N-body Methods on MIMD Supercomputers : Astrophysics on the Intel Touchstone Delta.

To appear in the Proceedings of the Fifth Australian Supercomputing Conference, 7-9 December, 1992

David P. Fullagar, Peter J. Quinn, Carl J. Grillmair*
Mount Stromlo and Siding Spring Observatories, Australian National University
John K. Salmon
Physics Department, California Institute of Technology
Michael S. Warren
Theoretical Astrophysics, Los Alamos National Laboratory

Abstract

We have used hierarchical N-body methods running on a MIMD parallel supercomputer to follow the evolution of systems with large numbers of particles ($N > 10^6$). The nonlinear gravitational collapse of galaxies from cosmological seed perturbations, the tidal stripping of globular clusters by galaxies and the collision of stellar systems have been studied in detail. The measured sustained performance of these methods for gravitational astrophysical problems exceeds 5 Gflops on the 512 node Intel Touchstone Delta[1].

1 Large N-body problems : the motivation

Traditionally, N-body simulations of collisionless systems (obeying the collisionless Boltzmann equation) have been run with the largest number of particles possible given the limitations on machine time and memory. This push to large N is mandated by the need to control the collisionality of the simulation over the desired number of dynamical times. Systems of particles interacting by Newtonian gravity can be considered collisionless over the time it takes individual particle–particle interactions to significant alter the motion of a given particle. This time scale (called the relaxation or "two–body" time scale, see [1]) is given approximately by

$$\tau_r = 0.1 \frac{N}{logN} \tau_{dyn} \qquad (1)$$

where τ_{dyn} is the time required for the particle to complete one orbit in the system. Galaxies consisting of 10^{11} stars are hundreds of dynamical times old. Globular clusters contain typically 10^6 stars and are 10^4 dynamical times old. Galaxies and to some extent globular clusters can be considered as collisionless systems since τ_r is larger than the age of the Universe. To represent a galaxy therefore requires at least as many particles such that τ_r is longer than the desired simulation time. In many cases this implies $N > 100,000$.

Other factors drive the need for large N in astrophysical modelling. The formation of galaxies and clusters of galaxies from small density perturbations at high redshift can be studied by following the non–linear gravitational collapse process using N-body techniques. In the cosmological setting we need to realize initial conditions consisting of small density fluctuations on a uniform background. In order to imprint a distinct signal at a given scale above the white noise background caused by particle discreteness, very large numbers of particles are usually called for, typically $N > 200,000$.

Recently it has been discovered that the departure of surface brightness contours of elliptical galaxies from perfect ellipses correlates strongly with important global and nuclear characteristics of the galaxy. Just why departures of a few percent from perfect ellipticity are related to global phenonema in ellipticals is unknown. N-body models with sufficient particles to study these small departures in detail ($N \sim O(400,000)$) have been conducted for a variety of formation scenarioes.

If one asks what level of discreteness is appropriate at the formation epoch of a single galaxy one should compare the Jeans mass near recombination ($10^6\,\mathrm{M}_\odot$) with the typical total mass of a large galaxy ($10^{12}\,\mathrm{M}_\odot$) which implies that at least 10^6 particles are needed. Indeed many stellar systems (globular cluster, dwarf galaxies) have 10^6 stars and large astronomical databases like the APM catalogue [2] of the local distribution of galaxies contain information on more than one million objects.

*Present Address: Lick Observatory, University of California at Santa Cruz.

[1]The Intel *Touchstone Delta* System is operated by Caltech on behalf of the Concurrent Supercomputing Consortium.

2 N-body methods

In an astrophysical N-body simulation, the phase-space density distribution is represented by a large collection of "bodies" (labelled by the indices i, j) which evolve in time according to the laws of Newtonian physics [3]:

$$\frac{d^2\vec{x}_i}{dt^2} = \sum_{j\neq i}^{N} \vec{a}_{ij} = \sum_{j\neq i} -\frac{Gm_j\vec{d}_{ij}}{|d_{ij}|^3}, \qquad \vec{d}_{ij} \equiv \vec{x}_i - \vec{x}_j \tag{2}$$

Direct implementation of the system of equations in Eqn. 2 is a trivial programming exercise. It is simply a double loop. It vectorizes well and it parallelizes easily and efficiently. It has been used for many years to study astrophysical systems. Unfortunately, it has an asymptotic time complexity of $O(N^2)$. Each of N left-hand-sides is a sum of $N-1$ right-hand-sides. The fact that the execution time scales as N^2 precludes the use of Eqn. 2 for values of N larger than a few tens of thousands, even on the fastest parallel supercomputers.

A number of *approximate* methods have been used which reduce the overall time, and allow simulation of systems with larger values of N. Methods employing an adaptive tree data structure have been popular in recent years because the resulting time complexity is $O(N\log N)$ or $O(N)$, and is relatively insensitive to the spatial distribution of particles [4, 5, 6]. N-body simulations which use adaptive tree data structures are referred to as *treecodes*.

The fundamental approximation employed by treecodes may be stated as:

$$\sum_j \frac{Gm_j\vec{d}_{ij}}{|d_{ij}|^3} \approx \frac{GM\vec{d}_{i,cm}}{d^3_{i,cm}} + \text{higher order multipoles}, \tag{3}$$

where $\vec{d}_{i,cm} = \vec{x}_i - \vec{x}_{cm}$ and $\vec{x}_{cm}$ is the centre-of-mass of the particles that appear under the summation on the left-hand side. In its simplest form, this approximation was known to Newton. The gravitational effect of an extended body like the Earth (consisting of some 3×10^{51} protons and neutrons) can be approximated by replacing the entire system by a point-mass located at the centre of mass. The resulting simplification is enormous: a factor of 10^{51} if one wishes to know the force on a falling apple.

The approximation of Eqn. 3 is not always valid. In fact, it doesn't converge at all if the point at which the force is desired is inside a sphere that circumscribes the other masses. Generally, the approximation is better if the measurement point is far from the other masses. The scale by which one should judge "near" and "far" is simply the diameter of the sphere that encloses the mass points.

At this point it is convenient to introduce a data structure that represents the distribution of matter on all length-scales. Suppose the particles are organized into a spatial, *oct-tree*. The root of the tree corresponds to a cube that encloses all the bodies in the simulation. Non-terminal nodes in the tree have up to eight daughters, corresponding to the eight sub-cubes that result from cutting the parent in half in each dimension. Any cell containing only one body is terminal. A two-dimensional example is shown in Fig. 1. Furthermore, suppose that each internal cell in the oct-tree contains the total mass of the bodies in it, their centre-of-mass and their quadrupole moments. Then it is possible to compute the force on any body by recursively traversing the tree according to the procedure outlined in Fig 2. Although it is not obvious from this brief description, the number of times Eqn. 3 is evaluated is during a traversal of the tree is proportional to $\log N$. Computing the forces on all N bodies requires traversing the tree N times and is thus $O(N\log N)$ in time.

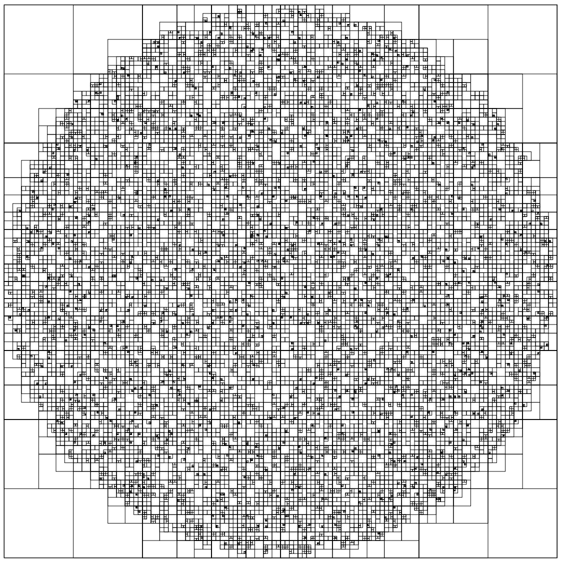

Figure 1: A two-dimensional tree with 10000 bodies uniformly distributed on the unit-disk.

3 Parallelism.

Astrophysical treecodes represent a formidable challenge for parallel computation. The difficulties stem from some fundamental properties of the problem:

- The distribution of bodies is highly non-uniform.
- The distribution of bodies is dynamic.
- The data structures are adaptive, and moderately complicated.
- Each body needs both global and local data for its update.

The non-uniformity of the data precludes use of a regular spatial decomposition. Instead, we

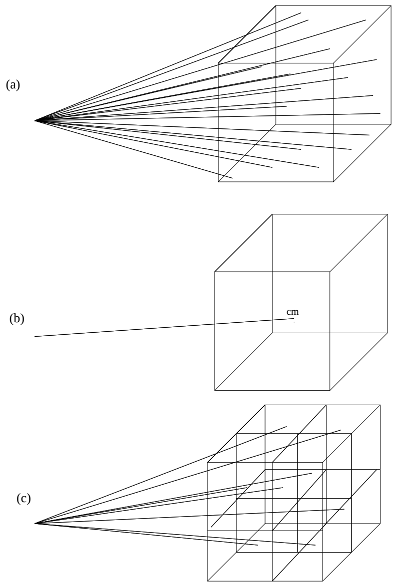

Figure 2: Schematic representation of the recursive structure of a treecode. (a) The exact force on a body is the result of a summation over all particles in a cell. (b) If the multipole approximation is valid, then the summation can be replaced by a single interaction. (c) Otherwise, the cell is subdivided into eight daughters, and the procedure, i.e., steps (b) and (c), is recursively applied to the daughters.

adopted the technique of *orthogonal recursive bisection*, ORB [7], whereby space is recursively divided in two, and half the processors are assigned to each domain until there is one processor associated with each rectangular domain. In order to avoid wasting time due to load imbalance and idle processors, it is crucial to divide space into domains with equal work-loads. The fact that the simulations are dynamic makes it impossible to precompute the decomposition or the communication pattern as would be the case with a static irregular problems.

Complex data structures present problems as well. Standard notations like Fortran90 or High Performance Fortran cannot readily represent a distributed adaptive tree. More importantly, exactly what one means by a "distributed adaptive tree" is not immediately obvious. It certainly depends on what one intends to do with the data structure. An answer comes from consideration of the last difficulty: the need for both global and local data in the update of each body.

Specifically, every body *sees* only a fraction of the complete tree. The distant parts are seen only at a coarse level of detail, while the nearby sections are seen all the way down to the leaves. The crucial observation is that nearby bodies *see* similar trees. In fact, if one considers all the bodies in an ORB domain, one can construct the union of all the trees they *see*. This union is called the *locally essential tree*. It is the data that will be required to compute the forces on every body in the domain. A locally essential tree, for a processor

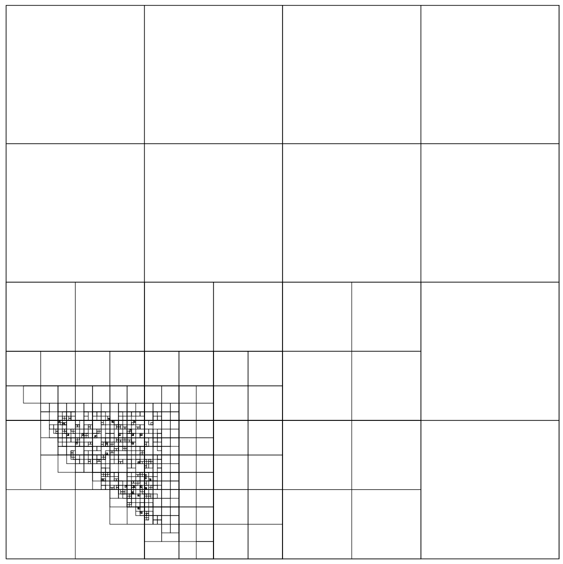

Figure 3: The locally essential tree for a processor whose domain is in the lower left corner of the system which gives rise to Fig. 1.

whose domain is the lower-left corner of Fig. 1 is shown in Fig. 3. A strategy is now clear, at least in principle:

- Obtain the locally essential tree in every processor.
- Proceed exactly as in the sequential case, traversing the tree once for each particle.

The fact that the second step is exactly the same as in the sequential case is significant. It constitutes the bulk of the time used by the algorithm, and it is useful to be able to write highly tuned sequential assembly language without regard to communication, synchronization or other parallel issues.

The problem now is to obtain the locally essential tree. The key is to use the recursive structure provided by the ORB. The result of ORB is a set of $\log_2 P$ spatial divisions and corresponding partitions of processors. Locally essential data can be routed to its correct location by looping over the ORB divisions, and repeatedly determining which local data is part of a locally essential tree on the *other side* of the cut. Every processor identifies its own exportable data, and then exchanges that data with a processor on the other side of the cut. After $\log_2 P$ exchanges, every processor is in possession of its locally essential tree.

The determination of which data to send is intimately related to how the data will be used. It is crucial that one can determine, *a priori*, which parts of a tree in local memory will also be needed in the evaluation of forces on an unspecified remote processor. Returning to the original algorithm, we find that this determination is, in fact, possible. Where the sequential algorithm asks "Is this multipole acceptable for the evaluation of the

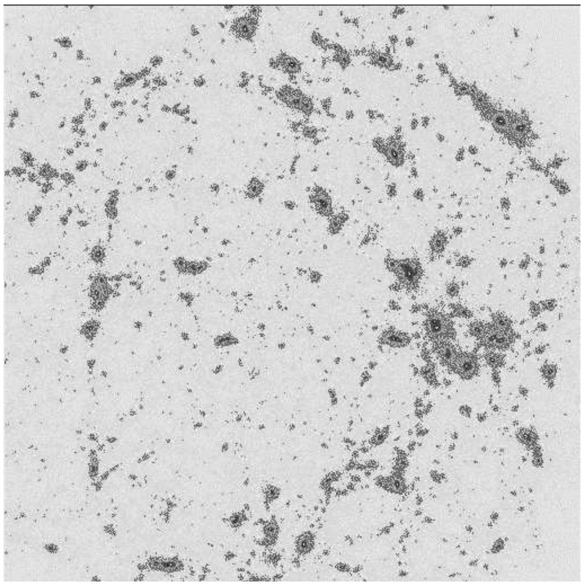

Figure 4: This image is a projection of about 5 million bodies from an 8.8 million particle cosmology simulation. The gray scale corresponds to the projected density along the line of sight. The length of the side of the box is approximately 7 Mpc (2×10^{20} km).

force at a particular location?", the parallel algorithm must also ask "Is this multipole acceptable for the evaluation of the force at any point in a rectangular parallelepiped?" The sequential algorithm prescribes that the first question is answered computing a ratio of distance to cell-size. We can answer the second by using the same criterion with a redefined distance, i.e., the distance is measured from the edge of the cell to the boundary of the rectangular domain.

4 Recent simulations.

Our parallel N-body code has been evolving for several years, and on several platforms. At press time, the fastest available platform is the Intel Touchstone Delta at Caltech, on which the simulations described below were run. We have also run significant simulations on an Intel ipsc/860, Ncube machines, and the Caltech/JPL Mark III [8, 9, 10, 11].

The statistics quoted below are based on internal diagnostics compiled by our program. Essentially, we keep track of the number of interactions computed. Each monopole interaction costs 29 flops, each quadrupole interaction costs 77 flops and each evaluation of the opening criterion costs 6 flops (using the Livermore Loops prescription of 1 sqrt = 4 flops, 1 division = 4 flops and 1 comparison = 1 flop). The flop rates follow from the interaction counts and the elapsed wall-clock time. The flop counts are identical to the best available sequential algorithm. We *do not* count flops associated with decomposition or other parallel constructs. The reported times are for the entire application, including I/O, communication, program initialization, etc. Overall, we see about 15% of the time spent in parallel overhead (communication, load imbalance, etc.), which corresponds to a parallel efficiency of about 87%.

4.1 Cosmological models.

In March 1992, we ran a simulation with 8,783,848 bodies on 512 processors for 780 timesteps. The simulation was of a spherical region of space 10 Mpc (Megaparsec) on a side; a region large enough to contain several hundred typical galaxies. An image from this run is shown in Fig. 4. Our simulation ran continuously for 16.7 hours, and carried out 3.24×10^{14} floating point operations, for a sustained rate of 5.4 Gflop/sec. Had we attempted the same calculation with a conventional $O(N^2)$ algorithm, it would have taken almost 3000 times as long to obtain an equivalent answer. We created 15 checkpoint files totalling 4.21 Gbytes. Had we saved all the potentially useful intermediate data, it would have required 273 Gbytes. A single checkpoint from this simulation exceeds 280 Mbytes. It is impractical to analyze the results on anything other than a parallel supercomputer. To that end, we have ported some of our analysis software to the Delta, and have isolated individual halos for further analysis. The 700 individual halos in this simulation have anywhere from 1,000 to 130,000 bodies, making them comparable in size to "state-of-the-art" isolated halo models running on Crays and other vector supercomputers. The Delta allowed us to evolve several hundred large halos simultaneously, in a realistic environment, providing us with much needed statistics, as well as information concerning environmental effects on evolution which cannot be obtained from isolated halo models.

In June 1992, in response to the recently announced measurement of the microwave background anisotropy by the COBE satellite [12, 13], we ran two large simulations of the Cold Dark Matter model of the Universe. The COBE measurement has constrained the last remaining free parameters left in this popular theory, and allows us to produce initial conditions in which the power spectrum of initial fluctuations is completely specified, including the overall amplitude. These are, by any measure, the largest N-body simulations ever run [2]. The results of the simulations are being analyzed. When compared with other observational constraints we hope they will provide compelling evidence either for or against the model

[2] A simulation with 17 million bodies has been reported [14] using a different approximation which suffers from limited spatial resolution. It ran for over 600 hours on an IBM vector supercomputer. In contrast, our simulation, which with marginally larger N, had twenty times the linear spatial resolution, and ran in one twentieth the time.

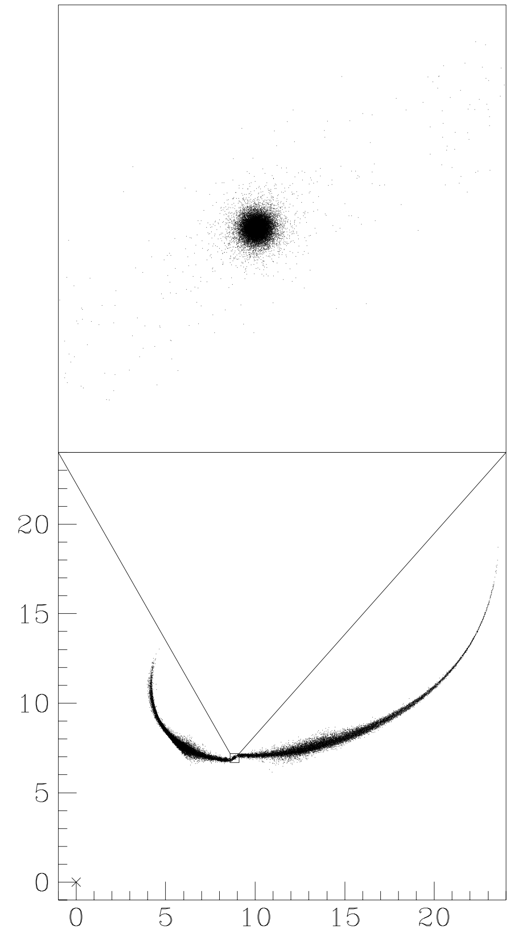

Figure 5: Two views of an evolved tidal system in the plane of the orbit. The bottom panel shows the large tidal tails extending away from the cluster after about two billion years of evolution. The side of the frame is 25 kpc and the centre of the modelled galaxy potential is marked with a cross at the origin. The top frame shows a enlarged view of the remaining cluster. The magnification factor is 50.

The simulations represented regions with diameter 250 Mpc and 100 Mpc, and had 17,158,608 and 17,154,598 bodies, respectively. The individual particles each represented about $3.3 \times 10^{10}\,M_\odot$ and $2.0 \times 10^9\,M_\odot$, respectively, so that galaxy-size halos are expected to form with tens to thousands of individual particles; enough to obtain reasonable statistics concerning the distributions and correlations of sizes. The spatial resolution was 20 kpc in both cases. They ran for 597 and 667 timesteps, in 23.5 and 28.6 hours, respectively, and wrote 21 and 27 data files for a total of 11.53 and 13.72 Gbytes. They respectively performed 4.33×10^{14} and 5.32×10^{14} floating point operations, sustaining rates of 5.2 Gflop/sec and 5.1 Gflop/sec.

4.2 Tidal stripping.

In January 1992, we began running a series of simulations to study the effects of a tidal field (as would be induced by a large galaxy) on the evolution of globular clusters [15]. Globular clusters are the oldest known stellar systems with ages in excess of 10 billion years. As these ancient stellar systems orbit a galaxy, they are stripped of stars by the tidal field. Using N-body models we can study the dependence of the stripping on the properties of the clusters, their orbital motion and the galactic tidal field parameters. By relating models to observations we can then establish the mass and density distribution of a given parent galaxy in the region probed by the cluster orbits (typically well outside the luminous body of the galaxy). We can also decern the orbital parameters of the clusters and hence constrain their formation mechanisms. The difficulty with such a study is the great discrepancy between the size of a globular cluster and it's characteristic orbital radius around the galaxy. Clusters have radii of order 10 parsecs and masses of order $10^6\,M_\odot$ and orbit parent galaxies with masses in the range 10^{11}–$10^{12}\,M_\odot$ at radii of 10,000 to 100,000 parsecs. Cluster orbital periods around the galaxy are typically 10,000 times longer than the time it takes stars inside the cluster to cross the cluster diameter. We therefore have to evolve the system for 10,000 to 100,000 dynamical times in order to study the physical situation accurately. An example of the tidal features in one of the models is shown in Fig. 5.

Each of the simulations is of modest size (only 65536 bodies), but must run for a very large number of timesteps. Because of the small size of the individual simulations, it is advantageous to run them on small meshes (16 to 64 processors), and to gain additional parallelism by running several simultaneously. This series is still underway. It has performed about 5.8×10^{14} flops, and created about 3600 data files with a total size of 7.5 Gbytes. When eight simulations run simultaneously on 256 processors, we see an aggregate speed of about 1.8 Gflop/sec. The globular cluster simulations could have been run on a workstation however, a 10 Mflop/sec workstation running 24 hours per day would accomplish the same task in about two years. No other system would have allowed us to run *five* simulations in such a short time.

4.3 The shapes of elliptical galaxies.

The universe contains two major classifications of galaxies - spirals and ellipticals. The spiral galaxies, like our own Milky Way, contain a large fraction of their stars in a flat disk which often also exhibits complex spiral structure of stars, gas, and dust. On the other hand, the elliptical galaxies are at first sight featureless objects, which as

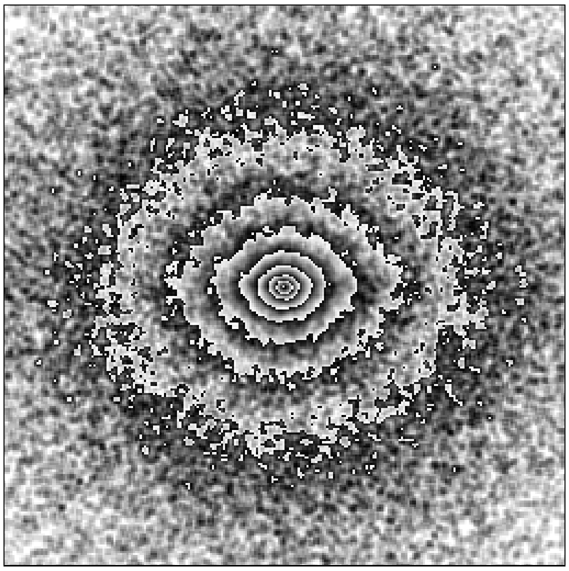

Figure 6: A remnant of a 10% merger, viewed in the plane of the off-centre encounter. The frame shows the *disky* isophotes of the total model material. The side of the box is five half-mass radii of the initial primary.

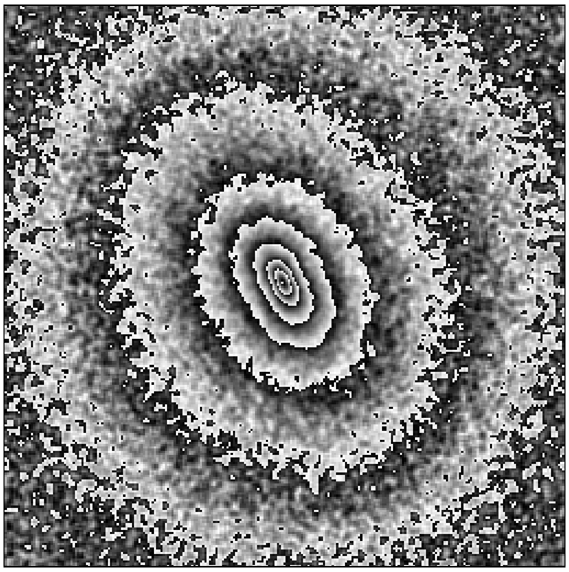

Figure 7: A remnant of a multiple hit merger of same mass models. This viewing angle shows *boxy* isophotes in the inner regions. The box side is five initial half-mass radii.

their name suggests are viewed as ellipsoids projected on the sky. However with recent advances in detector technology it has been observed that there exist deviations from perfect ellipticity at a few percent level in many ellipticals. If a fourier decomposition of the contours of constant surface brightness (isophotes) is made it is found that the coefficient of the $\cos(4\theta)$ term (conventionally named a_4) often accounts for a majority of the deviations. If the a_4 values of isophotes are negative the galaxy appears *boxy*, and if the a_4 values are positive the galaxy appears *disky* or diamond shaped. What is very interesting to studies of elliptical galaxy formation is that there are strong correlations between the a_4 shape parameter and other observed features [16]. Some of these include

1. Optical luminosity - *Boxy* ellipticals tend to be more luminous than *disky* galaxies.

2. Kinematics - Generally *boxy* galaxies are found to be flattened by anisotropic velocity dispersion, while *disky* galaxies are rotationally supported. The dimensionless spin parameter $\lambda = \frac{J|E|^{1/2}}{GM^{5/2}}$ for *boxy* galaxies is typically around 0.06, the value found in cosmological simulations due to tidal torquing. In disky ellipticals the value is in the range 0.1 to 0.4, closer to the measured value in spiral galaxies (about 0.5).

3. Radio / X-ray emission - *Boxy* ellipticals are frequently radio-loud, suggesting a central active region. They also show strong x-ray emission, thought to be associated with gaseous haloes. All cooling flow galaxies with measured a_4/a are *boxy* [17]. However *disky* ellipticals are usually undetected in both the radio and x-ray bands.

4. Cores - In a recent study by Nieto *et al.* [18] it was found that *boxy* and irregular ellipticals were the only galaxies with flat central profiles with all *disky* ellipticals having a peaked central profiles (i.e. they are unresolved).

So what is causing these correlations of the shape with both global and nuclear properties of the galaxy?

In order to determine possible formation scenarioes for the generation of *boxy* and *disky* ellipticals we have begun a study with two N-body approaches [19]. First, we examine the dark matter haloes which form in the cosmological models described in section 4.1. Fig. 4 shows the variety of structures produced in these simulations. By determining the shapes of the galaxy haloes, we can probe the relation between these sites for luminous galaxy formation and the resulting galaxies observed in the universe. Second, we are conducting a series of encounter experiments between galaxies that have already formed. A range of initial models and mass ratioes are evolved to see if late-time mergers can account for the differing shapes seen, and perhaps explain some of the correlations outlined above. Initial results suggest that mergers can produced both *boxy* and *disky* remnants - often also dependent on the viewing angle. Fig. 6 and Fig. 7 show examples of both classes.

In both these types of numerical experiment it is crucial to have large numbers of particles to model the galaxies. The isophote deviations are only at a few percent level and need to be followed to large radii from the centre of the galaxy. We have completed *five* simulations on the Delta during the first half of 1992, each with $N \sim O(400,000)$.

5 Conclusions and the future

Although the current production code is extremely sucessful[3] a new code is being developed to allow the incorporation of new "physics" and new numerical techniques.

The new treecode algorithm has been running for several months and is nearly ready for production The new *hashed oct-tree* (HOT) algorithm represents a major advance in style and sophistication. One of the limitations of the current production code is the complexity involved in determining locally essential data. The use of independent timestep integration methods which advance particles according to the local timescale is problematical with the old code. HOT can accommodate independent timestep integration schemes as well as an important new class of cell opening criteria [22] for which it is difficult to predict, a *priori*, which cells are required in a rectangular domain. HOT will also support $O(N)$ methods [6], which involve interactions between cells, in addition to the cell-body and body-body interactions which prevail in the $O(N \log N)$ methods we have used until now. The critical value of N at which $O(N)$ methods outperform $O(N \log N)$ methods is far from clear because the "constants" depend so strongly on both the specific problem and the most mundane of implementation details. However, by extending the error-estimates in [22, 23] to describe the interactions present in $O(N)$ algorithms, we believe that superior performance can be achieved for realistic values of N.

With the new code we plan to add gas dynamics via smooth particle hydrodynamics (SPH, see [24]) to the gravitational calculations. This will allow the extension of the projects described above and open possibilities for other studies.

Treecodes are by no means restricted to astrophysics. Many of the data generation and synthesis tasks associated with our simulation data can be expressed using hierarchical data structures, albeit not with gravitational interactions. Molecular dynamics simulations with charged or polar species (e.g., ions, water) are limited by the same $O(N^2)$ behavior as our gravitational simulations. Computational fluid dynamics using the *vortex method* requires solution of an N-body problem with a force law that is similar to that in gravity. Reaction-diffusion equations can be transformed into a computational problem with a similar structure. Even further afield are uses of spatial tree data structures in data compression, image processing, visualization and database searching. Our HOT implementation is far more flexible than the current production code. We believe it will not only be adaptable enough to satisfy our own needs for data analysis tasks, but will also enable us to collaborate with scientists in other disciplines to investigate broader uses of parallel treecodes.

[3] In May 1992, John Salmon and Mike Warren won the inaugural Intel Grand Challenges Award [20] and are also finalists in the 1992 Gordon Bell prize competition [21]

Acknowledgments

We thank the Concurrent Supercomputing Consortium, the Caltech Concurrent Supercomputing Facilities, the Australian National University Supercomputing Facility, and the Centre for Information Science Research (Australian National University) for providing computational resources. We acknowledge support from the Advanced Computing Division of the National Science Foundation, the Center for Research on Parallel Computing, the Institute of Geophysics and Planetary Physics and the Air Force Office of Scientific Research in the USA, and the Department of Industry Technology and Commerce, and the Queen Elizabeth II Silver Jubilee Trust in Australia.

References

[1] J. Binney and S. Tremaine, *Galactic Dynamics*. Princeton: Princeton University Press, 1987.

[2] S. J. Maddox, W. J. Sutherland, G. Efstathiou, and J. Loveday, "The APM galaxy survey - I. APM measurements and star-galaxy separation," *Monthly Notices of the Royal Astronomical Society*, vol. 243, pp. 692–712, 1990.

[3] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*. New York: Mcgraw-Hill International, 1981.

[4] A. W. Appel, "An efficient program for many-body simulation," *SIAM Journal of Scientific and Statistical Computing*, vol. 6, p. 85, 1985.

[5] J. E. Barnes and P. Hut, "A hierarchical O(NlogN) force-calculation algorithm," *Nature*, vol. 324, pp. 446–449, 1986.

[6] L. Greengard and V. I. Rokhlin, "A fast algorithm for particle simulations," *Journal of Computational Physics*, vol. 73, pp. 325–348, 1987.

[7] G. C. Fox, M. A. Johnson, G. A. Lyzenga, S. W. Otto, J. K. Salmon, and D. W. Walker, *Solving*

Problems on Concurrent Processors. Englewood Cliffs, NJ: Prentice Hall, 1988.

[8] J. K. Salmon, P. J. Quinn, and M. S. Warren, "Using parallel computers for very large N-body simulations: Shell formation using 180k particles," in *Heidelberg Conference on Dynamics and Interactions of Galaxies* (R. Wielen, ed.), (New York), pp. 216–218, Springer-Verlag, 1990.

[9] M. S. Warren, P. J. Quinn, J. K. Salmon, and W. H. Zurek, "Dark haloes formed via dissipationless collapse: I. shapes and alignment of angular momentum," *Astrophysical Journal*, 1992. (accepted for publication).

[10] M. S. Warren and J. K. Salmon, "A parallel treecode for gravitational N-body simulations with up to 20 million particles," *Bulletin of the American Astronomical Society*, vol. 23, no. 4, p. 1345, 1991.

[11] M. S. Warren, W. H. Zurek, P. J. Quinn, and J. K. Salmon, "The shape of the invisible halo: N-body simulations on parallel supercomputers," in *After the First Three Minutes Workshop Proceedings* (S. Holt, V. Trimble, and C. Bennett, eds.), (New York), AIP Press, 1991.

[12] C. L. Bennett *et al.*, "Preliminary separation of galactic and cosmic microwave emissions for the COBE DMR," COBE preprint 92-05, Goddard Space Flight Center, 1992.

[13] G. F. Smoot *et al.*, "Structure in the COBE DMR first year maps," COBE preprint 92-04, Goddard Space Flight Center, 1992.

[14] E. Bertschinger and J. M. Gelb, "Large cosmological N-body simulations," *Computers in Physics*, vol. 5, no. 2, p. 164, 1991.

[15] C. Grillmair, *Dynamics of Globular Cluster Systems.* PhD thesis, Australia National University, 1992.

[16] R. Bender, "New clues to structure and formation of elliptical galaxies," in *Heidelberg Conference on Dynamics and Interactions of Galaxies* (R. Wielen, ed.), (New York), pp. 232–243, Springer-Verlag, 1990.

[17] D. A. Forbes, "On the external origin of dust in elliptical galaxies," *Monthly Notices of the Royal Astronomical Society*, vol. 249, pp. 779–787, 1991.

[18] J.-L. Nieto, R. Bender, and P. Surma, "Central brightness profiles and isophotal shapes in elliptical galaxies," *Astronomy and Astrophysics*, vol. 244, p. L37, 1991.

[19] D. P. Fullagar, P. J. Quinn, and J. K. Salmon, "N-body simulations of a_4 isophote deviations in elliptical galaxies," in *Proceedings of ESO/EIPC Workshop on the Structure, Dynamics and Chemical Evolution of Early-Type Galaxies* (I. J. Danziger, ed.), (Munich), European Southern Observatory, 1992.

[20] J. K. Salmon and M. S. Warren, "Astrophysical N-body simulations on the delta," 1992. (winning entry in 1992 Intel Grand Challenge Award).

[21] M. S. Warren and J. K. Salmon, "Astrophysical N-body simulations at 5.4 gflop/sec," 1992. (entry in 1992 Gordon Bell Prize competition (finalist)).

[22] J. K. Salmon and M. S. Warren, "Skeletons from the treecode closet," *Journal of Computational Physics*, 1992. (submitted).

[23] J. K. Salmon, *Parallel Hierarchical N-Body Methods.* PhD thesis, California Institute of Technology, 1990.

[24] L. Hernquist and N. Katz, "TREESPH - a unification of SPH with the hierarchical tree method," *Astrophysical Journal Suppl.*, vol. 70, no. 2, p. 419, 1989.