inhomogeneous distributions (e.g., merging galaxies, clusters of galaxies, and the formation of large-scale structure in the early universe) and both collisional and collisionless (for large N) systems.

In this paper a FORTRAN implementation of the BH tree method, optimized for a supercomputer, is presented. Specific issues relating to the BH code are summarized in § II. Basic properties of the technique are discussed in § III, such as the scaling of CPU time with N and θ and the error induced in the force computation by the clustering of distant particles. The influence of the approximations on the relaxation time is considered in § IV. Astrophysical examples are given in § V in order to compare the tree method with other particle techniques. Finally, future prospects are outlined in § V.

II. IMPLEMENTATION

a) *Fundamentals*

This paper describes a FORTRAN implementation ("TREECODE") of the BH tree algorithm, a copy of which is available upon request to the author. The essential program consists of approximately 2000 statements, of which roughly 1000 are executable and the remainder comprise the in-line documentation. The code has been written in standard FORTRAN with one important exception: it has been assumed that the compiler allows recursive subroutines. This is not a limitation for CRAYs, since the most recent release of the CRAY FORTRAN compiler, CFT(1.14), includes this feature, as does the Livermore compiler CIVIC. In addition, recursion should be available on any computer which permits multitasked (i.e., parallelized) routines. However, TREECODE will not, in its

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

present form, run on VAX-type machines without some modification.

b) *Simple Refinements*

The basic structure of TREECODE is similar to the BH C code. However, a number of refinements have been introduced in order to optimize the code for a vectorizing machine. (All computations presented in this paper were performed on the MFE CRAY X-MP and CRAY-2 supercomputers at Livermore.) The major bottleneck for this technique is the section of the code that computes the acceleration on a given particle by walking through the tree and comparing the size of each node with the distance from the node to the particle. Rather than accumulating the force on a particle during the tree walk, the relevant subroutine in TREECODE instead records which nodes satisfy the tolerance criterion (1.1). Once the tree search has been completed, the acceleration is computed in a vector loop over all the relevant nodes. Although the tree search is not vectorized in its present form, the improvement that results from vectorization and compiler optimization is still considerable. A number of runs were made using versions of TREECODE compiled with and without vectorization and also with and without compiler optimization (using both CFT and CIVIC). Typically the gain in speed was a factor ~ 1.7–2 from vectorization and a similar factor from compiler optimization, representing an overall improvement by a factor ~ 3–4. Evidently the CRAY compilers are capable of producing highly efficient code.

The CPU requirements were further reduced by hand-optimizing versions of the force evaluation routines for specific situations (e.g., two and three-dimensional systems) and by removing all nonessential calculations from the tree scan. Finally, separate copies of the tree-walk subroutines were designed to remove a restriction present in the BH code. For large θ ($\theta \gtrsim 1.3$) the BH code (and the standard version of TREECODE) can unintentionally include a self-acceleration on a particle by not forcing subdivision of cells containing the particle. That is, if a particle is located near the edge of a cell (in which it resides) whose center of mass is located at the opposite edge of the cell, then the condition $s/d < \theta$ might be satisfied for this cell if $\theta \gtrsim 1$. However, the force on the particle will be inaccurate, since the particle itself will have contributed to the mass and center-of-mass coordinates of the node. The versions of the tree-walk subroutines that include forced subdivision allowed the severity of this problem to be investigated, and the results are described in § III.

c) *Force Calculation*

The pair force between individual *particles*, separated by a distance r, is obtained from the usual softened form of the Keplerian potential,

$$\varphi(r) = -\frac{m_1 m_2}{(r^2+\varepsilon^2)^{1/2}}, \tag{2.1}$$

where m_1 and m_2 are the particle masses and ε is the smoothing parameter (unless stated explicitly, it is assumed throughout that $G=1$).

The pair force between particles and nodes is determined from a multipole expansion of the cluster potential about its center of mass. In the monopole approximation the particle-node interaction is computed from equation (2.1) with m_2 chosen to be the mass of the cluster (if m_1 is the particle mass) and r equal to the distance from the particle to the center of mass of the cluster. Higher order terms in the particle-cluster potential can be included in a straightforward manner. Since the center of mass is used as the expansion center, the dipole term vanishes, and the lowest order correction is given by the quadrupole moment tensor (BH). For a collection of point particles in a bounded region the potential per unit mass at an exterior point $\boldsymbol{r}$ (relative to the center of mass of the cluster) is given by

$$\varphi(r) = -\frac{GM}{r} - \frac{1}{2}\frac{G}{r^5}\boldsymbol{r}\cdot\mathbf{Q}\cdot\boldsymbol{r}, \tag{2.2}$$

where M is the mass of the cluster and $\mathbf{Q}$ is the traceless quadrupole tensor, evaluated with respect to the center of mass, defined by

$$Q_{ij} = \sum_k m_k\left(3x_{k,i}x_{k,j} - r_k^2\delta_{ij}\right) \tag{2.3}$$

(e.g., Goldstein 1980; Jackson 1975). The acceleration at $\boldsymbol{r}$ corresponding to equation (2.2) is

$$\boldsymbol{a} = -GM\frac{\hat{\boldsymbol{r}}}{r^2} + \frac{G}{r^4}\mathbf{Q}\cdot\hat{\boldsymbol{r}} - \frac{5G}{2}(\hat{\boldsymbol{r}}\cdot\mathbf{Q}\cdot\hat{\boldsymbol{r}})\frac{\hat{\boldsymbol{r}}}{r^4}. \tag{2.4}$$

The quadrupole moment tensor of a cell can be obtained recursively from its subcells using the parallel-axis theorem (e.g., Goldstein 1980) to give

$$\mathbf{Q} = \sum_{l=1}^{n\text{subcell}} \mathbf{Q}_l + \sum_{l=1}^{n\text{subcell}} m_l\left(3\boldsymbol{R}_l\boldsymbol{R}_l - R_l^2\mathbf{1}\right), \tag{2.5}$$

where the index l labels the subcells, m_l is the mass of subcell l, $\boldsymbol{R}_l = \boldsymbol{X}_l^{\rm cm} - \boldsymbol{X}^{\rm cm}$ is the displacement vector from the center of mass of subcell l to the center of mass of the composite system, $\mathbf{1}$ is the unit matrix, and $\boldsymbol{Q}_l$ is the quadrupole moment tensor of subcell l. TREECODE allows quadrupole terms to be included or ignored as specified by an input parameter.

According to equation (2.5) the quadrupole moment tensor of a composite node consists of a sum over its subnodes, characterized by center-of-mass pseudoparticles (the second term in eq. [2.5]), and an additional part (the first term) describing the details of the mass distribution within each subnode. Thus the use of relation (2.5) will not be equivalent to descending an additional level in the tree in the monopole version, since the first term includes information from deeper sublevels.

d) *Relative Efficiency of TREECODE*

A number of runs were performed to assess the efficiency of TREECODE relative to the BH C version and to determine the fraction of time used by each section of the code. The

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

CPU breakdown for several simple test cases for $N = 8192$ is shown in Table 1 for both the monopole and quadrupole versions (results with other values of N were similar). (The computations were performed on the CRAY X-MP with the CFT compiler.) For the uniform-density models the particles were distributed homogeneously within a sphere of unit radius and mass. The Plummer models assumed $\rho \propto (r^2 + a^2)^{-5/2}$, with $a = 0.2$, and unit cutoff radius and mass. The various section headings correspond to the time spent in the tree-walk subroutine, the vectorized sum over nodes and particles that contribute to the force on a given particle, the collection of routines used to build and update the tree, and the remainder of the code, which advances the positions and velocities, performs input/output, and so on.

As can be seen from Table 1, the tree-walk subroutine dominates the CPU usage even though it has been heavily optimized. (This is not unexpected, since this subroutine, in its present form, consists entirely of scalar arithmetic.) Vectorization improved the efficiency of the force sum by a factor ~ 10. In nonvectoried form it required a time comparable to the tree search. The tree building typically requires a negligible amount of time ($\lesssim 10\%$ of the total time) compared with the force computation, in spite of the fact that the tree must be rebuilt at each step. Thus, it will be necessary to improve the efficiency of the force computation by a factor ~ 10 before the time spent updating the tree represents a significant burden.

Finally, the efficiency of the monopole version, running on the CRAY X-MP (compiled using CFT), relative to the BH C version, running on a VAX 11-780 (compiled under UNIX), was examined. The CPU ratios (VAX/CRAY) are shown in Table 2 for the homogeneous and Plummer profiles with $N = 1024$ (results for other values of N were similar) and for the tolerance parameter in the range $0.5 \leq \theta \leq 1.2$. (The ratios depend on both θ and the detailed form of the density profile, since the proportion of scalar to vector arithmetic is not constant in the CRAY version.)

There is a tendency for the performance of TREECODE to degrade, relative to the C version, with increasing θ. However, this appears to be a weak dependence for density profiles that are representative of astrophysical systems. Obviously it is impossible to generalize these conclusions to arbitrary density distributions. Nevertheless, it appears that it would not be unreasonable to expect a factor ~ 150–180 improvement in speed if TREECODE is used on a CRAY or other similar vector supercomputer.

TABLE 1

PERCENTAGE OF CPU TIME IN SECTIONS OF THE CODE

SECTION	PLUMMER		UNIFORM	
	$\theta = 1.0$	$\theta = 0.5$	$\theta = 1.0$	$\theta = 0.5$
Monopole Version				
Tree walk	81.2	88.3	70.3	85.6
Force sum	9.3	8.4	9.0	8.4
Tree build	9.4	2.5	13.1	3.8
Remainder ...	0.1	0.8	7.6	2.2
Quadrupole Version				
Tree walk	67.2	76.8	62.3	73.4
Force sum	22.7	20.4	20.5	21.6
Tree build	7.9	2.2	11.9	3.5
Remainder ...	2.2	0.6	5.3	1.5

TABLE 2

RELATIVE EFFICIENCY OF TREE CODE

θ	VAX/CRAY X-MP	
	Plummer	Uniform
0.5.....	183	178
0.6.....	190	172
0.7.....	187	165
0.8.....	186	158
0.9.....	178	148
1.0.....	179	150
1.1.....	178	138
1.2.....	173	131

III. PROPERTIES OF THE CODE

a) Scaling of CPU Time

For nonzero θ, BH argued that the tree method should scale as $\sim O(N \log N)$, since an increase in N by a constant factor will result in an increase in the number of terms in the force evaluation by a constant increment (see also Appel 1985). In the limit $\theta \to 0$, however, the behavior should approach that of a direct code, $\sim O(N^2)$. Thus, it is of interest to determine the typical range in θ for which the $N \log N$ trend is to be expected.

A large number of runs were made with N and θ in the ranges $1024 \leq N \leq 32{,}768$ and $0 \leq \theta \leq 2.0$, both with and without quadrupole terms in the force calculation. The tests presented below were all performed on the CRAY X-MP for uniformity, using the CFT compiler. Unfortunately, an analysis of the efficiency of this method is complicated by the fact that the required CPU time per step is sensitive to the detailed form of the density distribution. For example, the force computation for a sharply peaked density profile (such as that expected for an elliptical galaxy) will be more costly than that for a more homogeneous system, since the tree search will, on average, extend to deeper levels. As a compromise, spherical, isotropic Plummer models with density profiles

$$\rho(r) = \frac{3M}{4\pi} \frac{r_0^2}{\left(r^2 + r_0^2\right)^{5/2}} \tag{3.1}$$

were used for the timing tests. In equation (3.1), M is the mass of the system and r_0 is the scale length. The exact distribution function is known analytically in this case and is

$$f(E) = \frac{\sqrt{2}}{378\pi^3 G r_0^2 \sigma_0} \left(-\frac{E}{\sigma_0^2}\right)^{7/2}, \quad -6 \leq \frac{E}{\sigma_0^2} \leq 0, \tag{3.2}$$

where E is the energy per unit mass and $\sigma_0^2 = GM/6r_0$. The one-dimensional (isotropic) velocity dispersion is $\sigma(r) =$

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

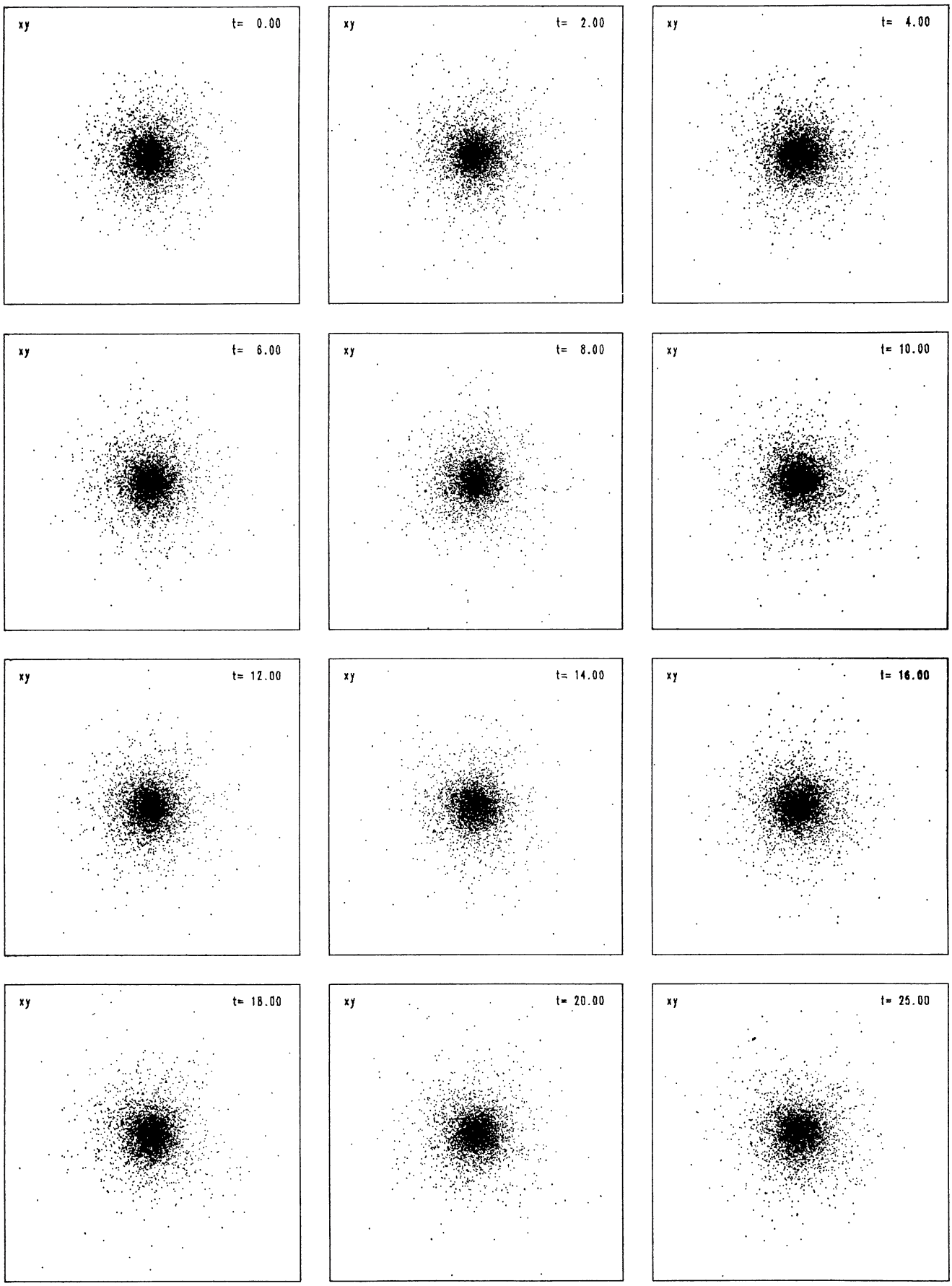

FIG. 1.—Evolution of a Plummer model with $N = 4096$ and $\theta = 1$, for 1000 time steps. Particle positions are plotted with respect to the center of mass of the system. The units of time are those appropriate for a system of unit size and mass, assuming $G = 1$.

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

$\sigma_0/[1+(r/r_0)^2]^{1/4}$. Initial conditions were generated directly from the distribution function (3.2) assuming a cutoff radius $R=1$, unit mass, $r_0=0.2$, and $G=1$.

The distribution function (3.2) is an exact solution to the Vlasov equation. In the absence of numerical errors and dynamical instabilities (e.g., Merritt and Aguilar 1985; Barnes, Goodman, and Hut 1986), therefore, models generated according to equation (3.2) should remain time-stationary. In order to verify the integrity of the tree method, several runs were made with a large number of time steps using Plummer profiles. An example is shown in Figures 1 and 2, which show the evolution of a system of unit size and mass with $r_0=0.2$ and $N=4096$. The tolerance parameter was chosen to be $\theta=1$, and quadrupole terms were not included in the force computation. The softening parameter, ε, was equal to the mean interparticle separation evaluated at the half-mass radius, $\varepsilon=0.032$ (see § III*c*). The system was evolved for 1000 time steps, each of length $\Delta t=0.025$, which corresponds to a total elapsed time of the order of 50 core crossing times. Figure 1 shows a projection of the particle density with respect to the center of mass of the system, while the radii (measured with respect to the center of mass) containing 10%, 50%, and 90% of the particles are given in Figure 2 as a function of time. (The variation in the position of the center of mass with respect to time is discussed in § III*d*.) As is evident from Figures 1 and 2, the approximations made in the force computation have not affected the behavior of the global properties of this model.

The timing estimates and analysis of the average number of force terms per particle presented below were obtained by averaging over a small ($\lesssim 10$) number of time steps in order to smooth out statistical fluctuations. Long-term simulations, such as that shown in Figures 1 and 2, indicate that the initial models generated from equation (3.2) are not precisely in equilibrium (owing to discreteness effects) and are subject to a brief transient period before settling into a steady state. For example, the average number of force terms per particle, $\langle n_{\rm terms}\rangle$, for the model in Figures 1 and 2 was approximately 170–175 for small t, but increased slowly to 195–200 during the first 50 time steps. Thereafter $\langle n_{\rm terms}\rangle$ fluctuated within the range 195–200 for the remaining 950 steps. Therefore, the estimates below may underestimate the actual CPU usage and average number of force terms in a long simulation by $\sim 10\%$.

The scaling of CPU time with N is shown in Figure 3 for several values of θ. Straight-line fits have been made to the curves with $0.4\leq\theta\leq 1.5$. The linear behavior for θ in this range indicates that the CPU time is indeed proportional to $aN\log N+bN$. Significant departures are not realized for this density distribution until $\theta\lesssim 0.3$. The dependence on θ for fixed N is shown in Figure 4. The CPU time per step per particle decreases rapidly for $\theta=0$ to $\theta\sim 1$, but much less slowly for $\theta>1$. This is a consequence of the behavior of the number of terms in the force evaluation due to the clustering of distant particles (see below). Based on considerations of the CPU time alone, it appears that $\theta\gtrsim 0.4$ is preferred. However, it must again be emphasized that Figures 3 and 4 are based on the density profile (3.1) and may not apply generally. For example, a uniform density profile gave similar scalings with N and θ, but with a reduction by a factor ~ 2 in CPU time used.

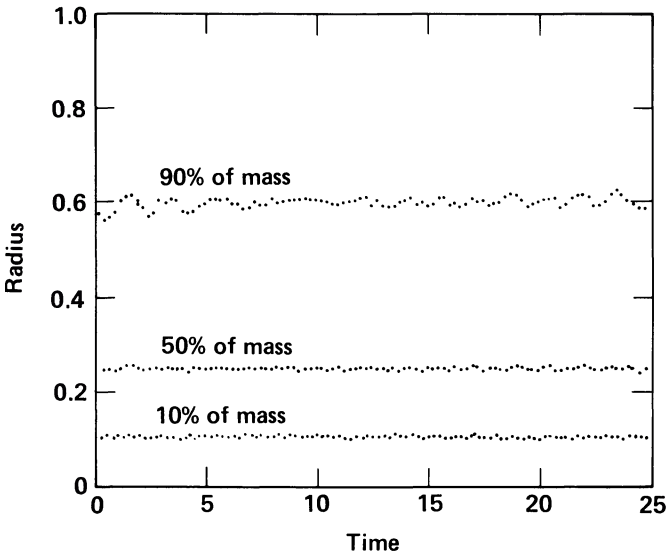

FIG. 2.—Behavior of the radii containing 10%, 50%, and 90% of the particles, with respect to the center of mass, for the model in Fig. 1, as a function of time, showing that the time-stationary nature of the distribution function is preserved by the tree method.

The tests summarized in Figures 3 and 4 were also performed with quadrupole terms included in the force computation. Typically the required CPU time was a factor ~ 1.5 times that for the monopole code, with a weak dependence on θ for small N ($N\lesssim 2000$) in the logarithmic regime. For $\theta\lesssim 0.3$, where departures from the $N\log N$ behavior are significant, the overhead associated with the quadrupole calculation declines, since the relative proportion of cells to particles among the terms that contribute to the force decreases. (A similar overhead and dependence on θ was also obtained for uniform density spheres.)

The mean number of terms in the force evaluation per particle, $\langle n_{\rm terms}\rangle$, was also determined for the models shown in Figures 3 and 4. The dependence of $\langle n_{\rm terms}\rangle$ on N was qualitatively similar to that of the CPU time. For $\theta=1.0$, $\langle n_{\rm terms}\rangle\approx 130$ for $N=1024$ and $\langle n_{\rm terms}\rangle\approx 220$ for $N=32{,}768$. For $\theta=0.5$, $\langle n_{\rm terms}\rangle\approx 350$ for $N=1024$ and $\langle n_{\rm terms}\rangle\approx 1060$ for $N=32{,}768$. The behavior with θ also closely matched that of the CPU time, shown in Figure 4. Thus, the rapid decrease in CPU time from $\theta=0$ to $\theta\sim 0.5$–1 follows directly from the clumping of distant particles in cells of various sizes. However, the improvement is less dramatic for $\theta\gtrsim 1$ because virtually all of the particles will have been clustered for $\theta\sim 1$, and a further increase in θ will be manifested primarily as a regrouping of cells.

A typical example of the distribution of the number of force terms for a system is shown in Figure 5*a* for a Plummer profile with $\theta=1$ and $N=32{,}768$. It is sharply peaked about the mean ($\langle n_{\rm terms}\rangle\approx 221$), with a steep decline at larger values and a smooth tail for smaller numbers of terms. (The particles with small values of $n_{\rm terms}$ are those located near the edge of the sphere where the density is low.) This structure is sensitive to the density profile, as can be seen from a comparison with Figure 5*b*, which shows the distribution of terms for a uniform sphere with $\theta=1$ and $N=32{,}768$. It is more tightly centered about the mean ($\langle n_{\rm terms}\rangle=121$) and does not have

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

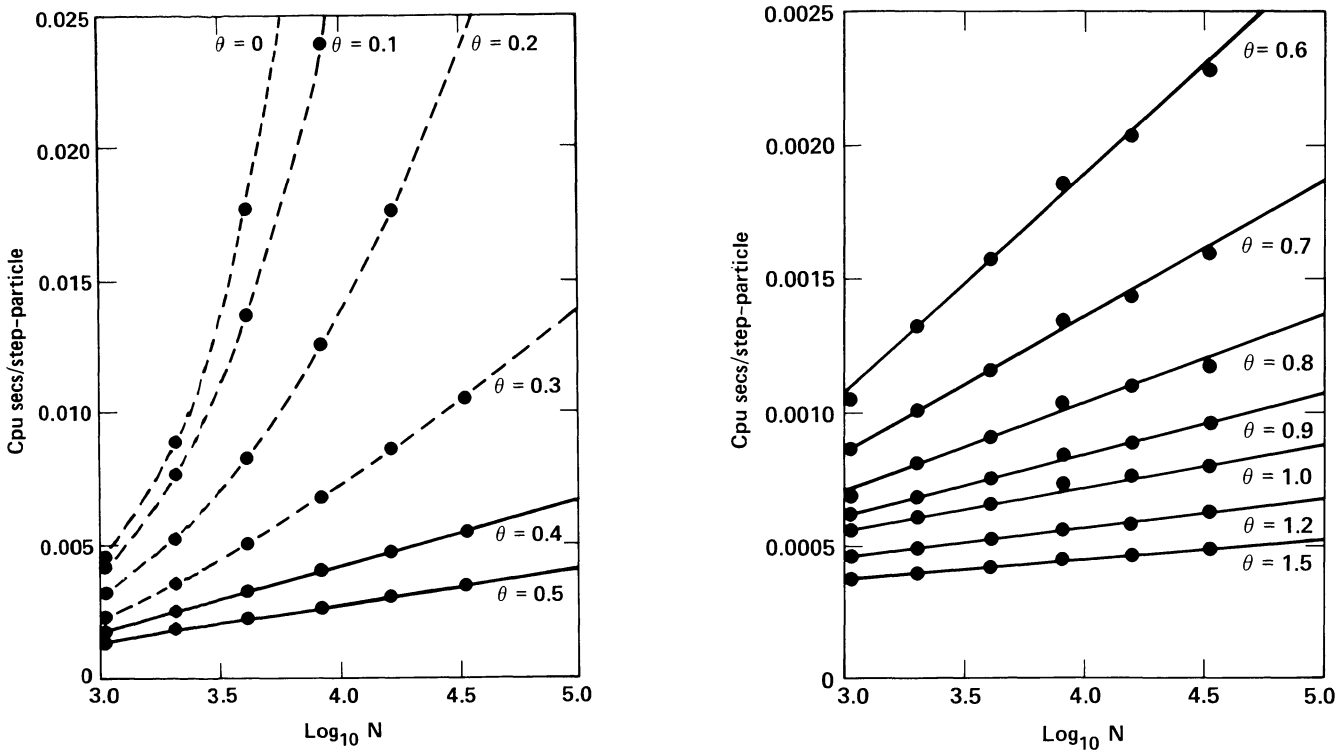

FIG. 3.—Scaling of CRAY X-MP CPU time (CPU seconds per step per particle) for spherical, isotropic Plummer models, as a function of the number of particles, for values of the clumping parameter θ in the range $0 \leq \theta \leq 1.5$. Only monopole terms have been included in the force computation.

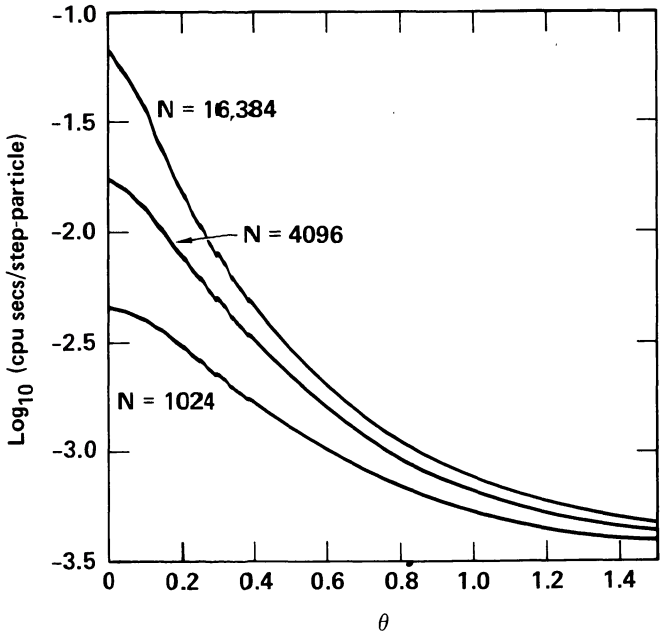

FIG. 4.—Dependence of the required CRAY X-MP CPU time (CPU seconds per step per particle) on θ for spherical, isotropic Plummer models for selected values of the particle number N. Only monopole terms have been retained in the force computation.

an extended tail at smaller numbers of terms, since the density gradient is zero except at the edge of the sphere.

b) Memory Requirements

The memory requirements of the code are determined by the number of bodies, N, the number of internal nodes (i.e., all nodes excluding the leaves) in the tree, $N_{\rm nodes}$, and the maximum number of terms in the force evaluation, $N_{\rm terms}$. The amount of memory used by the source code itself and the scalar variables is small for all values of N. Therefore, the total number of machine words, $N_{\rm words}$, required by the code can be estimated from the array variables, giving

$$N_{\rm words} = 3(1+n_{\rm dim})N + (2+n_{\rm dim}+2^{n_{\rm dim}})N_{\rm nodes} + (2+n_{\rm dim})N_{\rm terms} \quad (3.3)$$

for the monopole version, where $n_{\rm dim}$ is the number of spatial dimensions. If the quadrupole terms are also used, then the above estimate should be increased by $(2n_{\rm dim}+1)N_{\rm nodes} + 3n_{\rm dim}N_{\rm terms}$.

For an evenly balanced tree, the total number of levels, $n_{\rm levels}$, including the root node and leaves, is approximately $n_{\rm levels} \approx 1+\log_8 N$. Summing over all levels but that containing the leaves gives $N_{\rm nodes} \approx (N-1)/7$. For density distributions that are representative of astrophysical systems, however, it appears that typically $N_{\rm nodes} \sim 0.5N - N$. For useful values of θ, $\theta \gtrsim 0.3$, $N_{\rm terms} \ll N$. Hence, a worst-case estimate with $N_{\rm terms} \sim 0$ and $N_{\rm nodes} = N$ is, for three-dimensions, $N_{\rm words}$

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

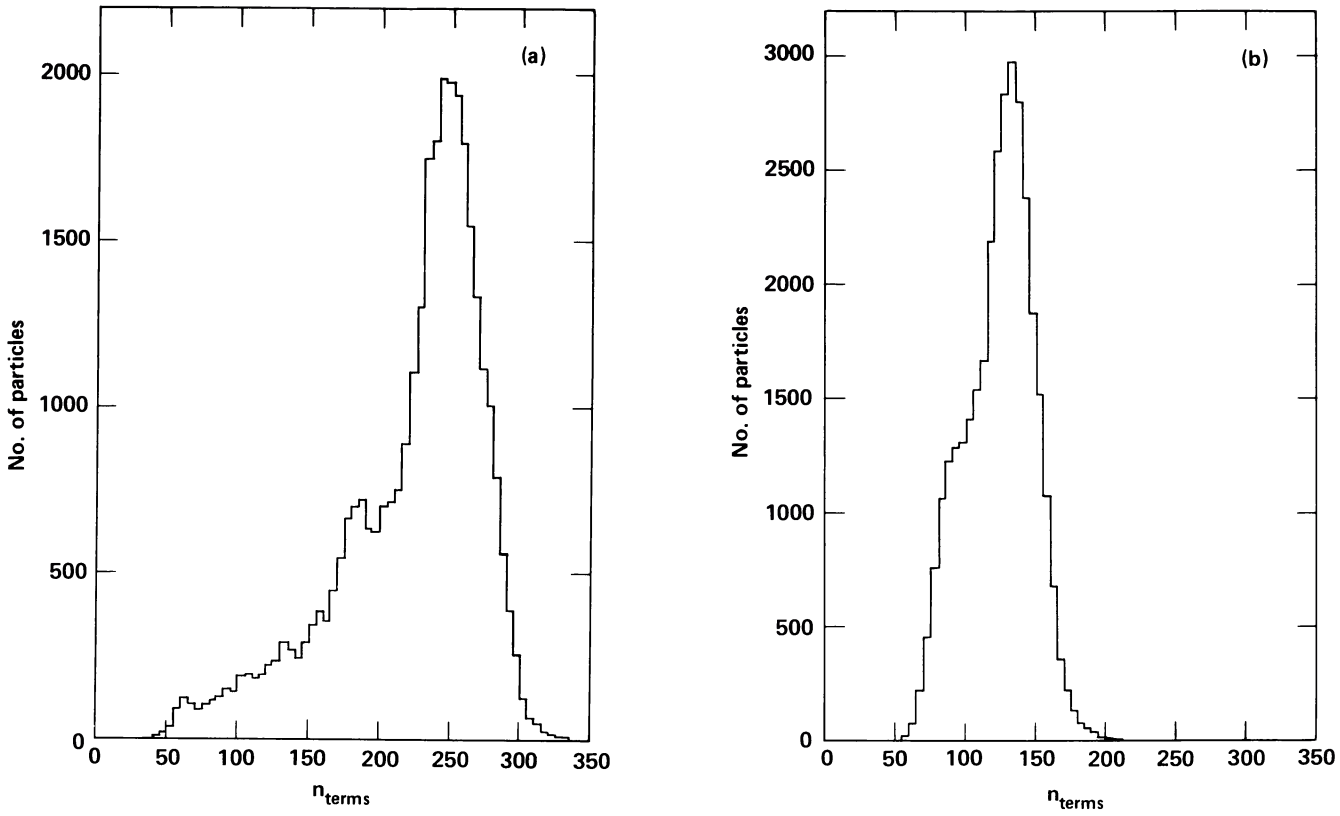

FIG. 5.—Distribution of the number of force terms for two density profiles with $N = 32{,}768$ and $\theta = 1.0$. Distributions of (a) a Plummer model with mean number of terms $\langle n_{terms} \rangle = 221$ and (b) a uniform sphere with $\langle n_{terms} \rangle = 121$.

$= 25N$ and $N_{words} = 30N$ for the monopole and quadrupole versions, respectively. Thus, the timing estimates presented in § IIIa indicate that TREECODE will usually be CPU-limited, and not memory-bound, on a machine such as a CRAY X-MP 48.

The FORTRAN implementation is somewhat less memory-efficient than the BH C code. The CPU efficiency has been enhanced by saving intermediate results, such as the acceleration and potential of the bodies, for use in vectorized loops. This does not appear to be serious, however, since memory charges are usually small compared with CPU costs.

c) *Errors in the Force Computation*

Because of the approximate nature of the force calculation, the acceleration on each particle will deviate from that which would have resulted from a sum over all pair interactions. In order to estimate the dependence of this effect on the parameters, a large number of runs, both with and without quadrupole terms, were performed for varying N, θ, and ε. At each step, the accelerations computed through the tree algorithm were compared with those obtained by a direct sum over binary interparticle forces. The typical error was estimated from the moments of the distribution of errors for an entire system. Specifically, for each component of the acceleration, a_i, the mean error, $\overline{\delta a_i}$, and mean absolute deviation, $A(\delta a_i)$, defined by

$$\overline{\delta a_i} = \frac{1}{N} \sum_{j=1}^{N} \left(a_{i,j}^{\text{tree}} - a_{i,j}^{\text{direct}} \right), \tag{3.4}$$

$$A(\delta a_i) = \frac{1}{N} \sum_{j=1}^{N} \left| a_{i,j}^{\text{tree}} - a_{i,j}^{\text{direct}} - \overline{\delta a_i} \right|, \tag{3.5}$$

were computed. [The standard deviation was also measured and was typically $\sim 30\%$–50% larger than $A(\delta a_i)$. For a discussion on the use of the mean absolute deviation to characterize the widths of distributions see Press *et al.* 1986.] In the analysis presented below, spherical, isotropic Plummer models of unit mass and cutoff radius were used with $r_0 = 0.2$ (see eqs. [3.1] and [3.2]).

In all cases, the mean of the error was found to be much smaller (in magnitude) than the mean absolute deviation—typically by a factor ~ 100. In addition, the means were centered about zero, with random sign. (It is problematical whether or not $\overline{\delta a_i}$ should vanish identically in the limit $N \to \infty$. The error in the acceleration on a particle is determined by the lowest order term not included in the multipole expansion of the cluster potentials. Typically these terms will be small in comparison with the lower order terms for reasonable values of θ, and will have arbitrary sign

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

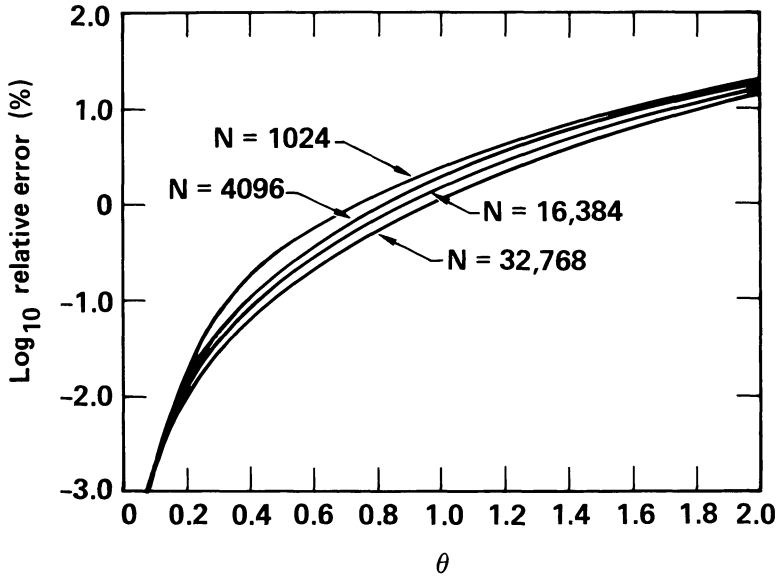

FIG. 6.—Magnitude of the typical error (in percent) in the tree force computation, relative to a direct sum, as a function of θ, for selected values of the particle number N. The calculations have assumed spherical, isotropic Plummer models with softening parameter $\varepsilon = 0$, and only monopole terms have been included in the force computations.

depending upon the detailed form of the mass distribution within each cell and the orientation of each cell with respect to a given particle. The mean error [3.4] is an average over the system, each term of which represents an average over the errors from individual cells. Intuitively, it does not seem unreasonable that $\overline{\delta a_i} \to 0$ as $N \to \infty$, but a detailed proof could be sensitive to the form of the global density profile.) Thus the distribution of errors is described by a profile which is much wider than its offset from the origin. The accumulation of errors, therefore, will be dominated by the fluctuations, $A(\delta a_i)$, and not the secular contribution, $\overline{\delta a_i}$, which is characteristic of a random walk (BH). The relatively slow evolution of the error implied by this conclusion is, in part, responsible for the behavior of the conservation properties of this method (§ III*d*).

The relative magnitude of the errors in the acceleration was determined by comparing the mean absolute deviation, $A(\delta a_i)$, with the absolute average acceleration computed by a direct sum,

$$\overline{a_i} = \frac{1}{N} \sum_{j=1}^{N} \left| a_{i,j}^{\text{direct}} \right| .$$

The ratio $A(\delta a_i)/\overline{a_i}$, expressed as a percentage, is shown in Figure 6 as a function of θ for several values of N, using the monopole force calculation. This particular set of calculations assumed $\varepsilon = 0$. (The dependence of the error on ε is discussed below.) The error increases monotonically with θ as fewer terms are included in the force calculation. For reasonably large N ($N \gtrsim 10{,}000$) the typical error in the acceleration will be $\lesssim 1\%$ for $\theta \lesssim 1$. The errors quickly become intolerable for $\theta > 1$.

As indicated by Figure 6, the error appears to decrease slowly with increasing N. For the Plummer models described in this section the error behaved roughly as $A(\delta a_i) \sim N^{-1/5}$. This may be a result of the mean field nature of the expansion of the cluster potentials. That is, consider a spherical distribution of N particles in a bounded region and an exterior particle, p. In the limit $N \to \infty$, the potential at p will be, by Gauss's theorem, $-M/r$, where M is the mass of the sphere and r is the distance from p to the center of mass of the sphere. The tree method will, as a result of the hierarchical clustering, predict a potential at p, $\varphi = -M/r$ for *arbitrary* N (assuming an appropriate choice of θ). A direct force calculation will yield a result differing from $\varphi = -M/r$, for finite N, owing to departures from spherical symmetry. As $N \to \infty$, however, the importance of the fluctuations decreases and the error in the tree calculation, relative to a direct sum, tends to zero.

The relative improvement in the accuracy of the force computation when quadrupole terms are included is shown in Figure 7 for a Plummer model and a uniform sphere, each assuming $N = 16{,}384$ and $\varepsilon = 0$. The acceleration computed with the quadrupole version is significantly more accurate for small θ, $\theta \lesssim 1.0$. As θ increases, the quadrupole calculation degrades with respect to the monopole result, and the errors become comparable for $\theta \sim 1.5$. This behavior is not surprising. The multipole expansion is essentially a power series in the ratio s/d, where s is the size of a cell and d is the distance from the cell to the particle on which to calculate the force. If $s/d \gtrsim 1$, as implied by $\theta \gtrsim 1$, the series will not converge and higher order terms are not guaranteed to improve the accuracy of the computation.

It appears from Figure 7 that the relative efficiency of the monopole and quadrupole versions, at fixed accuracy in the force computation, may be highly model-dependent. As noted in § III*a*, the quadrupole version is approximately 50% more costly than the monopole version for all N and θ. Thus, Figures 4, 6, and 7 indicate that for the $N = 16{,}384$ Plummer model the quadrupole code with $\theta = 1.0$ is roughly as efficient

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

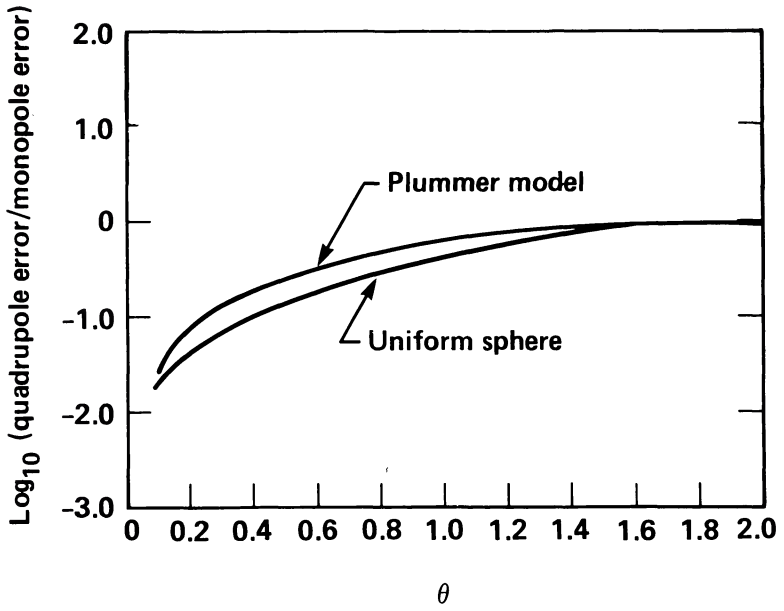

FIG. 7.—Ratio of the error in the monopole force calculation to that resulting if quadrupole terms are included, as a function of θ, for spherical, isotropic Plummer models and uniform spheres with $N = 16{,}384$.

and as accurate as the monopole code with $\theta = 0.8$. For $\theta = 0.5$, the quadrupole code is as accurate as the monopole code with $\theta = 0.3$. However, because of the rapid rise in required CPU time at small θ, the monopole calculation will be approximately twice as costly. For the uniform spheres the gain in efficiency is more impressive. In this case the quadrupole code with $\theta = 1.0$ is $\sim 50\%$ less costly than the monopole version at a similar accuracy and a factor of ~ 5 less costly if $\theta = 0.5$. On this basis it appears that it may be difficult to predict the gain in efficiency for an arbitrary density profile. However, since the quadrupole calculation represents a relatively modest overhead, it will generally be useful to include these terms unless memory constraints are severe (see § III*b*).

As noted in § II*b*, the original BH C code and the standard version of TREECODE admit the possibility of including a spurious self-acceleration on particles for large $\theta, \theta \gtrsim 1$. This effect was investigated by developing separate versions of the tree-walk subroutines that forced subdivision of cells containing the particle on which to compute the force, and by repeating the error measurements shown in Figures 6 and 7. It was found that the typical error in the force computation from this effect was completely negligible for $\theta \lesssim 1.2$.

Finally, the dependence of the error on the softening parameter ε was studied. For the Plummer models defined by equations (3.1) and (3.2) the natural scale length for ε is the mean interparticle separation, measured at the half-mass radius. For a truncated Plummer model with unit cutoff radius, the half-mass radius, $r_{1/2}$, is approximately $r_{1/2} \approx 1.24 r_0$. The mean interparticle separation, evaluated at $r_{1/2}$, for a collection of N particles, is then $\lambda \approx 2.5 r_0 N^{-1/3}$. The mean absolute deviation for the case $r_0 = 0.2$, $N = 16{,}384$, as a function of ε/λ, is shown in Figure 8 for $\theta = 0.5$ and $\theta = 1.0$, both with and without quadrupole terms. (Results for other values of N were similar.)

For $\varepsilon/\lambda > 1$, the error in the acceleration increases rapidly. This is not unexpected. The multipole expansion which has been assumed is actually appropriate for point particles, requiring $\varepsilon \ll d$, where d is the typical distance from a particle to the center of mass of a cell which contributes to the force. For large ε/λ this condition is no longer satisfied, and the force calculation deteriorates rapidly. Indeed, the discrepancy between the particle density profile implied by a smoothed potential and that of a point particle is so severe for $\varepsilon/\lambda \gtrsim 1$ that the quadrupole correction ceases to improve the accuracy of the force computation. (It was found that the ad hoc softening of the quadrupole terms in eq. [2.4], accomplished by the substitution $r^4 \rightarrow (r^2 + \varepsilon^2)^2$, reduced the error relative to the direct sum over pair interactions. The results shown in Fig. 8 include this adjustment.)

(For uniform spheres with $N = 16{,}384$ the accuracy of the force calculation similarly degraded with increasing ε. However, the quadrupole version remained more accurate for a much larger range in ε. It would appear that the relative merits of the use of quadrupole terms are, in this context, again model-dependent.)

The rapid rise in the error for $\varepsilon \gtrsim \lambda$ is, in part, due to the chosen form of the softening implied by equation (2.1). As noted above, the multipole expansion of the cluster potentials is valid only for point particles and is not suitable for large ε. The softened potential (2.1) converges slowly to the Keplerian result, $\varphi_K \propto -1/r$. For $r \gg \varepsilon$,

$$\frac{\varphi_{\text{soft}}}{\varphi_K} \sim 1 - \frac{1}{2}\frac{\varepsilon^2}{r^2} + O\left(\frac{\varepsilon^4}{r^4}\right).$$

The error in the force computation can be reduced, for nonzero ε, if the softened potential approaches the Keplerian

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

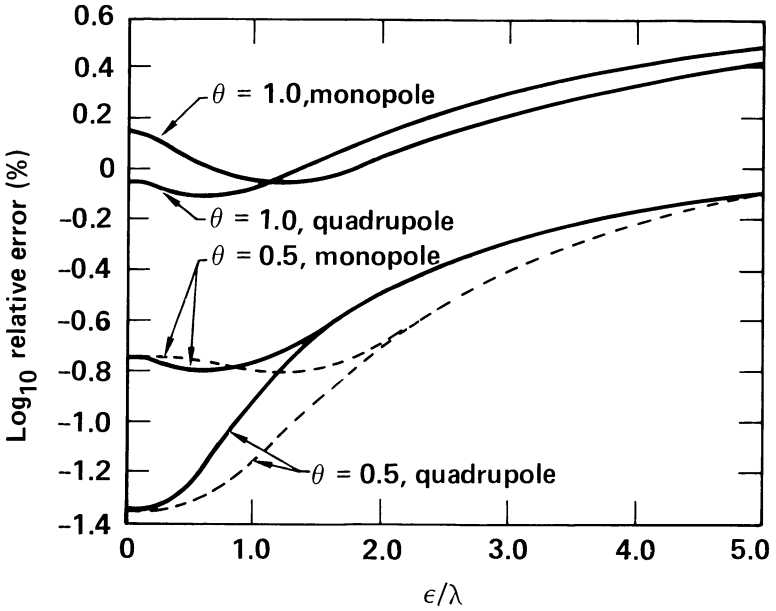

FIG. 8.—Dependence of the typical error in the force calculation (in percent), relative to a direct sum, on the smoothing parameter ε for $\theta = 1.0$ and $\theta = 0.5$, both with and without quadrupole terms. The results have assumed the Plummer models described in Fig. 7. The softening parameter is measured in units of the mean interparticle separation λ, evaluated at the half-mass radius of the density distribution. Dashed curves show the effect of assuming a smoothed potential $\varphi \propto (r^4 + \varepsilon^4)^{-1/4}$ for the case $\theta = 0.5$ (see text).

form more rapidly with increasing r, since the density profile will more closely resemble that of a point particle. In order to investigate this possibility, the error analysis shown in Figure 8 for the softened potential (2.1) was repeated with $\varphi \propto (r^4 + \varepsilon^4)^{-1/4}$, which converges to the Keplerian form according to

$$\frac{\varphi_{\text{soft}}}{\varphi_{\text{K}}} \sim 1 - \frac{1}{4}\frac{\varepsilon^4}{r^4} + O\left(\frac{\varepsilon^8}{r^8}\right)$$

for $r \gg \varepsilon$. The error was not significantly affected for large θ, $\theta \sim 1$. However, for smaller θ, $\theta \sim 0.5$, the error was reduced for $1.0 \lesssim \varepsilon/\lambda \lesssim 3.0$, as can be seen from the dotted curves in Figure 8. These results are not conclusive, but suggest that other softening profiles may be more appropriate for the tree method, at least for some range in θ. The potential $\varphi \propto (r^4 + \varepsilon^4)^{-1/4}$ was considered for simplicity, but is somewhat unappealing, since the implied density distribution, $\rho \propto r^2/(r^4 + \varepsilon^4)^{9/4} \sim 0$ as $r \to 0$. Perhaps the best choice would be a Gaussian smoothing kernel $\rho \propto \exp(-r^2/\varepsilon^2)$, since it converges uniformly to the point-particle result in the limit $\varepsilon \to 0$. (For a discussion see, for example, Lucy 1977; Gingold and Monaghan 1977,1982; Quinn 1982; van Albada 1986.)

d) *Conservation Properties*

As with other N-body techniques, numerical errors in the tree method will lead to a violation of the conservation of energy, linear momentum, and angular momentum. In general, several factors can contribute to the breakdown of the conservation laws, including round-off errors, approximations in the force computation, and truncation errors (for a discussion see Hockney and Eastwood 1981). The PP method conserves both linear and angular momentum exactly since no approximations are made in the evaluation of pairwise interactions. (In the present context, "exactly" will imply to within round-off accuracy.) However, energy conservation is weakly violated because of truncation errors in the time integrator. PM codes with Cartesian grids conserve linear momentum exactly if the grid density assignment and force interpolation functions are identical, but do not conserve angular momentum or energy. The expansion method weakly violates all three conservation laws (e.g., White 1982).

In view of the physical interpretation of the tree method as a synthesis of the PP and expansion techniques, it is not surprising that tree codes will also violate all of the laws for nonzero θ. For example, consider a single particle interacting with an isolated cluster that is not subdivided. The acceleration on each particle in the cluster from the single particle will be that appropriate for a particle-particle interaction. However, the influence of each particle in the cluster on the single particle will be included in the potential expansion of the cluster. Therefore, Newton's third law is not obeyed by the tree method, since the force between the single particle and each cluster particle is not symmetric. As a result, linear and angular momenta will not be conserved exactly. In addition, the approximations made in the force computation will contribute to the nonconservation of energy. In the limit $\theta \to 0$, however, tree codes reduce to the PP method. Hence it is of interest to determine the influence of the hierarchical clustering on the conservation properties.

A number of runs were made with Plummer models for a large (~ 1000) number of time steps. In particular, the simulation shown in Figures 1 and 2, with $N = 4096$ and $\varepsilon = \lambda = 0.0317$, was repeated for $\theta = 0$ and $\theta = 0.5$. (The model with $\theta = 0$ was evolved using both a PP code and the tree code to

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

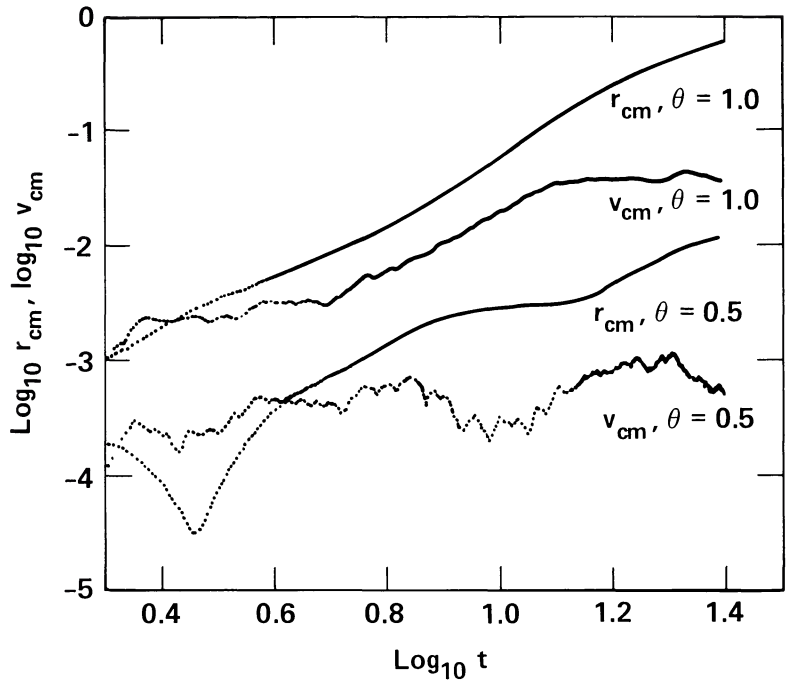

FIG. 9.—Radial position of the center of mass and center-of-mass velocity as function of time for the $N = 4096$ Plummer models evolved for 1000 time steps, each of duration $\Delta t = 0.025$, using $\theta = 1$ and $\theta = 0.5$. The dimensionless system of units is assumed.

verify that the correct limit was obtained as $\theta \rightarrow 0$. The results described below for $\theta = 0$, therefore, apply to either method.) The time step was chosen to be $\Delta t = 0.025$, which is approximately 5% of the core crossing time.

For the direct ($\theta = 0$) calculation, energy was conserved to approximately 0.20%, initially equal to $E_i = -0.8101$ and gradually decreasing to $E_f = -0.8085$ by the end of the simulation. This weak violation of the conservation of energy resulted from truncation errors in the time integration. (Throughout, the simple leapfrog scheme adopted by BH was used.) A similar, but slightly larger, effect was observed for the tree code. For $\theta = 1$, the energy was initially $E_i = -0.8116$ and decreased to $E_f = -0.8061$ after 1000 steps, representing a total change of $\Delta E/E = 0.68\%$. The corresponding results for the $\theta = 0.5$ simulation were $E_i = -0.8106$, $E_f = -0.8080$, and $\Delta E/E = 0.32\%$. (The initial values for E differ because of the approximations made in the force computation.) Thus the tree method conserves energy to an accuracy comparable to that of a PP calculation. (This effect was also noted by BH in a simulation of two colliding galaxies.) For reasonable values of the time step and of θ ($\theta \lesssim 1$), the contribution to the nonconservation of energy from the approximations in the force computation appears to be comparable to that due to truncation errors in the time integrator. Note, however, that unlike the PP technique, tree codes will not conserve energy exactly in the limit $\Delta t \rightarrow 0$.

Tree codes also appear to violate the conservation of angular momentum only weakly. The angular momentum, $\boldsymbol{l}$, vanishes for an isotropic Plummer potential. For finite N, $\boldsymbol{l}$ will be nonzero because of discreteness effects. In the simulations with $N = 4096$ described in this section, the magnitude of $\boldsymbol{l}$ was initially $l_i = 3.59 \times 10^{-3}$ and was conserved exactly by the direct calculation, as expected. The final values of l for the tree code with $\theta = 1$ and $\theta = 0.5$ were $l_f = 5.5 \times 10^{-3}$ and $l_f = 3.68 \times 10^{-3}$, respectively. The fact that the percent change in l was large ($\sim 50\%$ for $\theta = 1$) is not meaningful, since $l \approx 0$. A more convincing demonstration of the robust nature of the conservation of angular momentum is provided by a simulation of a system with large $\boldsymbol{l}$. For an isolated disk galaxy simulation with $N = 32{,}768$, l_z was conserved to within $\sim 0.01\%$ for 500 time steps, each of duration comparable to that assumed for the Plummer models, with $\theta = 1$ (see § V*b* for details).

The conservation law that appears to be most strongly violated by the tree method is that of linear momentum. The simulation using the PP technique conserved momentum exactly, and, hence, the center of mass of the system remained fixed at the origin. The lack of reciprocity in the force calculation between particles and cluster particles in the tree algorithm (see above) causes the center of mass to wander stochastically. An example of this effect is given in Figure 9, which shows the radius and magnitude of the velocity of the center of mass, as a function of time, for the simulations with $\theta = 1$ and $\theta = 0.5$. (Note that for a system of unit mass the center-of-mass velocity and momentum are identical.)

As is apparent from Figure 9, the nonconservation of momentum can lead to a significant drift in the position of the center of mass for large θ, $\theta \sim 1$, and a large number of time steps. The error is reduced dramatically for smaller θ. For $\theta = 0.5$ the center of mass wandered by only $\sim 1\%$ of the initial system size after 1000 steps, and the center-of-mass velocity remained small ($\sim 10^{-3}$) and roughly constant during the simulation. The center-of-mass acceleration was also computed at each step and was randomly scattered with time,

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

showing no significant correlation from one time step to the next.

It is unclear whether or not the nonconservation of momentum in the tree method is a serious problem. Although the center of mass of the model evolved with $\theta = 1$ drifted significantly, the internal dynamics of the system were not perceptibly affected (see Figs. 1 and 2), in comparison with the PP calculation. Obviously, however, it is not possible to generalize this assertion to arbitrary density distributions. If the violation of momentum conservation appears to be a serious issue, a smaller value of θ should be used. (An important advantage of the tree method is that the magnitude of the error is controllable.) Another possibility might be to constrain the center of mass to remain fixed by renormalizing the particle positions and velocities at each step. It would be necessary, however, to demonstrate that such a procedure did not introduce spurious dynamical effects into the simulations.

A number of tests were also performed for varying N and Δt. As expected, the conservation laws were less seriously violated for larger N, since the error in the force computation declines weakly with increasing N (see § III*c*). The use of a smaller time step improved the conservation of energy but did not significantly influence the violation of momentum conservation. Thus energy conservation is affected by both Δt and θ, while the accuracy of the conservation of linear and angular momentum is controlled by θ.

IV. RELAXATION EFFECTS

a) *Issues*

A serious issue confronting the tree method concerns the level of collisionality present in these codes. As noted by BH, the clustering of particles is, in a sense, equivalent to replacing groups by single, massive pseudoparticles. Thus, for nonzero θ it may appear to each particle that a system of size N has been replaced by a virtual system of size comparable to the average number of terms in the force computation. Since the relaxation time, t_r, scales approximately as the number of bodies (e.g., Chandrasekhar 1942), there exists a possibility that t_r could be reduced substantially relative to the value expected in a similar PP calculation.

A number of procedures have been suggested to estimate the relaxation time in N-body codes. These have included measurements of the energy exchange rate between different mass components (e.g., Aarseth 1966; Wielen 1967; Hohl 1973), and the mean square energy changes experienced by individual particles in equal mass systems (e.g., White 1978,1979; Smith 1981; Casertano, Hut, and McMillan 1987). For present purposes, a variation on the method proposed by Standish and Aksnes (1969) has been adopted. In this technique deflections suffered by a single test particle passing through a system of N field particles, fixed in space, is used to determine the relaxation time statistically. This approach is not completely rigorous, since it ignores dynamical effects such as polarization (e.g., Aarseth 1971; Hénon 1973; Aarseth and Lecar 1975). Nevertheless, it directly addresses, in a particularly simple and elegant manner, the fundamental issue of whether or not the hierarchical clustering increases the graininess of the force field. (Since the pseudoparticles cannot communicate with one another directly and are, therefore, not dynamical entities, the possibility of an enhanced graininess in the force field is the primary source for concern.)

b) *Method*

Following Standish and Aksnes (1969), N particles of identical mass were distributed uniformly within a sphere of unit radius and mass according to $n = 3N/4\pi$, where n is the number density. A test particle of zero mass was placed at the edge of the sphere on an inward radial orbit, directed toward the center of mass (the origin), with initial velocity v_0. The orbit of the test particle was then integrated numerically, using TREECODE to compute the force, until it emerged from the sphere. In the mean field limit, $N \to \infty$, the particle would pass through the sphere on a straight-line trajectory with constant velocity. For finite N the test particle will be deflected due to the graininess in the potential.

The relaxation time is estimated numerically from the sum of the deflections incurred during the passage of the test particle through the sphere, from $t_r = \Delta t / \Sigma_i \sin^2 \phi_i$, where Δt is equal to the transit time and the sum is over all deflections, each of angle ϕ_i. If $\Sigma_i \sin^2 \phi_i \ll 1$, and it is assumed that the binary encounters are independent, then $\Sigma_i \sin^2 \phi_i \simeq \sin^2 \Phi$, where Φ is the total deflection angle during the crossing of the sphere (Standish and Aksnes 1969). (The validity of this approximation has been verified numerically by Lecar and Cruz-Gonzalez 1972.) In order to reduce numerical noise, the relaxation time is computed as an ensemble average over many test particle orbits according to

$$t_r = \frac{\langle \Delta t \rangle}{\langle \sin^2 \Phi \rangle}, \tag{4.1}$$

where $\langle \Delta t \rangle$ is the average transit time and $\langle \sin^2 \Phi \rangle$ is the average of $\sin^2 \Phi$ over the individual trajectories (Standish and Aksnes 1969). Given m samples, each of k experiments, the standard deviation of t_r can be estimated from the standard deviations $\sigma_{\Delta t}$ and $\sigma_{\langle \sin^2 \phi \rangle}$ obtained from the m mean values. Since the time spent generating the distribution of field particles was negligible, a different set of initial conditions was computed for each test particle trajectory. This procedure eliminates spurious correlations that might be introduced from using the same set of field particles with a randomly chosen orientation for the initial test particle velocity.

c) *Results*

Tests were performed for a range in N and ε, both with and without quadrupole terms in the force calculation, to determine the dependence of the relaxation time on θ. A representative example of the results is shown in Figure 10 for the case $N = 4096$ with $\varepsilon = 0$ and $\varepsilon = \lambda$. (For uniform spheres, the mean interparticle separation, λ, is $\lambda \approx 1.6 N^{-1/3} \approx 0.101$ for $N = 4096$.) In each case the initial particle velocity was $v_0 = \sqrt{2}/2$, which is one-half the escape velocity at the edge of the sphere. (Results for other N and v_0 were similar.) For the runs with $\varepsilon = \lambda$ the ensemble averages were computed from 10 samples, each of 200 test particle orbits. For $\varepsilon = 0$ and $\theta = 0$ result was obtained from 10 samples, each of 500 test particle

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

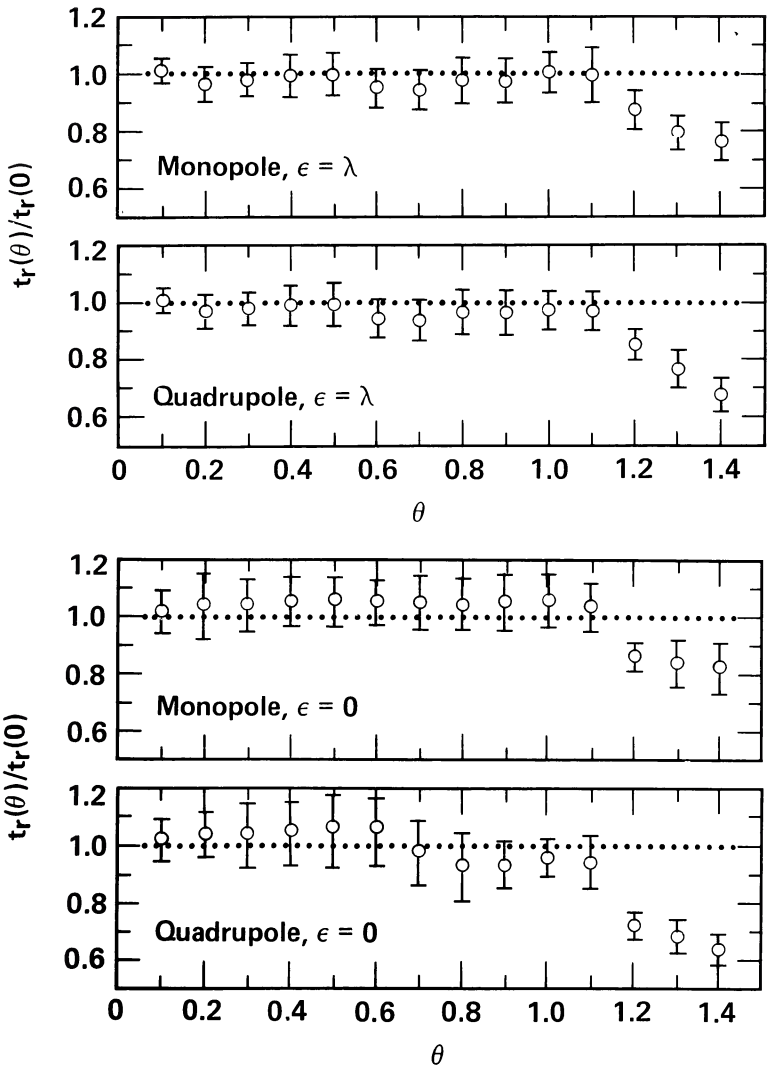

FIG. 10.—Relaxation time, t_r, relative to that in a direct ($\theta = 0$) calculation, as a function of θ, for $N = 4096$. The four panels show the results for the softening parameter equal to the mean interparticle separation and equal to zero, each with and without quadrupole terms in the force evaluation. The error bars are 2σ in length.

trajectories, while the averages for $\theta > 0$ were performed using 5 samples, each of 200 test particle orbits. In each case the time step was chosen to be sufficiently small that the results had fully converged.

The important feature of Figure 10 is that, given the statistical uncertainties in $t_r/t_r(0)$, the relaxation time in the tree calculation does not differ greatly from that of a PP code for $\theta \lesssim 1.2$. Only for large θ, $\theta \gtrsim 1.2$, is the level of collisionality noticeably higher in the tree method than with the direct technique. Furthermore, the reduction of t_r for $\theta > 1.2$ is small, of order $\sim 20\%$–40% for $\theta = 1.4$, in spite of the large errors present in the force computation.

V. COMPARISON WITH OTHER METHODS

a) PP Method

A simple direct $O(N^2)$ code was written in order to compare the PP and tree methods. Although it is possible to reduce the order of the PP technique through various refinements (see § I), the most naive implementation offers several advantages for CRAY-class machines. Vectorization of the inner loop in the force calculation gains a factor ~ 10 improvement in efficiency for the entire code. As a result, the $O(N^2)$ approach has proved to be useful in astrophysical applications (e.g., Quinn, Salmon, and Zurek 1986).

Several models were evolved for a large number of time steps using both the PP and tree codes (see § III*d*). As expected, the results were identical at early times for $\theta = 0$, but differed slightly for nonzero θ owing to the approximations made in the tree force computation. For example, the behavior of the radii containing 10%, 50%, and 90% of the mass for the $N = 4096$ Plummer model, shown in Figures 1 and 2, agreed statistically, but the individual features, such as the oscillations in the 90% mass radius, were not identical. This is not a cause for concern, since the PP method itself can only be interpreted statistically because of the propagation of

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

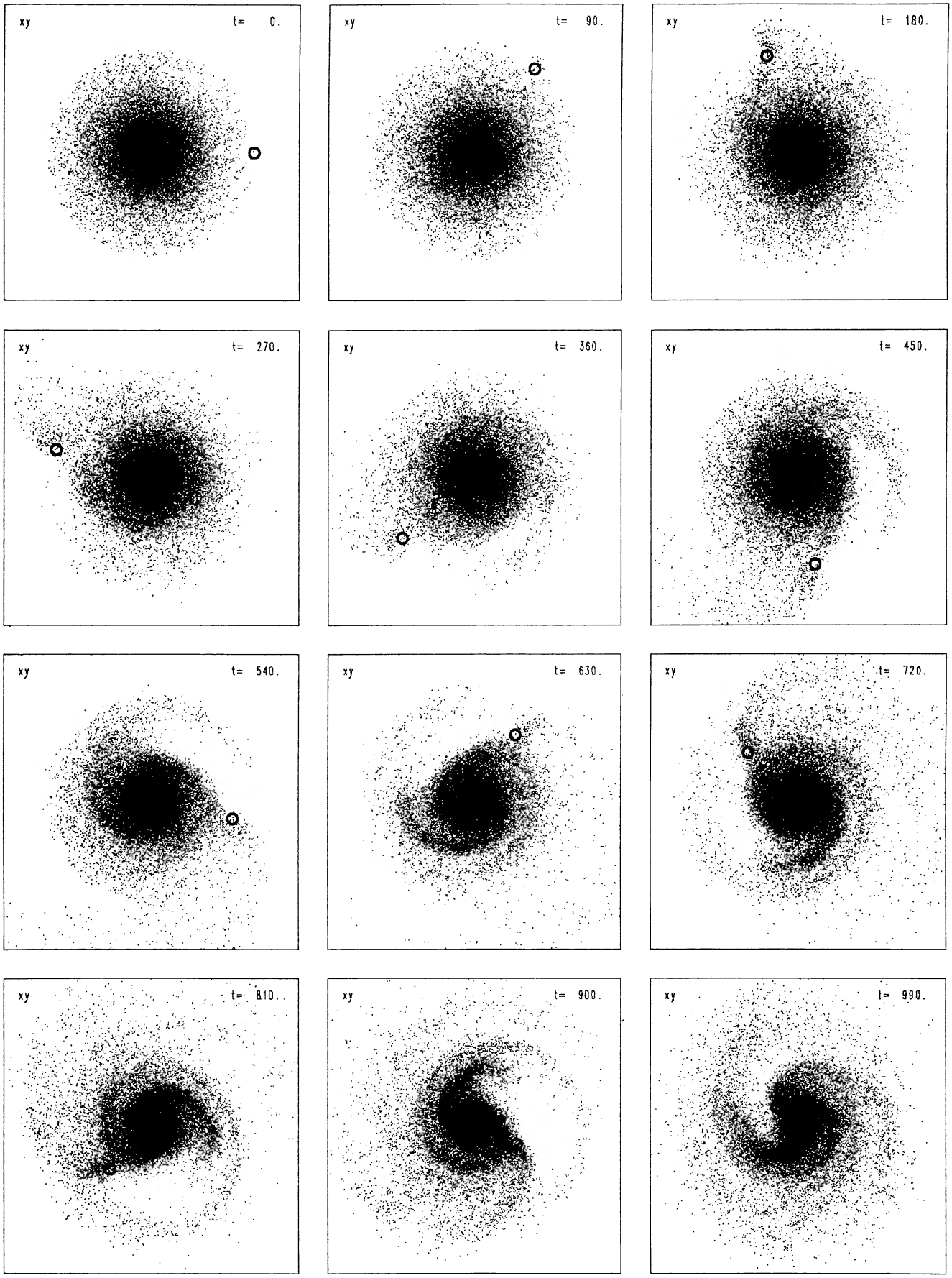

FIG. 11.—Decay of a satellite in a direct coplanar orbit around an exponential disk. Time is measured in units of 10^6 yr, and each frame is 60 kpc across. The final three panels shows the evolution of the system from an orthogonal projection.

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

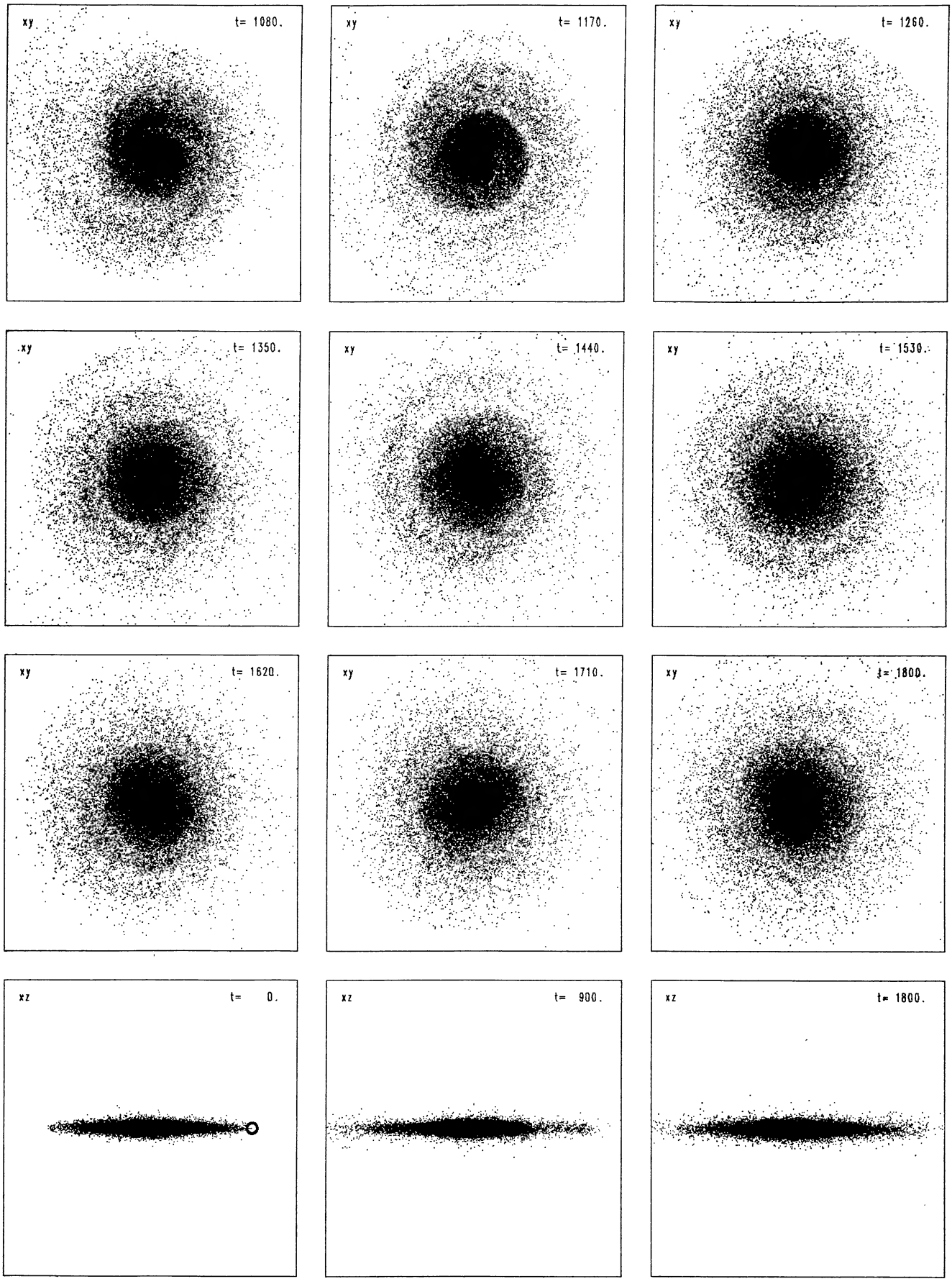

FIG. 11—*Continued*

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

numerical errors in the orbits of individual particles (for a discussion see, e.g., Aarseth and Lecar 1975).

The relative efficiency of the PP and tree methods can be determined from the data shown in Figures 3 and 4. The rate of CPU usage by the PP code on the CRAY X-MP, using the CFT compiler, was approximately $3.5\times10^{-7}\ N$ CPU seconds per particle per step, where the coefficient is independent of the form of the density distribution. For reasonable values of the clumping parameter, $\theta \gtrsim 0.3$, the tree method scales as $\alpha \log_{10} N + \beta$ CPU seconds per particle per step, where the values of α and β depend on θ. For example, if $\theta = 1$ and only monopole terms are retained in the force calculation, then $\alpha \approx 1.8\times10^{-4}$ and $\beta \approx 0$. The relative efficiency of the two methods then scales as direct/tree $\approx 1.9\times10^{-3}\ N/\log_{10} N$, and the two are comparable for $N \approx 1700$. If $\theta = 0.5$ and only monopole terms are retained, then $\alpha \approx 1.4\times10^{-3}$, $\beta \approx -2.7\times10^{-3}$, and the relative efficiency scales as direct/tree $\approx 1.3\times10^{-4}\ N/(0.5 \log_{10} N - 1)$, and the two are comparable for $N \approx 7000$.

The point at which the tree method is more efficient than the simple PP technique depends on the degree of clustering imposed upon the system. It appears that tree codes will not be useful for small-N systems, $N \lesssim 1000$, unless significant improvements can be made in the future. Similarly, tree algorithms will outperform the direct approach for $N \gtrsim 5000$, unless an unusually small value for θ is required. (Note that this comparison is not entirely fair, since more refined direct methods can also be vectorized [e.g., Aarseth and Inagaki 1986]. Thus the value of N given above for which the tree method is more efficient represents a lower limit in comparison with more sophisticated direct techniques.)

b) *Reduced N-Body, PM Methods*

Recent theoretical and observational results have indicated that interactions of galaxies with less massive companions could be responsible for a number of important evolutionary effects (for a review see Quinn 1986). The accretion of satellites can, in principle, significantly modify the distribution of matter in otherwise normal galaxies (e.g., Tremaine 1980; Toomre 1980; White 1983; Quinn and Goodman 1986). In particular, the simulations of Quinn and Goodman (1986) suggested that the decay of satellite orbits could lead to a heating of the disk and an increase in the vertical scale heights of spiral galaxies even for low-mass satellites. However, the results were not conclusive, since the restricted N-body approach used by Quinn and Goodman did not account for the self-gravity of the disk.

In order to demonstrate the suitability of the tree method for studying galactic systems, several simulations of the interaction of a disk with a less massive companion were performed for parameters similar to those assumed by Quinn and Goodman (1986). An example is given in Figures 11 and 12, which show the decay of a satellite initially on a direct, coplanar circular orbit around an exponential disk with $N = 32{,}768$. The disk model (which was kindly provided by S. Casertano) was of the form proposed by Bahcall and Soneira (1980). The total disk mass was $5.6\times10^{10}\ M_\odot$, with a radial scale length of 3.5 kpc and a vertical scale height of 350 pc. The disk was truncated at 6 radial scale lengths; i.e., $r_{\rm disk} = 21$ kpc. The surrounding isothermal halo was included as a rigid potential well with parameters such that the ratio of halo mass to disk mass within 14.7 kpc of the center was approximately unity (e.g., Bahcall and Casertano 1985). (The detailed structure of the halo is not important in determining the sinking rate of the satellite [Quinn and Goodman 1986].) The mass of the satellite was 4% of the disk, and the satellite radius was approximately 1 kpc. (The satellite-disk particle interaction was performed by a direct sum with the satellite modeled by a Plummer potential of scale length $\epsilon_{\rm sat} = 1$ kpc.)

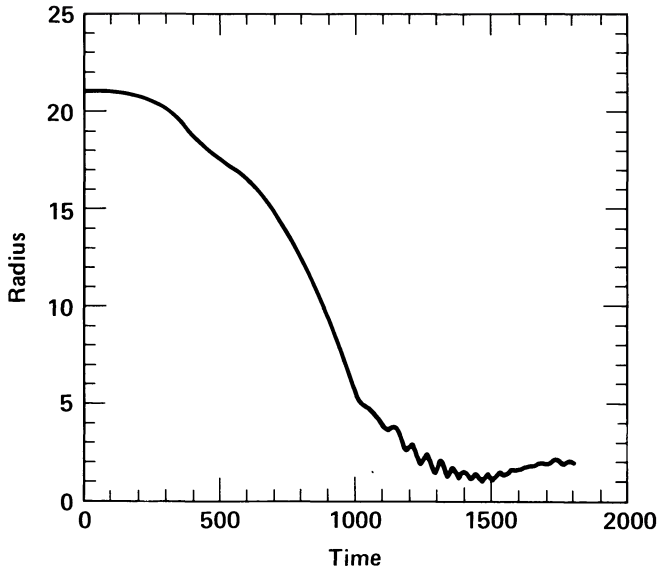

FIG. 12.—Radial separation of the satellite and disk centers of mass for the simulation shown in Fig. 11. Time is measured in units of 10^6 yr, and the radial separation is measured in kiloparsecs.

Time is measured in units of 10^6 yr, and each frame in Figure 11 is 60 kpc across. The tolerance parameter was chosen to be $\theta = 1$, and the simulation consisted of 300 time steps, each of duration 6×10^6 yr. Quadrupole terms were not included in the force computations. The softening parameter was equal to the mean interparticle separation evaluated at the half-mass radius of a flat, infinite exponential disk ($r_{1/2} \approx 1.7$ radial scale lengths) and at a z scale height thickness. (That is, $\epsilon = \lambda, \lambda = n^{-1/3}$, and $n = 0.5N/\pi r_{1/2}^2 2h_z$.) For the parameters discussed above, $\lambda \approx 0.2$ kpc.

During the course of the simulation the energy of the disk-satellite system was conserved to $\sim 0.1\%$, and l_z was conserved to $\sim 1\%$. The average number of terms per particle in the force evaluation fluctuated between 190 and 200. The entire simulation required approximately 2.0 CRAY X-MP CPU hours. (An isolated disk was evolved for 500 steps, each of duration 6×10^6 yr, with $\theta = 1.0$ to verify that the stability of the model was not affected by the numerical approximations. Energy was conserved to $\sim 0.5\%$, l_z was conserved to $\sim 0.01\%$, and the nonconservation of linear momentum resulted in a center-of-mass drift of 0.4% of the initial disk radius.)

The results shown in Figures 11 and 12 are in excellent agreement with the restricted n-body calculations of Quinn and Goodman (1986). The familiar leading cavity resulting from the loss of angular momentum by particles in front of

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

the satellite (see Quinn and Goodman 1986) is clearly visible at early times ($t \lesssim 1000$). A dramatic series of density waves is generated as the satellite nears the center of the disk ($t \sim 600$–1000), and persistent rings in the particle distribution can be seen at later times ($t \sim 1500$). Finally, by the end of the simulation, when the satellite has penetrated to the center of the disk ($t \sim 1800$), the disk has settled into an approximately axisymmetric equilibrium state. The satellite accretion has affected the radial distribution of particles but not the vertical scale height, as expected for a coplanar orbit.

The sinking rate, as inferred from Figure 12, is in good agreement with the Quinn and Goodman (1986) result. A comparison of Figure 12 can be made with their simulation of a 4% mass, 1 kpc radius satellite on a direct, coplanar orbit. The radial displacement of the satellite from the center of mass of the disk agrees to within $\sim 10\%$–15% for all times, until the satellite is within roughly 1 scale length of the center of the disk, at which point both calculations become rather noisy (owing to discreteness effects). The agreement between these two techniques suggests that the self-gravity of the disk particles may not be critical in determining the sinking rates of satellites. A complete study is currently underway to determine whether or not this is a general conclusion for all satellite parameters and to investigate the self-consistent response of the vertical structure of the disk to inclined satellite orbits (Hernquist and Quinn 1987).

The CPU time consumed by the tree code for the simulation shown in Figure 11 can be compared with that expected for a PM code. A total of 2.0 CRAY X-MP CPU hours were used for $N = 32{,}768$ particles and 300 time steps, corresponding to a rate of approximately 7.3×10^{-4} CPU seconds per particle per step. A similar calculation performed with the FFT code written by Jens Villumsen (Villumsen and Gunn 1986) for a $64 \times 64 \times 24$ grid required approximately 6×10^{-5} CPU seconds per particle per step. Thus the tree code with $\theta = 1$ is roughly 10 times slower than the most efficient PM codes. This ratio increases to ~ 50 if $\theta = 0.5$ is assumed. It should be noted, however, that the tree method may allow for for a larger dynamic range in spatial resolution at fixed CPU usage.

VI. CONCLUSIONS

This paper has presented a FORTRAN version of the Barnes-Hut hierarchical tree algorithm and analyzed its properties in the context of the gravitational N-body problem. The current implementation has been optimized for vectorizing supercomputers, resulting in a significant improvement in efficiency over the original Barnes-Hut code running on VAX-class machines. The tree method is considerably faster than the straightforward particle-particle (PP) technique, scaling as $O(N \log N)$ versus $O(N^2)$ for a wide range in the operating parameters of the code. The reduction in required CPU time is achieved at the expense of a small, controllable error in the force computation. The detailed structure of sufficiently distant clusters is neglected by performing center-of-mass expansions of their potentials. With the introduction of a tree data structure to organize the particles and the use of a simple opening-angle criterion to determine whether or not a given cluster should be subdivided, the approximations can be performed efficiently.

Additional refinements to the original Barnes-Hut code have included the additional of quadrupole terms to the expansion of the cluster potentials, resulting in an improvement in the accuracy of the force computation for fixed clumping parameter θ. In addition, the quadrupole version may be more efficient than the monopole code, at fixed accuracy, for some density distributions.

Considerations of the efficiency and accuracy of the method place tight constraints on the useful ranges of the code parameters. The requirement that the $N \log N$ scaling be maintained, and that the accuracy of the force calculation be tolerable, limits θ approximately to $0.3 \lesssim \theta \lesssim 1.0$. Furthermore, the violation of momentum conservation may limit θ to $\theta < 1$ for simulations with a large number of time steps unless the positions and velocities are renormalized at each step to preserve the center-of-mass properties. The tree method will not be useful for a small number of particles. For $N \lesssim$ a few thousand a direct sum over all pair interactions in a vectorized loop is faster. Finally, the error in the tree force computation, relative to a direct sum increases with the softening parameter, ε, since the expansion of the cluster potentials is accurate only for pointlike particles. The optimal range for ε appears to be $\varepsilon/\lambda \lesssim 1$–$2$, where λ is the mean interparticle separation.

Simple experiments indicate that the relaxation time is only weakly influenced by the approximations in the force computation. Since tree codes can, in principle, handle large N, they should be useful for studying collisionless systems. It remains an open question, however, whether or not systems which evolve as a result of subtle relaxation effects (such as globular clusters) can be modeled by the tree method.

The tree method offers several advantages over previous N-body techniques. In particular, tree codes will be more flexible than the PM and P^3M approaches, since they do not rely on fixed grids or place restrictions on the global geometry of the system to be studied. Due to the Lagrangian nature of tree algorithms, their dynamic range in spatial resolution is not limited, and individual nearby binary encounters can be modeled accurately. Furthermore, there will be no difficulties with fictitious images, as with FFT codes, or ambiguities resulting from particles exiting from a grid. Finally, the simplicity of the Barnes-Hut scheme suggests that a rigorous error analysis may be feasible for this approach.

The primary disadvantage of the tree method, at present, is that it is significantly slower than FFT codes (by a factor ~ 10). Thus, tree codes may not be advantageous in situations that can be treated with the PM technique. In addition, the pure expansion method is the most efficient N-body algorithm, and will usually be preferable for highly symmetric systems.

These disadvantages may be short-lived, however. Fourier techniques have been applied to N-body simulations for almost 20 years. It appears unlikely that further changes will lead to significant improvements in the efficiency of FFT routines. The tree method is relatively new and not yet fully developed. By analogy with PP algorithms, several refinements are possible that should improve the efficiency of tree codes considerably. These include the introduction of multiple time scales and the use of more sophisticated time integrators (e.g., Jernigan 1985). A partial vectorization of the tree

© American Astronomical Society • Provided by the NASA Astrophysics Data System

1987ApJS...64..715H

search would also be helpful. However, the future prospects appear brightest for the adaptation of tree codes to highly parallelized architectures. The tree method is better suited to parallel machines than other efficient N-body techniques. Given the rapid rise in interest in parallel computation, it is indeed possible that tree codes may represent the future of N-body simulations over the next decade.

I am especially grateful to Josh Barnes and Piet Hut for kindly providing me with a copy of their code and for their helpful advice throughout the course of this work. I would also like to thank Joss Bland, François Bouchet, Marc Davis, Sandy Faber, Claire Max, Dave Merritt, Peter Quinn, Jens Villumsen, Simon White, and Andrew Zachary for many enlightening discussions. In addition, Sverre Aarseth provided numerous valuable suggestions about the structure and content of this paper. This work was supported in part by a Calspace grant from the Theoretical Astrophysics Center at the University of California at Berkeley and under the auspices of the US Department of Energy by Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

REFERENCES

Aarseth, S. J. 1963, *M.N.R.A.S.*, **126**, 223.
_____. 1966, *M.N.R.A.S.*, **132**, 35.
_____. 1967, *Bull. Astr.*, **3**, 47.
_____. 1971, *Ap. Space Sci.*, **14**, 118.
_____. 1985, in *Multiple Time Scales*, ed. J. U. Brackill and B. I. Cohen (New York: Academic), p. 377.
Aarseth, S. J., and Inagaki, S. 1986, in *The Use of Supercomputers in Stellar Dynamics*, ed. P. Hut and S. McMillan (Berlin: Springer), p. 203.
Aarseth, S. J., and Lecar, M. 1975, *Ann. Rev. Astr. Ap.*, **13**, 1.
Ahmad, A., and Cohen, L. 1973, *J. Comput. Phys.*, **12**, 389.
Appel, A. W. 1981, Undergraduate thesis, Princeton University.
_____. 1985, *SIAM J. Sci. Stat. Comput.*, **6**, 85.
Bahcall, J. N. and Casertano, S. 1985, *Ap. J. (Letters)*, **293**, L7.
Bahcall, J. N., and Soneira, R. M. 1980, *Ap. J. Suppl.*, **44**, 73.
Barnes, J. 1986, in *The Use of Supercomputers in Stellar Dynamics*, ed. P. Hut and S. McMillan (Berlin: Springer), p. 175.
Barnes, J., Goodman, J., and Hut, P. 1986, *Ap. J.*, **300**, 112.
Barnes, J., and Hut, P. 1986, *Nature*, **324**, 446 (BH).
Birdsall, C. K., and Langdon, A. B. 1985, *Plasma Physics via Computer Simulation* (New York: McGraw-Hill).
Bouchet, F. R., Adam, J.-C., and Pellat, R. 1985, *Astr. Ap.*, **144**, 413.
Bouchet, F. R., and Kandrup, H. E. 1985, *Ap. J.*, **299**, 1.
Casertano, S., Hut, P., and McMillan, S. 1987, preprint.
Chandrasekhar, S. 1942, *Principles of Stellar Dynamics* (Chicago: University of Chicago Press).
Cooley, J. W., and Tukey, J. W. 1965, *Math. Comput.*, **19**, 297.
Davis, M., Efstathiou, G., Frenk, C. S., and White, S. D. M. 1985, *Ap. J.*, **292**, 371.
Eastwood, J. W., and Hockney, R. W. 1974, *J. Comput. Phys.*, **16**, 342.
Efstathiou, G., Davis, M., Frenk, C. S., and White, S. D. M. 1985, *Ap. J. Suppl.*, **57**, 241.
Gingold, R. A., and Monoghan, J. J. 1977, *M.N.R.A.S.*, **181**, 375.
_____. 1982, *J. Comput. Phys.*, **46**, 429.
Goldstein, H. 1980, *Classical Mechanics* (Reading: Addison-Wesley).
Greengard, L., and Rokhlin, V. 1986, Yale University preprint.
Hénon, M. 1964, *Ann. d'Ap.*, **27**, 83.
_____. 1973, in *Dynamical Structure and Evolution of Stellar Systems*, ed. L. Martinet and M. Mayor (Sauverny: Geneva Observatory), p. 183.
Hernquist, L., and Quinn, P. J. 1987, in preparation.
Hockney, R. W. 1967, *Ap. J.*, **150**, 797.
Hockney, R. W., and Eastwood, J. W. 1981, *Computer Simulation Using Particles* (New York: McGraw-Hill).
Hohl, F. 1973, *Ap. J.*, **184**, 353.
Hohl, F., and Hockney, R. W. 1969, *J. Comput. Phys.*, **4**, 306.
Jackson, J. D. 1975, *Classical Electrodynamics* (New York: Wiley).
James, R. A., and Weeks, T. 1986, in *The Use of Supercomputers in Stellar Dynamics*, ed. P. Hut and S. McMillan (Berlin: Springer), p. 125.
Jernigan, J. G. 1985, in *IAU Symposium 113, Dynamics of Star Clusters*, ed. J. Goodman and P. Hut (Dordrecht: Reidel), p. 275.
Lecar, M., and Cruz-Gonzalez, C. 1972, in *The Gravitational N-Body Problem*, ed. M. Lecar (Dordrecht: Reidel), p. 131.
Lucy, L. B. 1977, *A.J.*, **82**, 1013.
McGlynn, T. A. 1984, *Ap. J.*, **281**, 13.
Merritt, D., and Aguilar, L. A. 1985, *M.N.R.A.S.*, **217**, 787.
Piran, T., and Villumsen, J. V. 1986, in *IAU Symposium 127, Structure and Dynamics of Elliptical Galaxies*, ed. P. T. deZeeuw, in preparation.
Porter, D. 1985, Ph.D. thesis, University of California, Berkeley.
Press, W. H. 1986, in *The Use of Supercomputers in Stellar Dynamics*, ed. P. Hut and S. McMillan (Berlin: Springer), p. 184.
Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. 1986, *Numerical Recipes: The Art of Scientific Computing* (Cambridge: Cambridge University Press).
Quinn, P. J. 1982, Ph.D. thesis, Australian National University, Canberra.
_____. 1986, in *The Santa Cruz Workshop on Nearly Normal Galaxies*, in preparation.
Quinn, P. J., and Goodman, J. 1986, *Ap. J.*, **309**, 472.
Quinn, P. J., Salmon, J. K., and Zurek, W. H. 1986, in *The Use of Supercomputers in Stellar Dynamics*, ed. P. Hut and S. McMillan (Berlin: Springer), p. 130.
Smith, H. 1981, *M.N.R.A.S.*, **195**, 35.
Standish, E. M., and Aksnes, K. 1969, *Ap. J.*, **158**, 519.
Toomre, A. 1980, in *The Structure and Evolution of Normal Galaxies*, ed. S. M. Fall and D. Lynden-Bell (Cambridge: Cambridge University Press), p. 111.
Tremaine, S. 1980, in *The Structure and Evolution of Normal Galaxies*, ed. S. M. Fall and D. Lynden-Bell (Cambridge: Cambridge University Press), p. 67.
van Albada, T. S. 1982, *M.N.R.A.S.*, **201**, 939.
_____. 1986, in *The Use of Supercomputers in Stellar Dynamics*, ed. P. Hut and S. McMillan (Berlin: Springer), p. 23.
van Albada, T. S., and van Gorkum, J. H. 1977, *Astr. Ap.*, **54**, 121.
Villumsen, J. V. 1982, *M.N.R.A.S.*, **199**, 493.
Villumsen, J. V., and Gunn, J. E. 1986, preprint.
von Hoerner, S. 1960, *Zs. Ap.*, **50**, 184.
White, S. D. M. 1978, *M.N.R.A.S.*, **184**, 185.
_____. 1979, *M.N.R.A.S.*, **189**, 831.
_____. 1982, in *Morphology and Dynamics of Galaxies*, ed. L. Martinet and M. Mayor (Sauverny: Geneva Observatory).
_____. 1983, *Ap. J.*, **274**, 53.
Wielen, R. 1967, *Veröff. Astr. Rechen-Inst. Heidelberg*, No. 19.

LARS HERNQUIST: Department of Astronomy, University of California, Berkeley, CA 94720

© American Astronomical Society • Provided by the NASA Astrophysics Data System