Computer Physics Communications 48 (1988) 107–115
North-Holland, Amsterdam

HIERARCHICAL *N*-BODY METHODS

Lars HERNQUIST [1,2,3]

[1] *Dept. of Astronomy, Univ. of Calif., Berkeley, CA 94720, USA*
[2] *Inst. of Geophys. and Planetary Phys., LLNL, Livermore, CA 94550, USA*
[3] *Institute for Advanced Study, Princeton, NJ 08544, USA*

The hierarchical tree method offers the possibility of computing the interaction between N self-gravitating particles in a time $\sim \mathcal{O}(N \log N)$, without using a grid. The potential advantages of this technique for collisionless systems and for particle hydrodynamic schemes are discussed.

1. Introduction

N-body simulations have been used to study a wide variety of astrophysical systems during the past 25 years, ranging from small clusters of stars to galaxies and the formation of large-scale structure in the universe. (For a review see Hockney and Eastwood [1], van Albada [2], Sellwood [3].) Astrophysical applications place severe demands on numerical methods since large numbers of simulation particles are typically required. For example, galaxies are collisionless on the evolutionary time-scale of the universe; hence spurious relaxation effects must be avoided. Cosmological simulations must be able to handle density contrasts of order the ratio of the mass of a large cluster to that of an individual galaxy. Finally, an accurate modeling of the dynamical evolution of globular clusters requires that the number of simulation particles be equal to the number of stars comprising the cluster. In all three cases N must be $N \sim 10^5$–10^6.

The straightforward particle–particle (PP) technique, in which the potential is computed as a direct sum over all individual two-body interactions, has a number of advantages over other N-body methods. In particular, it is fully Lagrangian and does not use a grid. Thus, it offers a large dynamic range in spatial resolution and does not impose geometrical restrictions on the systems to be studied. However, it is not feasible to use the PP method for $N \sim 10^5$–10^6. A fully vectorized PP code uses of order $3 \times 10^{-7}\ N^2$ cpu s/step on a CRAY X-MP (e.g. Hernquist [4]). Hence, approximately one cpu year would be required to evolve systems with $N = 10^{5.5}$ for 1000 timesteps. It is possible to improve the efficiency of this technique using high order integrators and multiple time-scales (for a review see Aarseth [5]); nevertheless the computational expense remains prohibitive for large N (for a detailed analysis, see Makino and Hut [6]).

More efficient techniques have generally been obtained by compromising one or more of the features of the PP method. Typically this has involved imposing limitations on one or more of: the dynamic range in spatial resolution, the global geometry, the boundary conditions, and so forth (for a discussion see ref. [4]). In many situations the implied restrictions are not relevant. However, there exist large classes of problems for which these issues are crucial; e.g. systems with high density contrasts and/or large asymmetries. Hence there is strong motivation for an efficient, gridless Lagrangian method.

2. The hierarchical tree method

Recently, a new class of N-body algorithms has been proposed which retain many of the advantages of the PP technique (e.g. Appel [7,8],

0010-4655/88/$03.50 © Elsevier Science Publishers B.V.
(North-Holland Physics Publishing Division)

Jernigan [9], Porter [10], Barnes and Hut [11], Barnes [12], Greengard and Rokhlin [13], Greengard [14]). Rather than compromising the spatial resolution and/or imposing geometrical restrictions, these methods introduce approximations into the calculation of the potential. The force on an individual particle from nearly particles is, on average, computed as a direct sum. The influence of remote particles is included by performing multipole expansions of clusters or cells containing many particles, truncated at a fixed, relatively low, order. The force on a particle is then obtained from the expansions. Typically, the number of terms in an expansion is small compared with the number of particles in the corresponding cluster or cell, and a significant gain in efficiency is realized.

The process of computing the potential from a truncated multipole expansion is equivalent to neglecting the details of the distribution of particles within a given cluster or cell, to a specified level of accuracy. This procedure is physically well-motivated. For example, the dynamics of the Earth–Moon system is insensitive to the detailed mass distribution of each body. The approximation is also numerically well-motivated. Errors will always be present in N-body simulations from round-off, truncation, and discreteness effects. Given these, it is not consistent to compute the potential field to arbitrarily high precision.

The basic structure of the hierarchical algorithm can be summarized as follows. At each timestep the system is organized into a nested hierarchy of clusters or cells. For each particle, the force computation begins at the top of the hierarchy (i.e. at low spatial resolution) and the size of the current cluster or cell, s, is compared with the distance to the particle, d. If

$$s/d \leqslant \theta, \tag{1}$$

where θ is a fixed tolerance parameter [11], then the internal structure of the cluster or cell is ignored and the force on the particle is computed using the corresponding truncated expansion. Otherwise, the cluster or cell is subdivided into components and the criterion is recursively applied to each subunit.

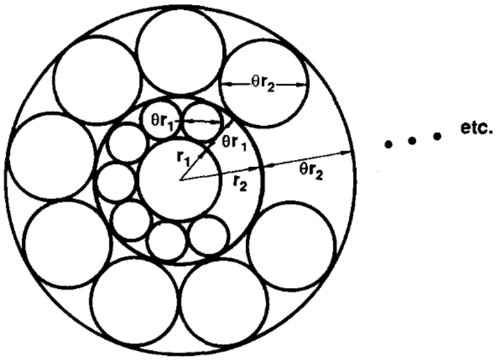


Fig. 1. Schematic representation of the force computation on a particle at the center of a homogeneous sphere of particles, using the hierarchical method. The influence of subregions of linear dimension θr_n, each containing many particles, is included through the use of multipole expansions.

For non-zero θ, a sum over N particles is replaced by a sum over log N interactions. To demonstrate this in an artificial situation, consider a single particle at the center of a homogeneous sphere. The remaining particles are organized into a hierarchy of clusters of cells. The total number of interactions can be estimated from the number of substructures contained within concentric shells surrounding the origin, as shown in fig. 1. The mass distribution within each shell will be unresolved since $s/d = r_i\theta/r_i = \theta$. Each shell will contain a number of subunits given roughly by

$$n^i_{\text{sub}} \sim \frac{4\pi r_i^3\theta}{\frac{4}{3}\pi r_i^3\theta^3/8} = \frac{24}{\theta^2}. \tag{2}$$

The total number of interactions will be $n_{\text{int}} \sim 24 n_{\text{sh}}/\theta^2 + n_0$, where n_{sh} is the total number of shells and n_0 is the number of direct interactions resulting from particles inside the sphere with radius r_1. Now, $r_i = (1+\theta)^{i-1} r_1$, implying $r_{n_{\text{sh}}}/r_1 = (1+\theta)^{n_{\text{sh}}-1}$. In addition, the system radius, R, is $R = r_{n_{\text{sh}}}(1+\theta)$. The radius r_1 is the radius at which clustering begins, i.e. $r_1 \sim n^{-1/3}/\theta$, where the number density n, is $n = N/\frac{4}{3}\pi R^3$. Taking $n_0 = N(r_1/R)^3$ and eliminating $r_{n_{\text{sh}}}/r_1$ gives

$$n_{\text{int}} \sim \frac{24}{\theta^2}\,\frac{\log\left[\theta(3N/4\pi)^{1/3}\right]}{\log(1+\theta)} + \frac{4\pi}{3\theta^3}. \tag{3}$$

(Note that in this case $\theta \geqslant (4\pi/3N)^{1/3}$.) Thus, the dominant behavior for $\theta > 0$ is $n_{\text{int}} \sim \log N/\theta^2$. The number of operations to compute the force on all N bodies will then scale as $\sim \mathcal{O}(N \log N)$.

Empirical tests for various density profiles (Hernquist [4], Hernquist and Quinn [15], Bouchet and Hernquist [16], Hernquist and Bouchet [17]) demonstrate that the $N \log N$ scaling is typically well obeyed for $\theta \gtrsim 0.3$. The principal dependence on θ, $n_{\text{int}} \sim \theta^{-2}$, appears to apply accurately in three dimensions to both periodic systems [16,17] and isolated systems (provided that N is sufficiently large that edge effects are negligible [4]). For highly flattened systems such as disk galaxies, the cpu costs increase somewhat more slowly than θ^{-2} for decreasing θ [15].

In principle, therefore, the hierarchical method offers a significant reduction in computing cost compared with the PP technique, at the expense of a small, controllable error in the force calculation. This, of course, presupposes that the organization of the particles and subsequent determination of interactions can be performed efficiently. A naive implementation might insist on sorting all particles with respect to distance from each particle during the force computation, which would require $> \mathcal{O}(N^2)$ operations. The potential difficulties can be overcome with the introduction of a tree data structure (e.g. refs. [7–9,11–14]). As a result the codes derived from these principles have become known as tree codes. The various schemes differ primarily in their detailed organization of the particle information, but share the same underlying physical principle – the use of multipole expansions to approximate the potential of a distant mass distribution.

3. Empirical tests

In order to investigate the properties of the hierarchical tree method in some detail a version of the Barnes and Hut [11] algorithm was optimized for supercomputers by Hernquist [4]. The Barnes–Hut method relies on a hierarchical subdivision of space into regular cubic cells. At each step, prior to the force evaluation, a tree data structure is built to store the hierarchy. Each node in the tree is associated with a cubic volume of space containing a given number of particles. Each volume is subdivided into eight subunits (in 3-D) of equal volume, which become the descendents of the original node in the tree structure. This process is continued recursively until each fundamental subcell contains either one or zero particles. Empty cells are not stored explicitly in the tree, thus the leaves represent volumes of space containing precisely one particle. As part of the tree-building procedure, the total mass, center of mass coordinates, and quadrupole moments are computed recursively for each cell. The force on a given particle is determined by walking through the tree, beginning at the top of the hierarchy (i.e. largest volume). The tolerance criterion (1) is applied to each node. If $s/d \leqslant \theta$, then the influence of all particles within the cell is computed as a single particle–cell interaction. Otherwise, the cell is subdivided by continuing the descent through the tree until either the tolerance criterion is satisfied or an elementary cell is reached. In this manner, all operations, including tree construction and force evaluation can be performed in $\mathcal{O}(N \log N)$ time.

3.1. Timing analysis

The timing tests performed by Hernquist [4] on a CRAY X-MP can be summarized as follows. The $N \log N$ scaling appears to be well obeyed for a variety of density profiles, if $\theta \gtrsim 0.2$–0.3. For smaller θ the force calculation asymptotically approaches the N^2 behavior characteristic of a direct sum. The fraction of time involved in initializing the tree structure is small, typically of order a few percent for $\theta < 1$. On a non-vectorizing machine such as a VAX the tree initialization is even less significant. This is an important advantage of the Barnes–Hut algorithm over other tree codes which organize the particles into a hierarchy of clusters. The tree algorithm is more efficient than a fully vectorized direct sum for $N \gtrsim$ a few thousand. (Of course the force calculation is less accurate; a point which will be discussed below.) The use of higher order terms in the multipole expansions is generally useful. For example, at the same level of accuracy in the force computation the algorithm is more efficient if quadrupole terms are included and a correspondingly larger θ is used, relative to the monopole version with smaller θ. (Note that the multipole expansions are essentially power series in the ratio s/d.) Finally, the efficiency of the hierarchical tree method is not compromised by highly inhomogeneous systems (see below).

3.2. Conservation properties

The error introduced into the force calculation as a result of the multipole approximation increases monotonically with θ. Typically, the error, relative to a direct sum, is $\lesssim 1\%$ for $\theta = 1$, increasing rapidly for larger θ. This suggests that $\theta = 1$ may be a practical upper limit. (In practice, $\theta \lesssim 0.7$–0.8 appears to be desirable in order to avoid spurious effects associated with nearby cells.)

The influence of the approximations on the dynamical evolution of representative systems is currently under investigation (e.g. refs. [15–18]). Crudely speaking, the Hamiltonian will now consist of three parts

$$H = H_{\mathrm{mf}} + H_{\mathrm{disc}} + H_{\mathrm{fluc}}. \tag{4}$$

The term H_{mf} describes the mean-field behavior and would comprise the entire Hamiltonian of a collisionless system. The factor H_{disc} accounts for departures from the mean-field limit due to discreteness effects, which may or may not be physical. Finally, the term H_{fluc} accounts for the fluctuations in the potential field as a result of the multipole approximation. Since H_{fluc} depends explicitly on time, the energy of the system will no longer be conserved even in the limit of a vanishingly small timestep. However, initial test cases [4,11,15–17] indicate that energy conservation is only weakly violated, in the sense that the departures from exact conservation due to H_{fluc} appear to be comparable to those typically tolerated from time integration errors.

The interaction of a particle with particles in a cell is not symmetric and, hence, momentum and angular momentum conservation are also violated. However, empirical tests [4,15–17] show that the departures from exact conservation are small for $\theta \lesssim 1$. Furthermore, it may be possible to restore momentum conservation by symmetrizing the force computation between cells (e.g. refs. [13,14]).

3.3. Statistical behavior of a system

Although the hierarchical method faithfully reproduces the global evolution of representative systems, the approximations in the force computation can lead to subtle changes which may affect statistical properties in some cases. For example, it is not clear that relaxation effects will always be handled correctly. Simple tests [4] show that the relaxation time is not strongly influenced by the force errors for $N < 10^5$, as long as $\theta \lesssim 1$. In this limit, it appears that $H_{\mathrm{fluc}} \ll H_{\mathrm{disc}}$. However, as $N \to \infty$, $H_{\mathrm{disc}} \to 0$ monotonically, while the dependence of H_{fluc} on N is uncertain. Tests using Plummer mass models suggest that H_{fluc} decreases slowly with N [4], but this may not be a general conclusion valid for all density profiles. This is a point of concern since H_{fluc} may become comparable to H_{disc} for sufficiently large N, unless H_{fluc} decreases at least as quickly as H_{disc} with increasing N. If H_{fluc} begins to dominate, then relaxation will proceed as a result of numerical errors rather than discreteness effects. It remains an open question, therefore, whether or not the hierarchical tree method will be suitable for systems such as globular clusters.

3.4. General remarks

The empirical tests [4,15–17] and detailed analysis [18] performed to date suggest that the hierarchical tree method will be essentially equivalent to a fast direct sum for a wide variety of applications. The lack of a grid and the efficiency of the technique indicate that tree codes will offer large dynamic ranges in mass and spatial resolution. Furthermore, the freedom from geometrical restrictions implies that this method will be well-suited for highly inhomogeneous systems with large density contrasts. The possibility of both small and large-N simulations suggests that this method will be useful for studying collisional as well as collisionless processes. Finally, the errors introduced through the multipole approximation are simply related to θ. For a simple geometrical tree structure as that underlying the Barnes–Hut algorithm, a rigorous error analysis is feasible [18].

The primary disadvantage of the hierarchical tree method is that in its present form it is slow in comparison with particle–mesh codes for situations in which a large dynamic range in spatial resolution is not required [4]. However, recent developments [19] demonstrate that a vectorization of the tree search is possible leading to a factor ≈ 3–5 improvement in efficiency. In addition, the cpu efficiency can be improved by symmetrizing the force calculation between cells. In the Fast Multiple Method [13,14,20] the force on each particle is computed by first computing cell–cell interactions. For log N cells this requires a time $\sim \mathcal{O}(\log^2 N)$. Once these have been determined the force on a single particle can be calculated in a time independent of N, resulting in an overall scaling $\sim \mathcal{O}(N)$.

4. Examples

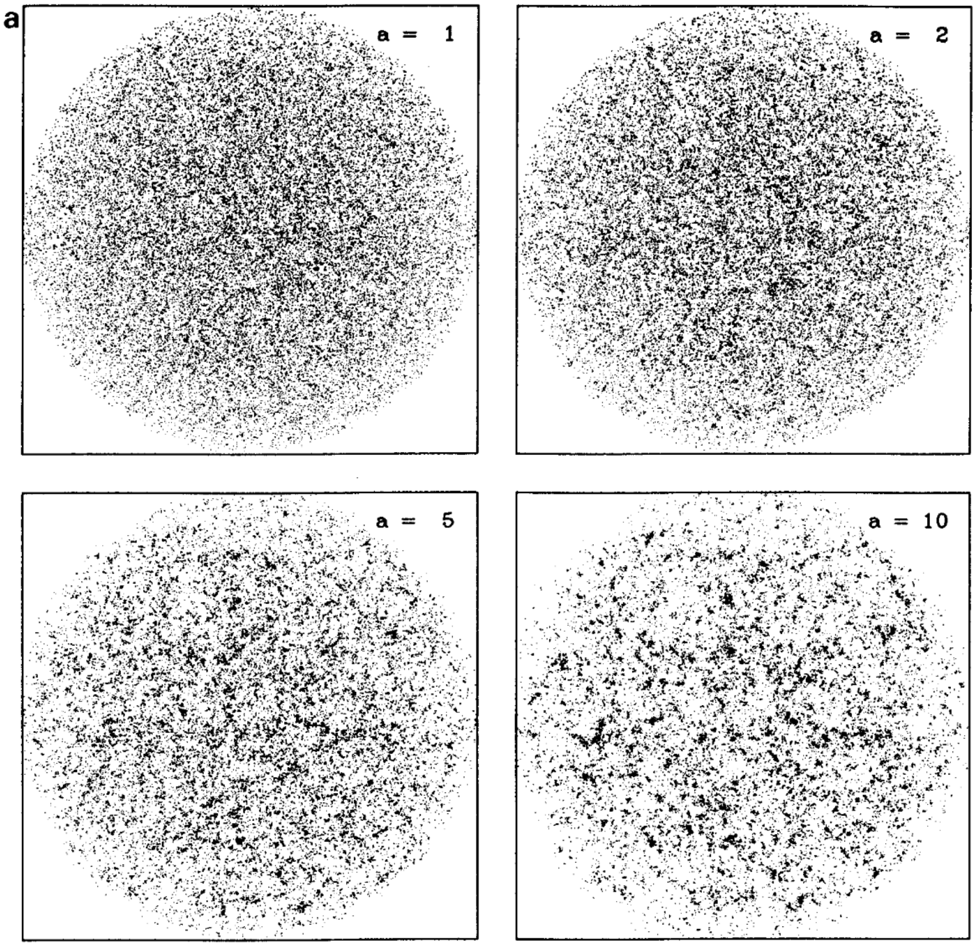
4.1. Cosmology

Previous cosmological studies (e.g. refs. [21,22], Marc Davis, this volume) have been successful in placing constraints on dark matter candidates and in relating conditions early in the universe to observed properties of galaxies and clusters of galaxies. Simulations of this nature place severe demands on numerical methods due to the large range in mass and spatial scales involved. The P^3M (particle–particle–particle–mesh) technique has been successful in overcoming many of these problems (e.g. ref. [23]). However, the performance of P^3M codes tends to degrade for highly clustered states, placing limitations on N and the degree of non-linearity which can be attained. The properties of the hierarchical tree method suggest that tree codes may be ideal tools for simulations of this nature.

A sample result is shown in figs. 2a, b for a sphere of particles, with $N = 32{,}768$, expanding into a vacuum. The particles were initially distributed randomly within the sphere (white noise) and the expansion velocity was that appropriate for a marginally bound universe ($\Omega = 1$). The expansion factor, a, which is the instantaneous radius of the system relative to that initially, is shown in all frames. The code parameters were $\theta = 1$, including up to quadrupole terms in the multipole expansions, with a two-body smoothing length of $\epsilon = 0.005$, corresponding to 0.1 times the initial mean interparticle separation. The units of length were such that the sphere had radius $R = 1$ at $a = 1$. The mean number of interactions per particle varied between ≈ 130–150 during the simulation. Energy and momentum were both well conserved. For cosmological simulations energy conservation is measured by comparing the change in the total energy, E, to the change in, for example, the potential energy, U. In this case, $\Delta E/\Delta U$ was $\approx 0.04\%$ after 30 expansion factors, and the maximum departure during the course of the simulation was $\Delta E/\Delta U \approx 0.1\%$. Non-conservation of momentum resulted in a drift in the center of mass coordinates of $\Delta\, r_{\mathrm{cm}}/a = 3.2 \times 10^{-4}$ at $a = 30.0$. Simulations in periodic, comoving coordinates give similar estimates of the conservation properties of the hierarchical tree method [16,17].

A useful statistical diagnostic of cosmological simulations is given by the two point correlation function $\xi(r)$, which is shown in fig. 3 for four different expansion factors. In this case, $\xi(r)$ was computed using only particles within a radius equal to 80% of the system radius to minimize edge effect. The amplitude and slope of $\xi(r)$ are both in good agreement with previous work (e.g. ref. [23]) if the results are scaled to the same system of units. In particular, $\xi(r)$ shows the self-similar evolution expected for white noise initial conditions with $\Omega = 1$. Finally, a more complete study [16,17] demonstrates that the fluctuations in $\xi(r)$ due to errors in the force computation are small compared with variations due to differential initial realizations. The errors in this case are, therefore, not statistically meaningful.

The cpu usage for this simulation in CRAY X-MP seconds per step, as a function of the expansion factor, is shown in fig. 4. Initial transient fluctuations are due to a resizing of the simulation volume represented by the tree structure as the system expands. Aside from these features the cpu time is essentially independent of a. That is, it is insensitive to the degree of clustering in the system. This striking conclusion is a result of the adaptive nature of the tree structure and the scale-free nature of the tolerance criterion (1).



Figs. 2a, b. Evolution of substructure in an expanding sphere of particles as a result of the gravitational instability, using the hierarchical tree method. The expansion factor, a, is indicated in each frame. The particle distributions are shown in a set of coordinates comoving with the expansion.

Thus, the hierarchical tree method offers a large dynamic range in spatial resolution without compromising efficiency, i.e. allowing a large dynamic range in mass scales. In addition, the flexibility of this method allows the influence of imposed boundary conditions on the formation of non-linear structures to be studied, including vacuum, quasi-periodic [16,17], and fully periodic boundary conditions [14].

4.2. Unified SPH / hierarchical N-body paradigm

The adaptive nature of the tree data structure suggests that it may be useful for performing functions other than computing global forces, such as those required by particle hydrodynamic methods. As summarized by Joe Monaghan (this volume) smoothed particle hydrodynamics models a fluid by a collection of particles. The particles, which represent a sampling of the fluid elements comprising the system, evolve according to the

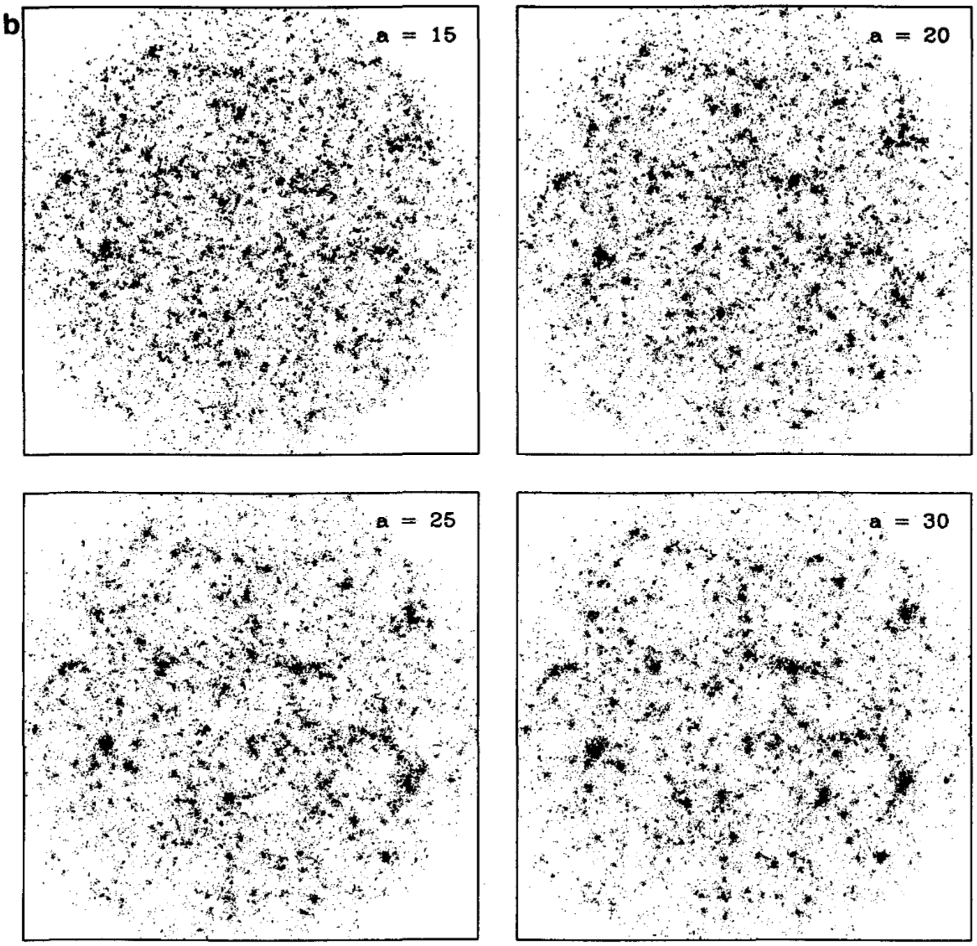


Fig. 2. Continued.

Lagrangian hydrodynamic conservation laws. The structure of SPH is, therefore, similar to that of an N-body simulation, except that additional terms are present in the equations of motion due to local hydrodynamic forces such as those resulting from pressure gradients and viscosity. The estimates of these quantities are obtained from the local properties of the fluid; i.e. from all particles within some specified distance of a given point. Thus, a fundamental requirement of SPH is that nearest neighbors must be found efficiently. This has usually been performed using grids and linked lists (e.g. ref. [24]). Grid searching is adequate for relatively homogeneous density distributions, but tends to degrade for clumpy systems (analogous to the difficulties encountered by the P^3M method). In addition, such a process introduces a length-scale into the problem (the cell width) which is usually dictated by memory constraints and not physical considerations. Thus this procedure will not be optimum if the search interval is small compared with the grid spacing.

Nearest neighbor searches in more than one dimension are an example of a more general problem known as range searching, which is an active area of research in computer science (for a general

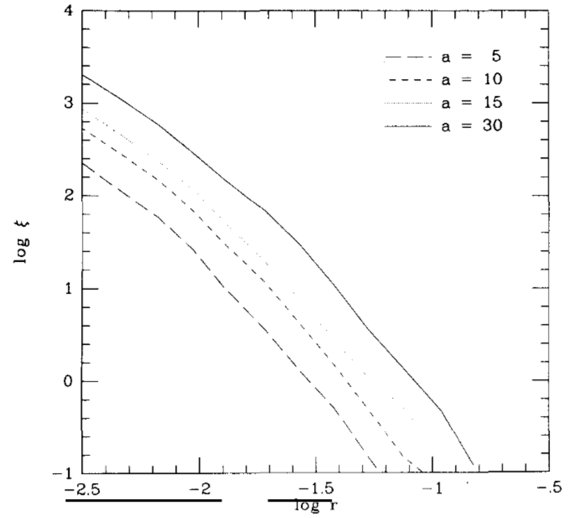


Fig. 3. Two point correlation function, ξ, as a function of the separation, r, for the simulation shown in figs. 2a, b.

discussion see Sedgewick [25]). In one dimension, range searching can be performed by first sorting the objects under consideration. Tree data structures effectively provide a means for performing multi-dimensional sorts, and it is not surprising that trees can be used to perform range searching. The optimal structures appear to be those with simple geometric interpretations, such as that used in the Barnes–Hut algorithm. Within this framework typical searches can be performed in $\mathcal{O}(\log N)$ time (see, e.g. refs. [26–28]).

There appears to be a deep correspondence between SPH and the hierarchical tree method. Both techniques are fully Lagrangian and do not require a grid. Tree codes are adaptive in nature since the tree structure is updated according to the phase space evolution of the system. This property is shared by SPH and is most clearly demonstrated by the possibility of a spatially variable smoothing length (note, however, that the implementation of this procedure is not without difficulties; e.g. Monaghan [24]). These considerations suggest that there may be a considerable advantage to combining the two methods. The tree data structure would be used to determine both global (gravitational) and local (hydrodynamic) forces, thereby eliminating the need for a grid and the introduction of an unphysical length scale into the problem. All operations could be performed in a time scaling as $\mathcal{O}(N \log N)$, with only a weak dependence on the degree of non-linearity in the system. Finally, much of the machinery developed for SPH, such as prescriptions for performing error analysis, could be carried over to the N-body problem. A unification of the two schemes should, therefore, allow for a more efficient exchange of ideas between the two fields.

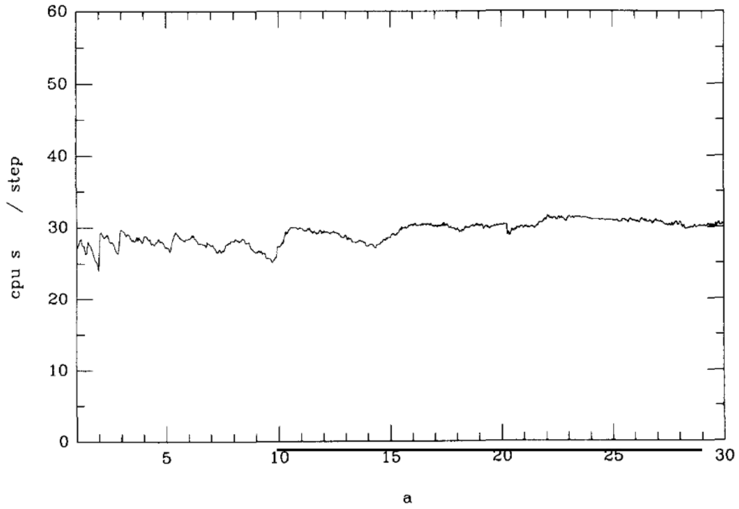


Fig. 4. Cpu time per step as a function of the expansion factor, a, for the simulation shown in figs. 2a, b.

Acknowledgements

I would like to thank Josh Barnes, Piet Hut and François Bouchet for many stimulating discussions about the hierarchical tree method. In addition, I am grateful to Leslie Greengard, Victor Rokhlin, and John Ambrosiano for describing the details of the Fast Multipole Method and to Joe Monaghan for useful advice about SPH. 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 No. W-7405-Eng-48.

References

[1] R.W. Hockney and J.W. Eastwood, Computer Simulation Using Particles (McGraw-Hill, New York, 1981).

[2] T.S. van Albada, in: The Use of Supercomputers in Stellar Dynamics, eds. P. Hut and S. McMillan (Springer-Verlag, Berlin, 1986) p. 23.

[3] J.A. Sellwood, (1987) preprint.

[4] L. Hernquist, Ap. J. Suppl. 64 (1987) 715.

[5] S.J. Aarseth, in: Multiple Time Scales, eds. J.U. Brackbill and B.I. Cohen (Academic Press, New York, 1985) p. 377.

[6] J. Makino and P. Hut, Institute for Advanced Study (1987) preprint.

[7] A.W. Appel, Undergraduate thesis, Princeton University (1981).

[8] A.W. Appel, SIAM J. Sci. Stat. Comput. 6 (1985) 85.

[9] J.G. Jernigan, IAU Symp. 113, Dynamics of Star Clusters, eds. J. Goodman and P. Hut (Reidel, Dordrecht, 1985) p. 275.

[10] D. Porter, Ph.D. thesis, University of California, Berkeley (1985).

[11] J. Barnes and P. Hut, Nature 324 (1986) 446.

[12] J. Barnes, in: The Use of Supercomputers in Stellar Dynamics, eds. P. Hut and S. McMillan (Springer-Verlag, Berlin, 1986) p. 175.

[13] L. Greengard and V. Rokhlin, Yale University preprint (1986).

[14] L. Greengard, Ph.D. thesis, Yale University (1987).

[15] L. Hernquist and P.J. Quinn (1987) in preparation.

[16] F.R. Bouchet and L. Hernquist, (1987) in preparation.

[17] L. Hernquist and F.R. Bouchet, (1987) in preparation.

[18] J. Barnes and P. Hut, (1987) in preparation.

[19] J. Makino and J. Barnes, Institute for Advanced Study (1987) preprint.

[20] J. Ambrosiano, L. Greengard and V. Rokhlin, Comput. Phys. Commun. 48 (1988) 117.

[21] M. Davis, G. Efstathiou, C.S. Frenk and S.D.M. White, Ap. J. 292 (1985) 371.

[22] P.J. Quinn, J.K. Salmon and W.H. Zurek, in: The Use of Supercomputers in Stellar Dynamics, eds. P. Hut and S. McMillan (Springer-Verlag, Berlin, 1986) p. 130.

[23] G. Efstathiou, M. Davis, C.S. Frenk and S.D.M. White, Ap. J. Suppl. 57 (1985) 241.

[24] J.J. Monaghan, Comput. Phys. Rep. 3 (1985) 71.

[25] R. Sedgewick, Algorithms (Addison-Wesley, Reading, 1983) p. 335.

[26] R.A. Finkel and J.L. Bentley, Acta Informatica 4 (1974) 1.

[27] J.L. Bentley and D.F. Stanat, Information Processing Lett. 3 (1975) 170.

[28] J.L. Bentley and J.H. Friedman, Comput. Surv. 11 (1979) 397.