Concurrent Supercomputing Consortium

CCSF-28-92 October 1992

Skeletons from the Treecode Closet

John K. Salmon and Michael S. Warren

ABSTRACT

Conventional N-body treecodes are susceptible to surprisingly large systematic errors which are capable of disrupting a small secondary galaxy in a two-galaxy encounter simulation. The errors arise because under certain geometrical circumstances, the conventional multipole acceptability criterion (MAC) with accuracy parameter $\theta \geq 0.577$ accepts multipole approximations with potentially unbounded errors. We present two alternative MACs, and we compare them on the basis of a strict analytic bound on the maximum allowed relative error in the acceleration. Although the new MACs eliminate a troublesome source of error, they still require remarkably small values of θ and/or high-order multipoles to *guarantee* reasonable accuracy. We propose a third MAC based on a sum of absolute errors, rather that individual relative errors. It is justified analytically and tested empirically. We find that the new MAC with strictly monopole interactions outperforms all three conventional MACs using quadrupoles.

Mailing Address: CCSF Techpubs, California Institute of Technology, Mail Code 158-79, Pasadena, CA 91125.
Email: techpubs@ccsf.caltech.edu. Phone: (818) 356-3907.

Skeletons from the Treecode Closet

John K. Salmon
California Institute of Technology, Mail Code 206-49, Pasadena, California 91125

Michael S. Warren [1]
Theoretical Astrophysics, Los Alamos National Laboratory, Los Alamos, NM 87545

Abstract

Conventional N-body treecodes are susceptible to surprisingly large systematic errors which are capable of disrupting a small secondary galaxy in a two-galaxy encounter simulation. The errors arise because under certain geometrical circumstances, the conventional multipole acceptability criterion (MAC) with accuracy parameter $\theta \gtrsim 0.577$ accepts multipole approximations with potentially unbounded errors. We present two alternative MACs, and we compare them on the basis of a strict analytic bound on the maximum allowed relative error in the acceleration. Although the new MACs eliminate a troublesome source of error, they still require remarkably small values of θ and/or high-order multipoles to *guarantee* reasonable accuracy. We propose a third MAC based on a sum of absolute errors, rather than individual relative errors. It is justified analytically and tested empirically. We find that the new MAC with strictly monopole interactions outperforms all three conventional MACs using quadrupoles.

Subject headings: treecodes — fast multipole methods — N-body methods

Submitted to: *Jourrnal of Computational Physics*, July 31, 1992

[1]Department of Physics, University of California, Santa Barbara

1 Introduction

Integration of the motion of N independent bodies under their mutual gravitational interaction has been an important tool in computational astrophysics for many years. Until recently, it was believed that such integrations required time proportional to N^2, because the long-range nature of the gravitational interaction requires that one consider the contribution of all $N-1$ other bodies to the force on each body. Algorithms which use a hierarchical data structure and an approximate force-law for aggregates of bodies were introduced independently by Appel[1], Jernigan and Porter[2] and Barnes and Hut[3]. Subsequently, a number of other authors have expanded on the theme of multipole approximations and hierarchical data structures. [4, 5, 6, 7, 8]. The computer programs described by these authors are known collectively as "treecodes" because the underlying data structure in all cases is a tree. These programs typically can compute the forces on all N bodies in a system in time proportional to $N \log N$ or even N. This qualitative change in the time-complexity of the algorithms has led to simulations much larger than were possible with direct-summation algorithms [9].

2 Overview of Treecodes.

Obviously, the detailed structure of the tree, and how it is constructed is of great importance in designing treecodes. Nevertheless, except for terminology, we shall not concern ourselves with tree-construction in this paper, as it appears to constitute a relatively small fraction of the total expense in using treecodes. Tree construction has been extensively covered in the literature [3, 10, 11, 7].

In this paper we shall assume that trees are made up of *cells*. A cell represents a bounded region of space, and contains some aggregate information about the bodies that lie within that region. Often, a cell is a cubical volume [3], but it need not be [7]. Similarly, the aggregate data is often a multipole expansion [3], but again, it need not be [8]. Cells also contain information (e.g., pointers) that can lead a program to *daughter cells* which represent smaller, disjoint subspaces which make up the parent. In this way, cells are organized into a tree. Some cells are *terminal*, which means that they do not contain daughter cells. Instead, they contain pointers to the bodies that lie within the region bounded by the cell. Terminal cells are often defined to have exactly one body [3], but it is sometimes desirable to construct trees whose terminal cells contain several bodies [11].

Treecodes may be broadly classified according to the most complicated type of *interaction* that is explicitly evaluated by the implementation. *Body-Cell* treecodes [3, 7, 2] compute interactions between individual bodies and cells in a hierarchical tree. These interactions are essentially the evaluation of the far field of a multipole expansion. Typically, $O(N \log N)$ interactions must be evaluated to find the accelerations on N bodies. Conversely, *Cell-Cell* treecodes [1, 5, 6, 8] rely on interactions (sometimes called "translations" in the literature) between pairs of cells, both of which contain numerous bodies. Typically, these interactions are much more complicated than the corresponding Body-Cell interactions, but the number of them is bounded by $O(N)$. The tradeoffs between accuracy, performance, and the constants of proportionality that are hidden by the "big-O" notation are still unclear. Owing to their widespread acceptance in the astrophysics community, we shall concentrate on Body-Cell treecodes in the remainder of this paper.

Body-Cell treecodes all share a common control structure. Each body is treated independently, without reference to the interactions of nearby bodies. Whenever the force on a body is required, the tree is traversed, starting at the root. Whenever a cell is visited, decisions are made to determine how to proceed. If the cell is terminal then direct Body-Body interactions with each of the bodies within the cell are computed, and the traversal terminates. Otherwise, we ask whether the multipole expansion stored in the cell will provide an adequate approximation for the effect of the contents of the cell on the body. This test is the "Multipole Acceptability Criterion", (MAC). The test is based on geometric considerations about the size of the cell, the location of the body and, perhaps, the contents of the cell. If the MAC passes, then a Body-Cell interaction is computed and the traversal (down this branch of the tree) terminates. Otherwise, each of the children of the cell is traversed in turn. This formulation of the tree traversal is naturally recursive. It maps immediately into a computer program in a language that supports recursion, e.g., C, Lisp. Alternatively, it is possible to formulate the traversal iteratively so it can be expressed in FORTRAN.

A well chosen MAC is crucial for treecodes. It strongly influences speed, and accuracy, and in a parallel implementation, it influences memory usage and utilization of parallel processing hardware [11]. Whenever the MAC succeeds, time is saved, but an approximation is made and errors of some magnitude are introduced into the result. Thus, the MAC must strike a balance between speed and accuracy. In their original formulation, Barnes and Hut[3], introduced a parameterized MAC based on an opening angle, θ. The BH MAC asserts that the multipole approximation is acceptable only if the ratio of the size of a terminal cell to the distance from the body to the center-of-mass of the cell is less than a tunable parameter, θ. The geometry is illustrated in Figure 1.

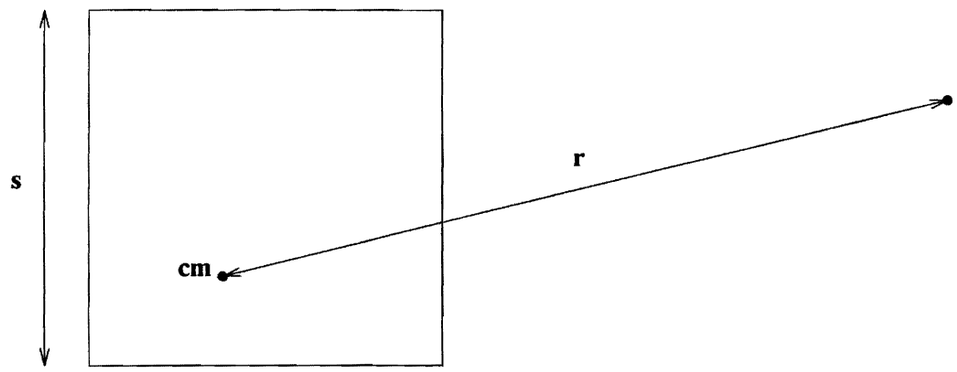

Figure 1: Geometry for BH MAC test. The multipole approximation is acceptable if and only if $s/r < \theta$.

Obviously, the precise value of the parameter θ is crucial. A very low value implies that a body must be very distant from a cell before the multipole approximation is acceptable. Since the multipole approximation is rarely acceptable, the tree is traversed to a deeper level, and a large number of interactions are computed. A larger value of θ implies greater confidence in the multipole approximations, and fewer interactions. The number of interactions scales approximately as θ^{-3} [11], so it is clear that the performance of the algorithm is sensitive to θ. Astrophysical simulations employ a fairly small range of values, $0.7 \leq \theta \leq 1.0$, and the multipole approximation is usually terminated after the quadrupole term. However, Appendix A shows that it is a simple matter to extend the multipole approximation to arbitrarily high order.

The BH MAC has been used in numerous simulations, and has been extensively analyzed [12]. Unfortunately, commonly used values of θ can introduce large systematic errors when applied to certain configurations of bodies. Furthermore, the geometric arrangements that lead to trouble arise naturally in astrophysical N-body simulations.

3 Detonating galaxies

Now we turn to problems that can arise when the BH MAC is used with values of $\theta \gtrsim 1/\sqrt{3}$. Consider the situation in Figure 2. A large (primary) galaxy, P, is at the lower left corner of one of the cells, C, in the tree. Assume that the cell is of unit size. A smaller (secondary) galaxy, S, is partially inside the the upper right corner of C, i.e., some of the bodies in S are inside C and some are outside. Unless θ is very small, the internal dynamics of the secondary can be severely disrupted by errors introduced by the BH MAC.

Since the primary is, by construction, much heavier than the secondary, the center-of-mass of C is near its lower left corner. Now consider the operation of the BH MAC for a body, X, bound to the secondary, but just outside the boundary of C [1]. The distance from X to the center-of-mass of C is about $\sqrt{2}$ in the figure, but can clearly be as large as $\sqrt{3}$ in three dimensions. Thus, for any value of $\theta > 1/\sqrt{3}$, the BH MAC reports that the multipole approximation is acceptable. That is, as far as body X is concerned, any part of the secondary that has passed

[1] Note that the point X is outside of cell P, so this problem is distinct from the "self-acceleration" problem discussed by Hernquist[13]. In any event, the effect is certainly not "negligible for $\theta \lesssim 1.2$."

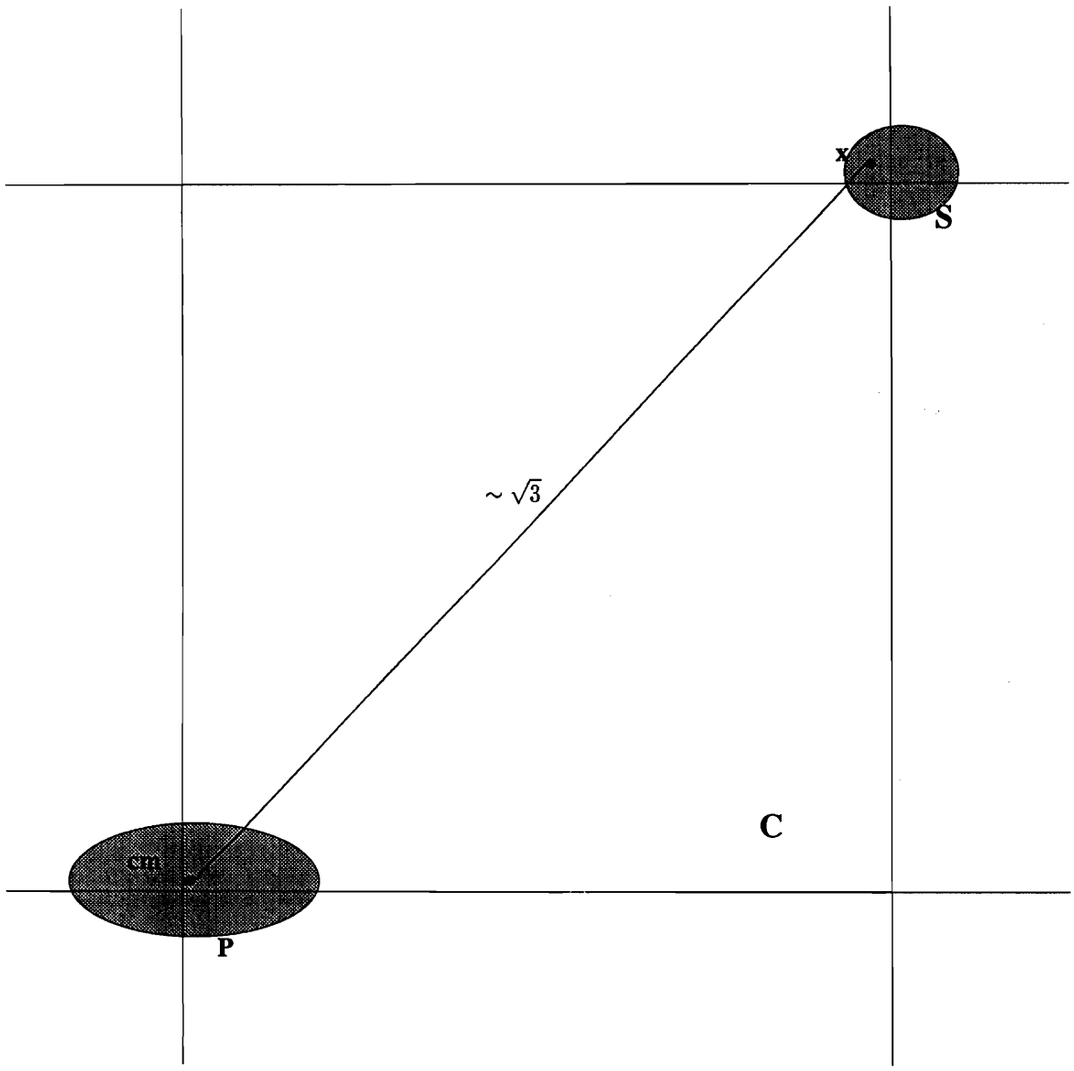

Figure 2: Geometry of "Detonating Galaxy" pathological situation for the BH MAC criterion.

inside the boundary of C will be treated as though its mass is concentrated at the center-of-mass of C. By artificially removing the mass of the secondary to the far corner of C, the self-gravity of the secondary is drastically (and incorrectly) reduced. The quadrupole correction for cell C has the correct sign, but its magnitude is much too small to significantly improve matters. In a simulation, the secondary will "explode" as it crosses the boundary of C and loses its self-gravity. If it passes entirely into C, it's dynamics return to "normal", but the damage has been done.

Systems susceptible to this source of error are not as rare as one might hope. Consider a merger simulation with $G = 1$, $dt = 0.025$, Plummer softening parameter, $\epsilon = 0.03$, and two truncated spherical Jaffe models with the parameters shown in Table 1 [2].

	M	$\vec{x}_0$	$\vec{v}_0$	r_J	N
primary	1.0	(0.3, 0.3, 0.3)	(0,0,0)	1.0	15,000
secondary	0.05	(7.,7.,7.)	(-0.25, -0.25, -0.25)	0.2	3,000

Table 1: Dynamical parameters for two-galaxy simulation.

Both models are truncated at a radius of $10r_J$. The simulations were carried out using Barnes-Hut trees in which the root cell was a cube of size 24.0, centered on the origin. We expect a situation like Figure 2 to occur near $t = 4.0$, when the secondary passes through the corner of the cell at $\vec{x} = (6, 6, 6)$.

Figure 3 shows the energy (T+U) and the 10% and 50% mass radii of the secondary in the system described above for several values of θ. The discontinuity in energy and mass-radius at $t = 4.0$ is the result of the error introduced by the Barnes-Hut MAC. Notice that a discontinuity in energy occurs even for the extremely conservative value of $\theta = 0.65$. Implementing Hernquist's additional test that a body cannot be inside a cell with which it interacts makes a substantial difference, but falls far short of actually repairing the problem. Without Hernquist's criterion, at $\theta = 0.7$, the secondary simply detonates near $t = 4.0$. Using Hernquist's criterion, the 50% mass radius jumps from 0.18 to 0.25. Below $\theta = 0.65$ the Hernquist criterion predictably makes little difference. At such small values of θ, it is extremely unlikely for a body to pass the BH MAC and fail the Hernquist criterion (even for a pathological configuration like Figure 3.)

One might hope that by randomly moving the origin or size of the root of the BH tree, one could avoid the problem. After all, the disaster strikes as a result of a "conspiracy" between the positions of the galaxies and the positions of the cells in the BH tree. We claim without proof that this is not a viable solution. If the root cell moves around, it may not be predictable when the disaster occurs, but it is certain to occur eventually. Even one timestep with acceleration errors approaching 30% (see Figure 4) seems to us to be too large to take chances with.

Figure 4 confirms that the errors primarily affect the bodies in the secondary. It shows the cumulative probability distribution for relative errors in the acceleration for three distinct samples chosen from the $\theta = 0.7$ simulation. At $t = 3.0$, (well before the discontinuities in Figure 3), the errors appear well-behaved, with the majority of errors below 0.2%. The situation is dramatically different at $t = 4.0$. The distribution is clearly bi-modal, with the majority of bodies still subject to errors less than 0.2%. However, about 10% of the bodies are subject to relative errors exceeding 20%. The average error is over 4%, which is roughly consistent with the size of the discontinuity in energy shown in Figure 3 (about 1%). When the sample is restricted to the bodies in the satellite galaxy, the situation is even worse. Almost 70% of the secondary is subject to errors greater than 20%, and a significant fraction is subject to errors approaching 100%.

It is curious that the BH MAC can give rise to such large errors when it has been tested so extensively. Indeed, very similar physical system have been used to verify the accuracy of treecodes [12, 15], but the galaxies in these test case were of equal mass and on an orbit parallel to one of the Cartesian axes. In this configuration, the error does not arise, and the algorithm "passed" the test.

[2] This is not an entirely contrived example. A very similar system was evolved using the MD MAC (Sec. 4.1), and reported in Salmon et al[14].

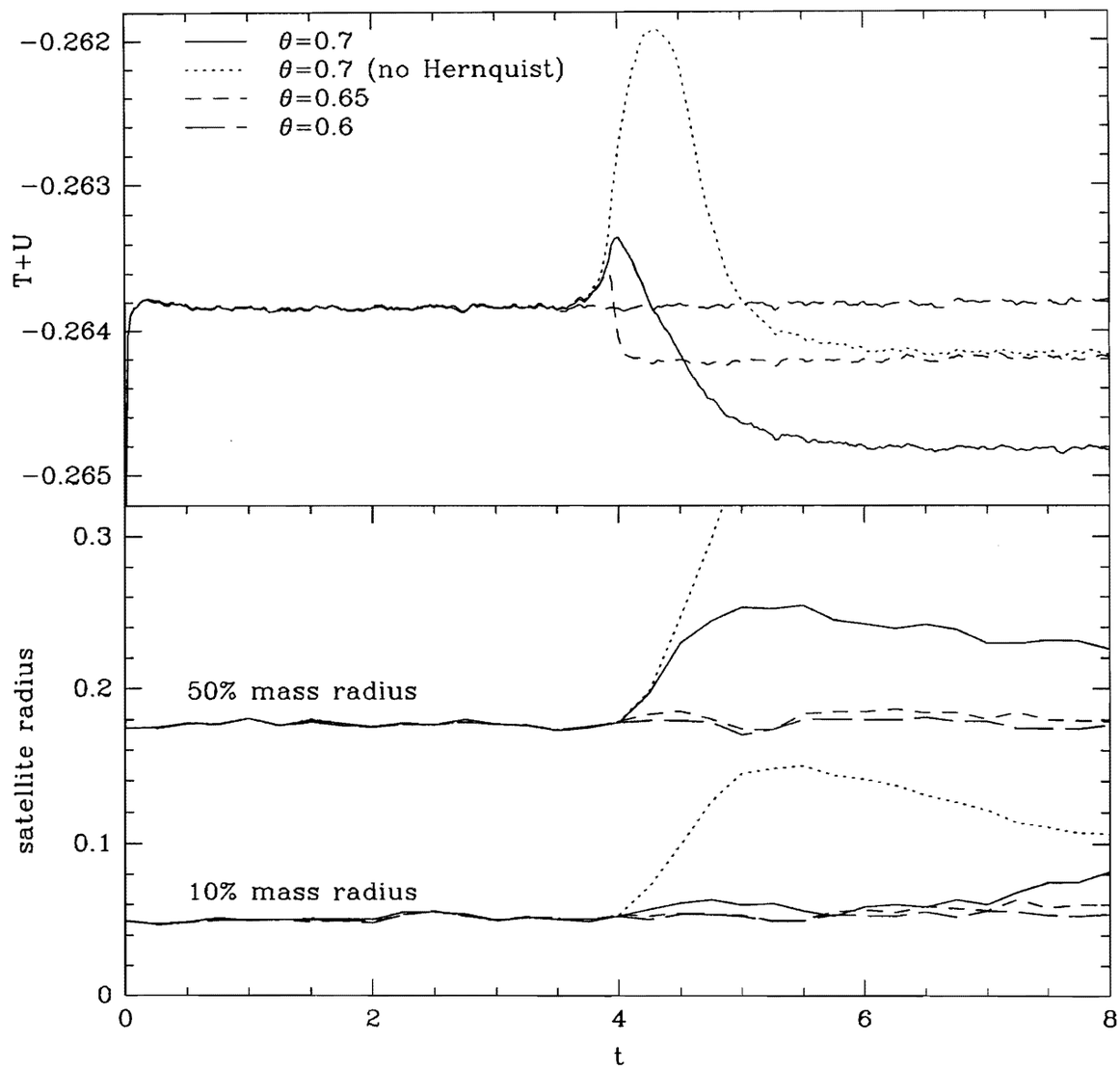

Figure 3: Total energy, T+U and selected mass-radii vs. t for a simulation consisting of a heavy primary and a light satellite (mass ratio, 20:1) on a headon orbit along the $x = y = z$ line.

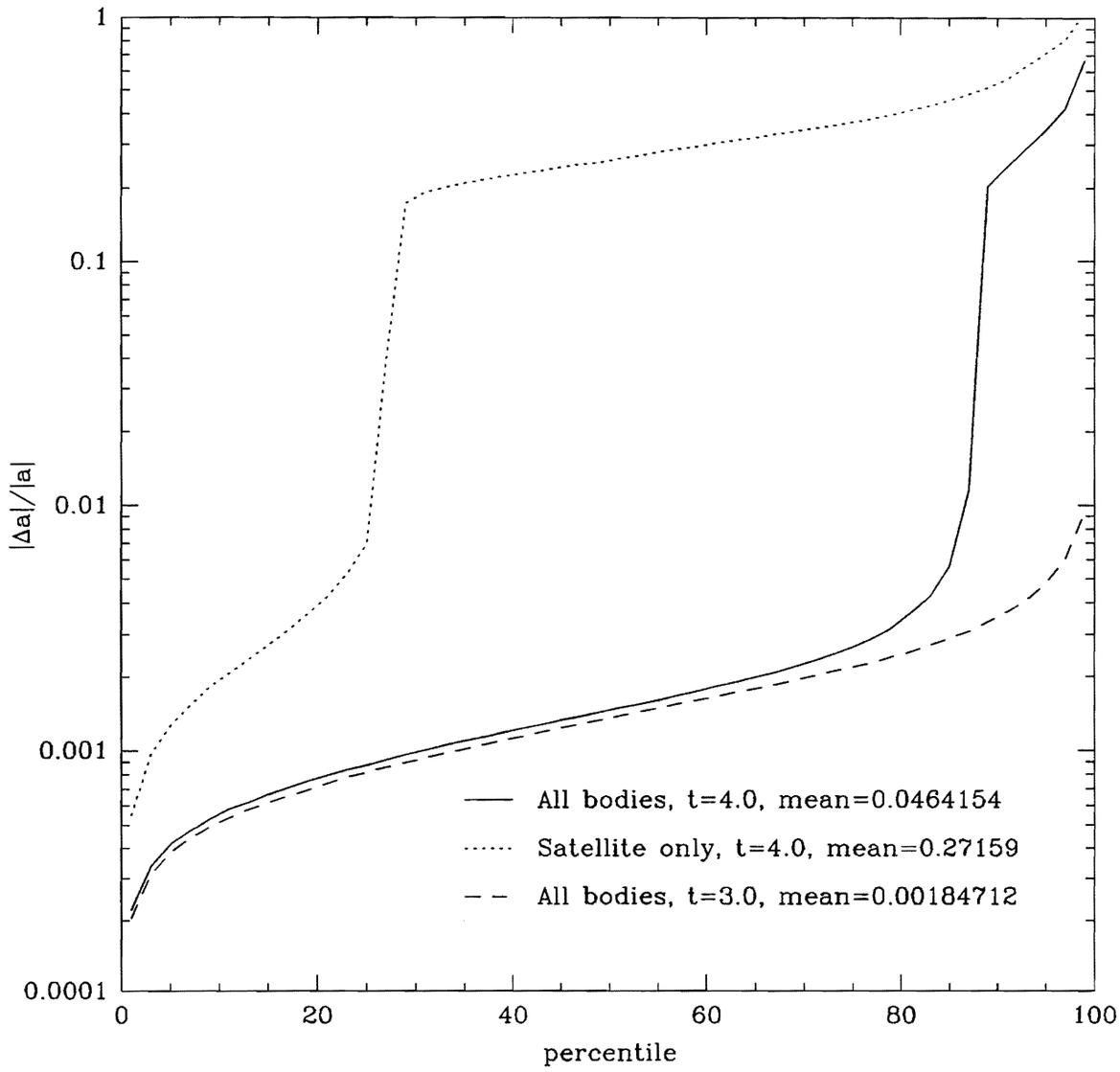

Figure 4: Cumulative probability distribution for relative errors in the acceleration for three samples from the $\theta = 0.7$ model shown in Figure 3. The errors are computed as $\frac{|\Delta\vec{a}|}{|\vec{a}_{exact}|}$, where $\vec{a}_{exact}$ is computed by direct, $O(N^2)$ summation.

A number of authors have empirically measured errors introduced by the BH MAC over a range of values of θ [13, 16, 12, 10]. The situation described in this section is not inconsistent with the previous work. It only serves to point out the danger of relying on statistical measures of the error. The cited papers all report rms and/or average relative errors. Even in the worst case reported here ($\theta = 0.7$, no Hernquist criterion), the errors are negligible almost everywhere, almost always. Simple statistical measures are not sensitive to the presence of a rare, but catastrophic, "tail" in the distribution of errors. Unfortunately, even though the errors are statistically rare, they are not benignly distributed. The error only affects a small fraction of the bodies for only a brief time but it does so in such a way as to significantly disrupt the physics of the secondary galaxy.

Even if the other authors had reported maximum relative errors, rather than rms or means, it is unlikely that they would have had cause to warn against $\theta \gtrsim 1/\sqrt{3}$. Only certain, rather special geometrical arrangements of bodies give rise to errors of the magnitude of Figures 3 or 4. Analysis of systems with isolated Plummer models or pairs of equal-mass Plummer models will not exhibit such catastrophic behavior. Indeed, we may note that two types of mass distributions are modeled very accurately by the quadrupole approximation: uniform distributions of mass, and distributions with the mass concentrated at a single point. Thus, test cases in which most of the matter is uniformly distributed and/or concentrated at a single point are tailor-made for the quadrupole approximation, and will exhibit negligible errors. It is possible that in the course of a single large cosmological simulation two halos will interact with a geometry similar to Figure 2, but the number of affected bodies will likely be very small compared to the total. The "glitch" in total $T + U$ will probably be lost in the noise, and it would be hard to identify such an occurrence if one were not looking for it *a priori*.

Unfortunately, it is difficult to see how the empirical analyses can be significantly improved. Certainly, one could imagine a suite of tests which would include, among others, isolated ellipticals and disks, equal and unequal pairs of galaxies in various orbits and orientations, triplets, complex interacting systems, cosmological examples from various epochs, etc. It would be extremely time-consuming to analyze such a suite, and, more importantly, it would be impossible to know that it was complete. Is there, for example, another pathological configuration lurking, untested, just outside the phase-space covered by the suite? The failure of previous empirical studies to detect a situation as simple as Figure 2 make us pessimistic about our ability to design a complete suite of empirical tests.

The alternative to additional empirical testing is the application of analytic methods. It is possible to place strong, analytic bounds on the errors arising from the multipole approximation, and to adjust the MAC so that one is mathematically guaranteed to obtain a specified level of accuracy. Applied naively, this procedure implies very low (and hence costly) values of θ, and/or the use of very high-order multipoles. We shall return to this in Section 5, where we discuss an economical and mathematically defensible alternative MAC.

4 Alternative MACs.

First, however, we consider some simple alternatives to the BH MAC which eliminate the detonating galaxy pathology. In effect, we propose MACs which do not allow use of the multipole approximation for points like X in Figure 2. The alternative MACs of this section are recommended as "patches" to existing treecodes. Implementing them in an existing code would require changing a very few lines of code and would eliminate the possibility that a significant source of error could invalidate a simulation. While they certainly eliminate the detonating galaxy pathology, they should still be considered a stopgap, as large errors are still possible unless θ is made quite small (probably unacceptably so). The methods of Section 5 are recommended as a more viable, long term solution. After we present two new MACs, we will compare them in terms of accuracy and performance.

4.1 The Min-Distance MAC

The BH MAC of Figure 1 is based on the rather vague (but certainly correct) assertion that the accuracy of a multipole approximation at location X is determined by the ratio of the "size" of the cell, to the "distance" from X to the cell. It is clear from Figure 2 that the errors discussed in Section 3 arise because the distance from X to the bodies in the secondary inside C is much less than the distance from X to the center-of-mass of C. It is the latter which is used by the MAC, and the former which is the source of the error. This observation suggests an alternative

MAC which substitutes the *minimum distance from a body to any point in the cell* for the distance used by the BH MAC. We shall refer to this as the "minimum-distance" (MD) MAC. The geometry is illustrated in Figure 5.

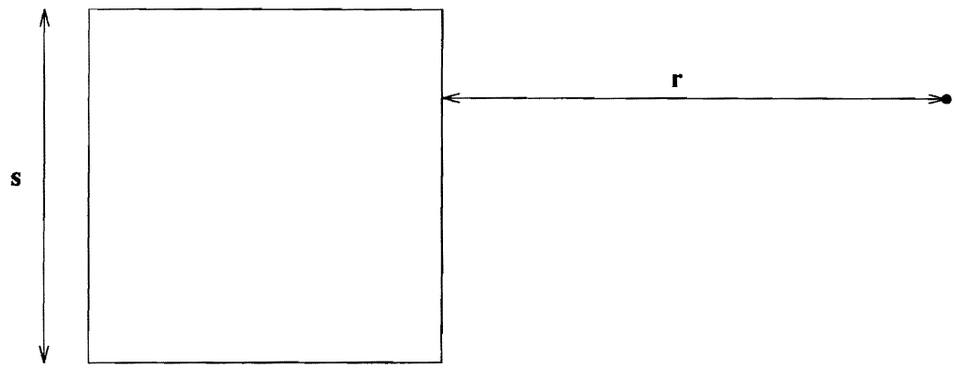

Figure 5: Geometry of for the MD MAC criterion. The multipole approximation is acceptable if and only if $s/r < \theta$.

One interesting aspect of the MD MAC is the fact that it is completely independent of the contents of the cell. That is, it is possible to evaluate the MD MAC without knowing anything about a cell other than its size and location. In contrast, the other MACs we consider may be evaluated only after the center-of-mass has been determined, which in turn requires that the positions of all bodies in the cell are known. The ability to evaluate the MAC before the contents of a cell have been determined makes it attractive for use on parallel distributed memory computers [11], where one might wish to evaluate the MAC for cells whose data resides in another processor's memory.

4.2 The Bmax MAC.

Another MAC is motivated by Eq.(69), according to which the largest possible error, for any distribution of material in a cell, is proportional to the magnitude of the monopole term, and a monotonically increasing function of b_{max}/r, where b_{max} is the maximum distance from the multipole origin, $\vec{r}_0$, usually taken to be the center-of-mass of the cell, to any other point in the cell, i.e.,

$$b_{max} = \max_{x \in C} |\vec{x} - \vec{r}_0| \,. \tag{1}$$

Since the error is an increasing function of b_{max}/r, it is natural for b_{max}/r to appear in the MAC. The "Bmax" criterion, shown in Figure 6, allows that the multipole approximation is acceptable iff $b_{max}/r < \theta$.

We now have three MACs, which may be used with multipole expansions of arbitrary order, and arbitrary values of θ. It is imperative that we find some consistent and meaningful way to compare these methods with one another. As we argued above, running test models through the MACs at various values of θ and p is both time-consuming and ultimately not a satisfactory indicator of reliability. The alternative is to determine the largest possible error that would be "accepted" by each MAC, and use that as a figure of merit by which they can be compared. The likelihood of obtaining an error comparable to the worst case is, or course, small, but it cannot be ignored, as the detonating galaxy of Section 3 attests. The configuration of particles in real simulations is anything but random, so it is unwise to discount a worst-case analysis simply on the basis that it requires a "conspiracy" in the configuration of bodies. The detonating galaxy of Section 3 requires just such a conspiracy, and it constitutes an entirely reasonable intermediate state of an interesting simulation.

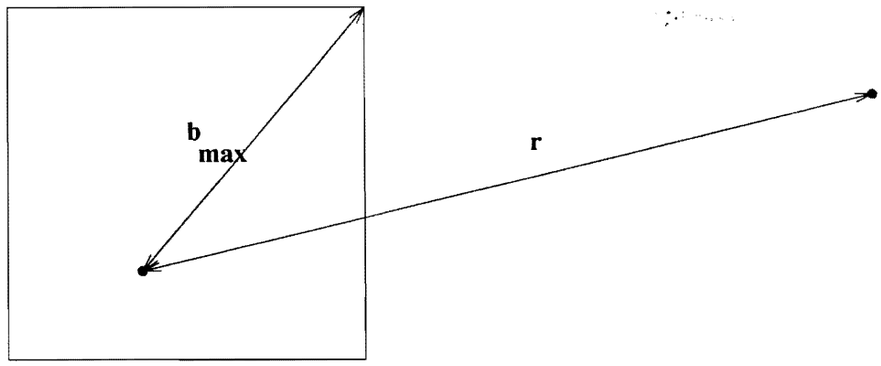

Figure 6: Geometry of for the Bmax MAC. The multipole approximation is acceptable if and only if $b_{max}/r < \theta$.

4.3 *Accuracy of alternative MACs*

We shall be computing relative errors (dimensionless) and all the MACs under consideration use a dimensionless value of θ, so we may, without loss of generality, restrict attention to a unit cell with unit mass, and a gravitational constant of unity. We are seeking the distribution of mass that leads to the most inaccurate multipole approximation of a given order p, at a location acceptable to a particular MAC with a given value of θ. Intuitively, the multipole approximation is least accurate when a large portion of the mass is far from the center-of-mass of the cell. Thus, we conjecture that the largest errors occur when the mass distribution consists of two point masses at opposite corners of the cube, i.e., separated by $\sqrt{3}$. Furthermore, we conjecture that the worst-case for the position of the measurement point is as near as possible to one of the masses, i.e., on the line that connects the two point masses. The situation is now one-dimensional, and is represented schematically by Figure 7 and analytically by Eq.(2). The relationship between the values of x_1, x_2 and r depends on which MAC is under consideration, but in any case x_1, x_2 and r, are determined trivially from Eq.(2) once m_1 is chosen.

$$\begin{aligned} m_1 + m_2 &= 1 \\ m_1 x_1 + m_2 x_2 &= 0 \\ x_2 - x_1 &= \begin{cases} \sqrt{3} & \text{BH or MD} \\ 1 & \text{Bmax} \end{cases} \\ r &= \begin{cases} \frac{1}{\theta} & \text{BH or Bmax} \\ x_2 + \frac{1}{\theta} & \text{MD} \end{cases} \end{aligned} \tag{2}$$

The exact acceleration, a_{exact}, the order-p multipole approximation $a_{(p)}$ and the relative error, e are readily computed as follows:

$$\begin{aligned} a_{exact} &= \frac{m_1}{(r - x_1)^2} + \frac{m_2}{(r - x_2)^2} \\ a_{(p)} &= \frac{1}{r^2} \sum_{n=0}^{p} (n+1) \frac{m_1 x_1^n + m_2 x_2^n}{r^n} \\ e &= \left| \frac{a_{exact} - a_{(p)}}{a_{exact}} \right| \end{aligned} \tag{3}$$

The problem is thus reduced to a one-dimensional optimization, i.e., find the value of m_1 in the range $0 \leq m_1 \leq 1$ that gives rise to the largest possible value of e. Simple numerical techniques using parabolic interpolation [17] provide us with a maximum possible relative error for any (θ, p) pair and any MAC. The results are shown in Figure 8.

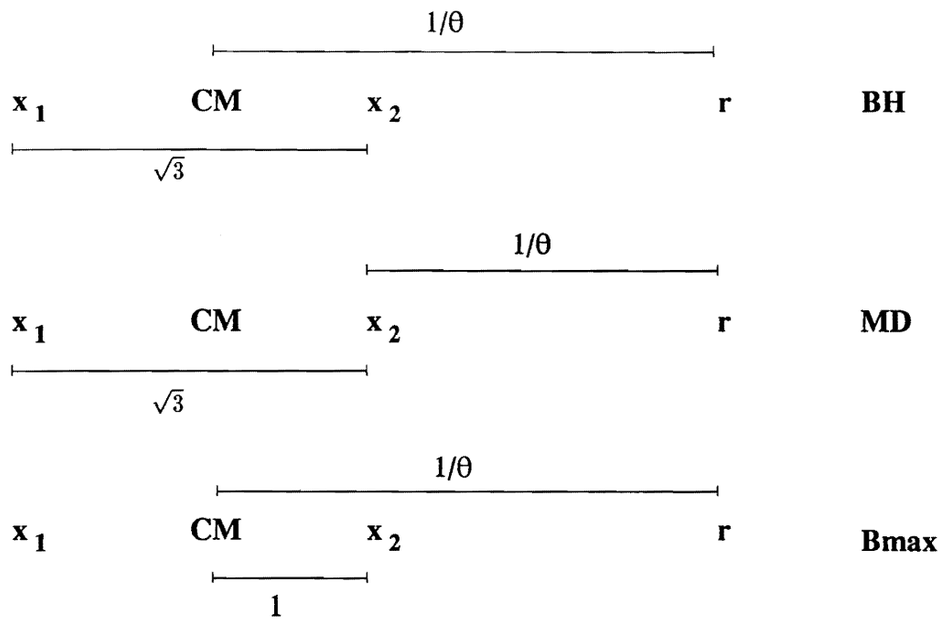

Figure 7: Relationships between positions of two point masses inside cell (at x_1 and x_2), and the point at which the acceleration is measured, r, for the three MACs.

The errors displayed in Figure 8 are an order of magnitude or more worse than the empirical results obtained by a number of authors [13, 16, 12, 10]. Again, we may account for this discrepancy by noting that the other authors report an rms or an average error. The majority of interactions have very much lower errors than the upper bound shown in Figure 8.

Inspection of Figure 8 reveals that even modest accuracy requires either a very low values of θ, or a very high order multipole approximation. For example, relative errors of 5% or less are guaranteed by the standard (p=2) BH MAC only for $\theta < 0.25$. Since the number of interactions is roughly proportional to θ^{-3} [11], such a simulation will compute almost 30 times as many interactions as one performed with a more conventional value of $\theta = 0.75$. On the other hand, the $\theta = 0.75$ simulation risks the introduction of unbounded errors by the detonating galaxy mechanism of Section 3. The situation is only somewhat better if one is willing to evaluate the multipole approximation up to order $p = 8$, where 5% error bounds are guaranteed with $\theta < 0.42$.

4.4 Computational cost of alternative MACS

Figure 8 is not immediately useful for assessing which MAC is optimal for achieving a desired level of accuracy. Different MACs cannot be compared at the same value of θ because the same value of θ leads to vastly different computational expense when used with two different MACs. There are two possible ways to compare MACs. One could fix some measure of accuracy, and compare on the basis of computational expense, or one could fix the computational expense and compare on the basis of accuracy. For the purposes of this discussion, the measure of accuracy employed is the maximum possible relative error in the acceleration.

It has been noted [3, 13, 10, 11] that the computational expense of a tree-code depends on the details of the particle distribution. Thus, we do not have the luxury of a precisely defined, problem-independent measure of the computational expense of a given MAC. However, Salmon[11] shows that the number of interactions required to compute the force on all particles in a tree-code is proportional to the average volume of the *interaction region* of a unit-cell, regardless of the particular form of the MAC. The interaction region of a cell is defined as the region for

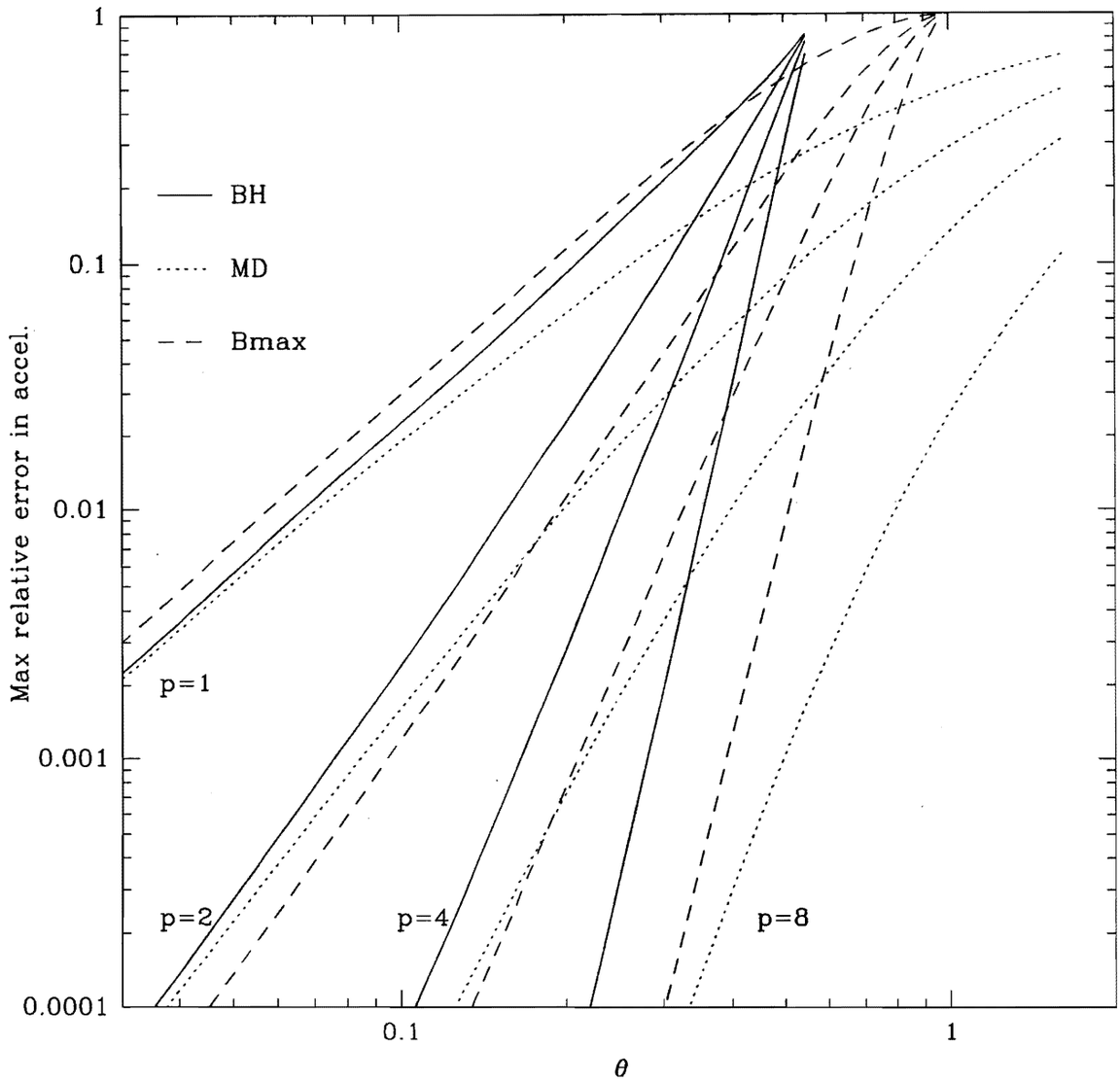

Figure 8: Plot of maximum fractional error in the force, as a function of θ for values of the multipole order, $p = 1, 2, 4$ and 8, and the BH, MD and Bmax MACs. The different values of p may be distinguished on the plot by their asymptotic slope at small θ, for which $e \propto \theta^{(p+1)}$. It is worth noting that the $p = 1$ results are valid as well for $p = 0$, i.e., monopole-only calculations, since the dipole moment about the center-of-mass vanishes exactly.

which the MAC fails for the parent of the cell but passes for the cell itself. That is, it's the region in which bodies will actually interact with the cell. Computing this volume is a simple exercise in analytic geometry. For the three MAC's under consideration, the interaction volume is:

$$\begin{aligned} V_{BH} &= \frac{28\pi}{3\theta^3} \\ V_{MD} &= \frac{28\pi}{3\theta^3}\left(1+\frac{9\theta}{4}+\frac{9\theta^2}{2\pi}+\frac{3\theta^3}{4\pi}\right) \\ V_{Bmax} &= \frac{28\pi}{3\theta^3}\text{mean}(b_{max}^3). \end{aligned} \tag{4}$$

Note that the expression for the interaction volume of the Bmax MAC contains yet another average. Unlike the other two MACs, the interaction volume of a cell depends on the contents of a cell. In particular, it depends on the value of b_{max}. Clearly, we have

$$\frac{3\sqrt{3}}{8} \leq b_{max}^3 \leq 3\sqrt{3}, \tag{5}$$

from geometric considerations, but it is difficult to be more precise. For a contrived ensemble of unit-cells in which the center-of-mass is a random variable uniformly distributed within the cell, we have:

$$\text{mean}(b_{max}^3) \approx 2.355 \tag{6}$$

Lacking a compelling reason to choose a different value, we shall use the value in Eq.(6) when a specific values of V_{Bmax} is required. We note, however, that the the true value is dependent on the particulars of the particle distribution, and the only general statement one can make with certainty is Eq.(5).

Additional complications arise when comparing different values of p. We may characterize the p-dependence of the computational expense, together with the MAC and θ dependence as:

$$T_{compute} \propto V_{MAC}(\theta) f(p) \tag{7}$$

since the total number of interactions is proportional to $V_{MAC}(\theta)$, and the computational expense of each interaction clearly depends on the multipole order p, but not on θ or the MAC.

Evaluating the multipole approximation even at low order on modern processors is an extremely delicate affair. The real time required to perform an interaction depends strongly on parameters of the particular computer and implementation. However, we may draw some tentative conclusions based on an ad hoc model of the computational expense. In Salmon[11], it is shown that the number of additions and multiplications required to compute a body-cell interaction with multipole terms through order p is proportional to $\binom{p+3}{3}$. This analysis does not include the time required to compute the distance vector $\vec{r}$ itself, nor does it include the time required to compute the 1/r from $\vec{r}$, and of course, the constant of proportionality is an unspecified value of order unity. Somewhat arbitrarily, we adopt the following expression for the computational expense of a single interaction through order p[3]:

$$f(p) \approx 24 + 5\binom{p+3}{3}. \tag{8}$$

Figure 9 shows a plot of $T_{compute}$ vs. relative error. In principle, this plot answers the question of which MAC is "optimal" for any desired level of accuracy. One simply selects the leftmost curve at the desired level of accuracy. According to the figure, the MD MAC, at various values of p is superior to both the BH and the Bmax MACs. Not surprisingly, as the required accuracy increases, the optimal value of p increases as well, with $p = 4$ dominating $p = 2$ near a required relative accuracy of 10%. Similarly, $p = 8$ is superior to $p = 4$ at a required relative accuracy of 1%. Of course, intermediate values of p become optimal at intermediate values of required relative accuracy.

[3] This expression has the correct asymptotic form, and roughly correct values for $p = 0$ and $p = 2$.

Figure 9 does not directly give a value of θ. In practice, one must determine a desired level of accuracy based on external consideration, then use Figure 9 to determine the optimal MAC and value of p, and finally, use Figure 8 to find the appropriate value of θ which gives the desired level of accuracy for the specific MAC and value of p. For example, for 10% relative error, Figure 9 tells us that the $p = 2$ and $p = 4$ MACs are approximately equivalent. Inspection of Figure 8 reveals that 10% relative error can be achieved with $p = 2$ and $\theta \approx 0.54$ or $p = 4$ and $\theta \approx 0.88$. Alternatively, one could use the conventional BH MAC with $p = 2$ and $\theta \approx 0.31$, but according to Figure 9 this would be about twice as expensive.

Figure 9 however, must be used with great caution. Eq.(6), (7) and (8) which are used to generate the abscissa of Figure 9 are crude estimates, and cannot precisely reflect the running time of real simulations. In particular, the scaling of execution time with p is highly implementation dependent, and the appropriate value of mean(b^3_{max}) depends on the distribution of bodies in the simulation. Rather than treat the details of Figure 9 as anything more than suggestive, it is recommended that treecode users characterize the p-dependence and θ-dependence of their specific code and problem domain and generate a problem-specific analog of Figure 9. Then the procedure outlined above may be carried through with greater confidence.

5 The Sum MACs

The new MAC's introduced in Section 4 improve upon the conventional BH MAC by eliminating a significant source of error. Nevertheless, Figures 8 and 9 imply that even the alternative MACs require very low values of θ and/or very high-order expansions to guarantee reasonable levels of accuracy. In this section we introduce a completely new MAC that can achieve guaranteed levels of accuracy with substantially fewer calculations.

To prevent confusion, and to avoid long-winded repetition of stock phrases, let us define some of the terminology to be used in this section. We make no claim that this terminology is of general significance; it simply makes the following discussion more precise. First, the *acceleration* of a body is generally the vector sum of N *terms*. That is, the *terms* are the result of individual body-body interactions obeying Newton's or Plummer's Green's function. Bodies may be grouped into *cells* according to some geometric criterion. The bodies in a *cell* each contribute a *term* to an *acceleration*. The vector sum of all the terms contributed by a single *cell* is referred to as a *partial acceleration* or just a *partial*. It is the *partial accelerations* that are the subject of the multipole approximation. We distinguish between the *approximate partial* as computed according to the multipole approximation, and the *exact partial*, computed by direct summation over the bodies belonging to the *cell*. The magnitude of the difference between the exact and approximate partials is denoted $\Delta a_{partial}$. Similarly, the magnitude of the difference between exact and approximate total acceleration is denoted Δa_{tot}. The MAC is responsible for determining when an *approximate partial* is accurate enough to be used instead of an *exact partial*. Typical treecodes use MACs with the property that the number of partials that contribute to any given total acceleration grows logarithmically with increasing N. Finally, we will use the notation $\lceil x \rceil$ to denote an analytic upper bound on the quantity x. These bounds are derived in detail in the Appendix.

As we saw in the previous section, conventional MACs are characterized by the relative error they allow to be introduced into each partial. With this type of MAC, large magnitude partials contribute large errors and small magnitude partials contribute small errors. If the individual partials all have comparable magnitudes, then this scheme is a reasonably efficient one for achieving the ultimate goal of limiting the total error. However, if the individual partials occur over a large range of magnitudes, then this type of MAC is not at all efficient. One wastes a great deal of time evaluating the small magnitude partials to very high (absolute) accuracy, despite the fact that most of the total error will be contributed by the errors in the large-magnitude partials. Figure 10 shows magnitudes of the partials in a typical cosmological simulation. Relative magnitudes (normalized by the total acceleration) occur over at least five orders of magnitude. The user is faced with a dilemma. If one opts for high accuracy, i.e., small θ, a great deal of effort is wasted reducing the relative errors of partials which are, themselves, small. On the other hand, if one opts for low relative accuracy, i.e., large θ, then one runs the risk of a single dominant interaction being computed with low accuracy.

The solution is to abandon the use of MACs based on the relative errors of individual partials, and consider a more general class of MACs that address absolute, rather than relative errors. Let's assume that the force-evaluation

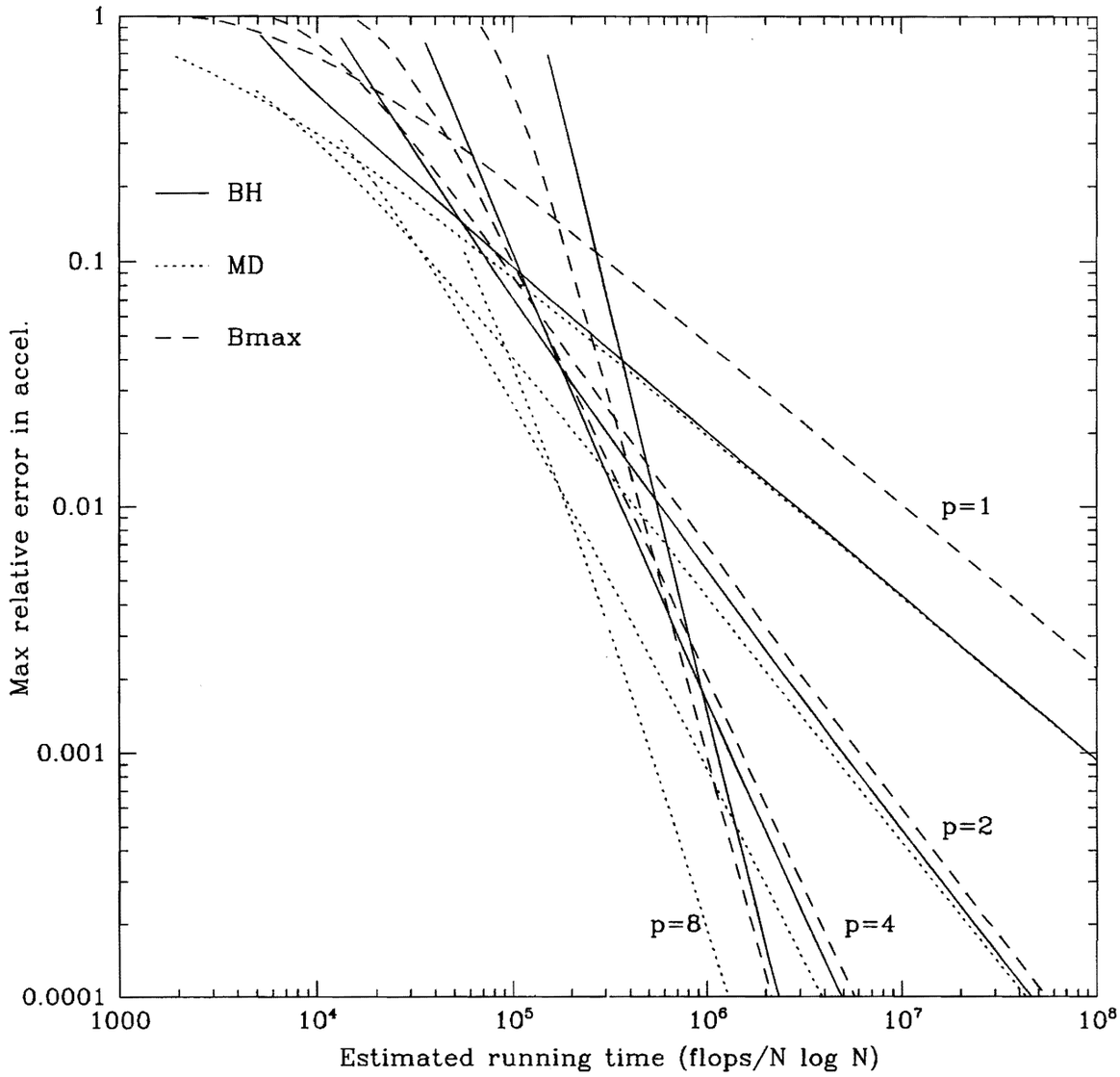

Figure 9: Plot of maximum fractional error in the force, as a function of computation time, estimated as $V_{MAC}(\theta) * f(p)$, for the BH, MD and Bmax MACs, and multipole orders, p=1, 2, 4 and 8. $V_{MAC}(\theta)$ and $f(p)$ are discussed in the text. The different values of p may be distinguished by their slope at asymptotic slopes at high accuracy, for which $e \propto T^{-(p+1)/3}$. Note that the time for the $p = 1$ case is computed as $V_{MAC}(\theta)f(0)$ because the multipole expansion through order $p = 1$ may be terminated after the monopole term, i.e., $p = 0$, since the dipole term vanishes identically.

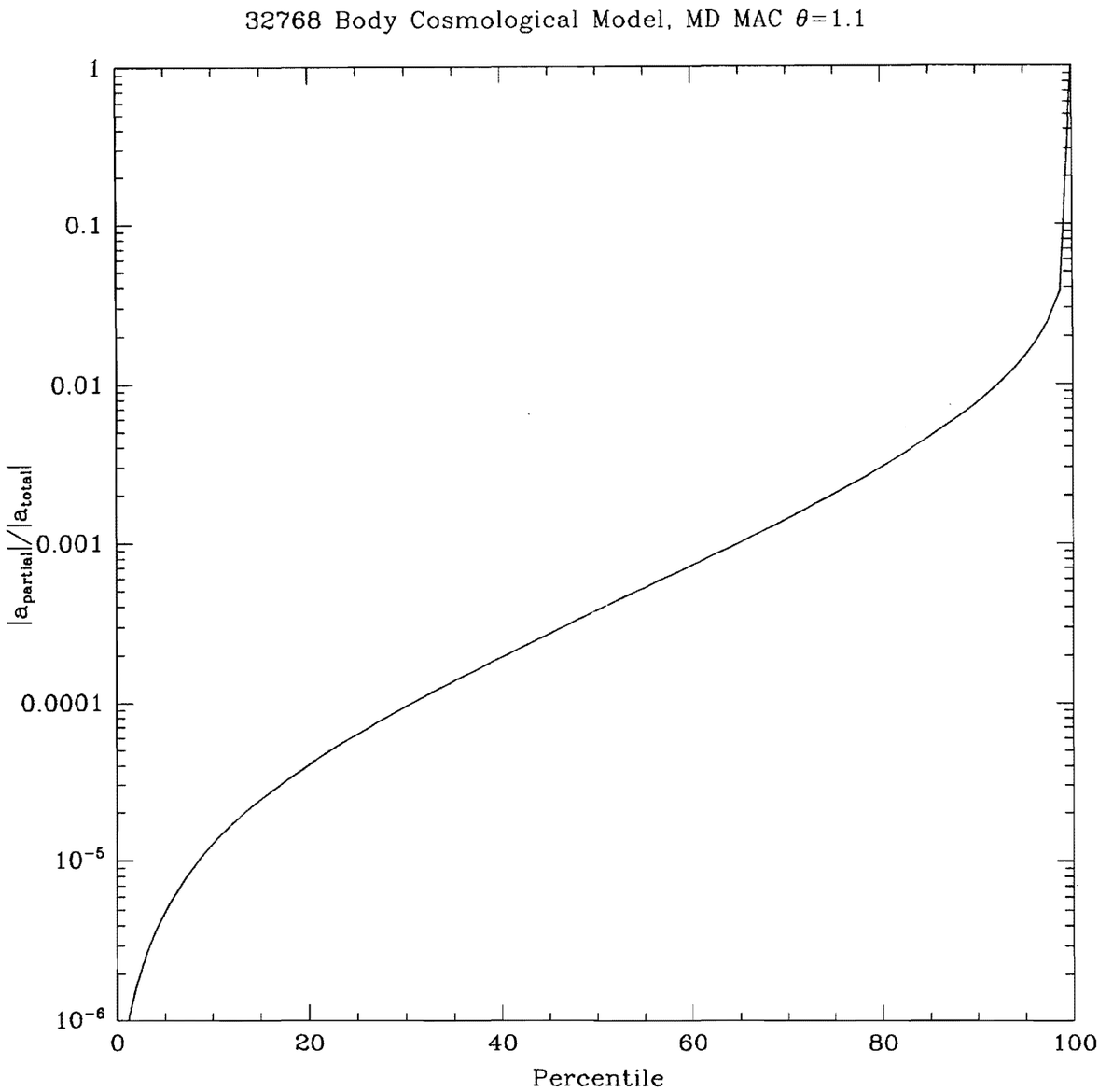

Figure 10: Percentiles of $|a_{partial}| / |a_{total}|$ for approximately 400000 partial accelerations computed using the MD MAC with $\theta = 1.1$ in a 32768 body system representing an evolved, clumpy cosmological simulation. The parameters of the system are described in Table 2.

procedure is modified so that the integrator asks for the acceleration of each body *with a specified tolerance,* Δ_{tot}, which may vary from one body to the next, or from one timestep to the next. The MAC should analytically guarantee that:

$$\lceil \Delta a_{tot} \rceil \leq \Delta_{tot} \tag{9}$$

The tolerance, Δ_{tot} has units of acceleration. This is quite different from the conventional procedure in two ways. First, the strictness of the MAC varies on a per-particle basis. This may be of considerable value in "individual timestep schemes", [18, 19, 20]. Second, we are seeking to control the magnitude of the error in the total acceleration, rather than the relative error of each partial.

This discussion leads us to ask about bounds on the absolute error rather than the relative error associated with each acceptable multipole term. In the Appendix we show that the maximum possible error after p multipole terms have been added is expressible in terms of moments of the mass distribution inside the cell. Eq.(60), which gives the maximum possible absolute error in a partial acceleration after p multipole terms have been added, is reproduced here:

$$\Delta a_{(p)}(r) \leq \frac{1}{r^2} \frac{1}{(1 - \frac{b_{max}}{d})^2} \left((p+2) \left(\frac{\lceil B_{(p+1)} \rceil}{d^{p+1}} \right) - (p+1) \left(\frac{\lfloor B_{(p+2)} \rfloor}{d^{p+2}} \right) \right). \tag{10}$$

The moments, $B_{(n)}$ are defined in Eq.(59), and $d = |\vec{r} - \vec{r}_0|$, the distance from r to the origin of the multipole expansion.

The obvious application of Eq.(10) is in the form of an "Absolute MAC", whereby the multipole approximation is acceptable if and only if

$$\lceil \Delta a_{partial} \rceil \leq \Delta_{partial}, \tag{11}$$

where $\lceil \Delta a_{partial} \rceil$ is given by the right-hand-side of Eq.(10). This MAC, however, begs the question of an appropriate value for $\Delta_{partial}$. Since we shall return to it shortly, we call it the "Absolute MAC".

First, however, we note the inequality:

$$\Delta a_{total} \leq \sum_{partial} \lceil \Delta a_{partial} \rceil, \tag{12}$$

which leads to the following expression which describes the "Sum MAC":

$$\sum_{partial} \lceil \Delta a_{partial} \rceil \leq \Delta_{tot} \tag{13}$$

If one assumes that the errors from individual partials are uncorrelated random variables then one also has:

$$rms(a_{total}) \leq \left(\sum_{partial} \lceil \Delta a_{partial} \rceil^2 \right)^{\frac{1}{2}}, \tag{14}$$

which leads to the "Sum Squares MAC":

$$\sum_{partial} \lceil \Delta a_{partial} \rceil^2 \leq \Delta_{tot}^2. \tag{15}$$

In each of these two MACs, the set of partials which are taken under the sum is unspecified. Clearly, for performance purposes, the set of accepted partials should be as small as possible. Strictly speaking, this is a constrained optimization problem (the constraints are that the partials that are selected must account for every body exactly once and that the Eq.(13) or 15 be satisfied. The quantity to optimize is the number of partials.), and as such, it is exceedingly hard to solve exactly. As is often the case with optimization problems, it is not necessary to find the smallest possible set of partials that satisfies the constraints. An approximate set will suffice.

We find that a method roughly analogous to "steepest descent" works remarkably well and can be implemented very efficiently using a priority queue. Conventional treecodes traverse the tree in an order governed by the tree itself, e.g., depth-first, with each set of daughter cells taken in a particular order. We propose to traverse the tree in an order governed by the magnitudes of the partial errors (as estimated by Eq.(10)). Instead of applying a MAC to whichever cell happens to be "next" in some canonical ordering of the tree, we always open the current *worst* cell, i.e., the unopened cell having the largest value of $\lceil \Delta a_{partial} \rceil$. When a cell is opened, values of $\lceil \Delta a_{partial} \rceil$ for its children are computed, and they are inserted into the priority queue. This process continues until Eq.(13) or Eq.(15) is satisfied [4].

There are a number of possible ways to implement priority queues [21]. Typically the cost of insertion and deletion is logarithmic in the size of the queue. Experience suggests that the queues arising in treecodes will contain up to about one thousand entries. The cost of maintaining such queues, while not unmanageable, is not entirely negligible either. It may be worth amortizing the cost of the queue maintenance over a number of timesteps. Notice that upon completion of the traversal, the set of partials that remains satisfies a simple criterion:

$$\lceil \Delta a_{partial} \rceil \leq \Delta_{max} \tag{16}$$

where Δ_{max} is the largest key in the queue. Clearly, this is simply the Absolute MAC with $\Delta_{partial}$ equal to Δ_{max}. If the value of $\Delta_{partial}$ were known in advance, the whole machinery of priority queues could have been bypassed, in favor of a depth-first or breadth-first traversal using the Absolute MAC. One traversal using one of the Sum MACs, however, can provide a value of $\Delta_{partial}$ that can be used for several timesteps because, by design, the physical environment of a body cannot change significantly during the course of a single timestep (if it did, the timestep was too large). Thus, it may be profitable to recompute $\Delta_{partial}$ (using a priority queue) every few timesteps, and perform a traditional tree traversal with the Absolute MAC in the interim. Obviously, the Absolute MAC requires some additional storage, as a distinct value of $\Delta_{partial}$ must be associated with each particle.

5.1 *Verifying the bound.*

The Sum MACs depend critically on the error bound set by Eq.(10). Eq.(10) is a strict analytic bound, so we can be sure that the error will not be exceeded. However, if the error is often very much smaller than the bound, we will find ourselves in much the same situation as before, i.e., we will face the dilemma of, on the one hand, a large amount of unnecessary work, or, on the other hand, the possibility of making unacceptably large errors. Figure 11 shows percentiles of the ratio of partial errors to the bound set by Eq.(10) for partials which were "accepted" by the Sum Square MAC in our 32768 body test model (see below). Very small errors indicate that Eq.(10) is not doing a good job of correctly estimating the error, which in turn, results in missed opportunities for using the multipole approximation, and hence wasted effort. Thus, it is "good" that the majority of errors shown in Figure 11 are large. The figure demonstrates that the bound set by Eq.(10) is very tight. In the mean, the actual error is about 20% of the bound, with occasional partials approaching the bound very closely.

5.2 *Numerical tests.*

Based on the experience cited in Section 3 concerning detonating galaxies, we are skeptical of empirical numerical tests of the new MAC. However, despite the fact that a positive result from an empirical test cannot be considered a proof-of-concept, a negative result can refute either the analysis or its underlying assumptions. As discussed above, average or rms errors can mask a variety of ills. We attempt to circumvent this by presenting the errors introduced by each MAC plotted as percentiles of relative error, i.e., errors normalized to the magnitude of the total acceleration. This is equivalent to plotting an estimate of the cumulative probability that the relative error on a particular particle is less than a given value. We have restricted ourselves to relatively modest numbers of bodies in the test cases because it is important to be able to calculate exact forces for comparison with the approximations produced by the treecode. In addition, in each system we compute the forces exactly on only a subset of the bodies.

[4] Minor difficulties arise because $\lceil \Delta a_{partial} \rceil$ can be infinite when the body and cell are too close. Since one is keeping a running total of errors, it is obviously necessary to treat the infinite values of $\lceil \Delta a_{partial} \rceil$ separately.

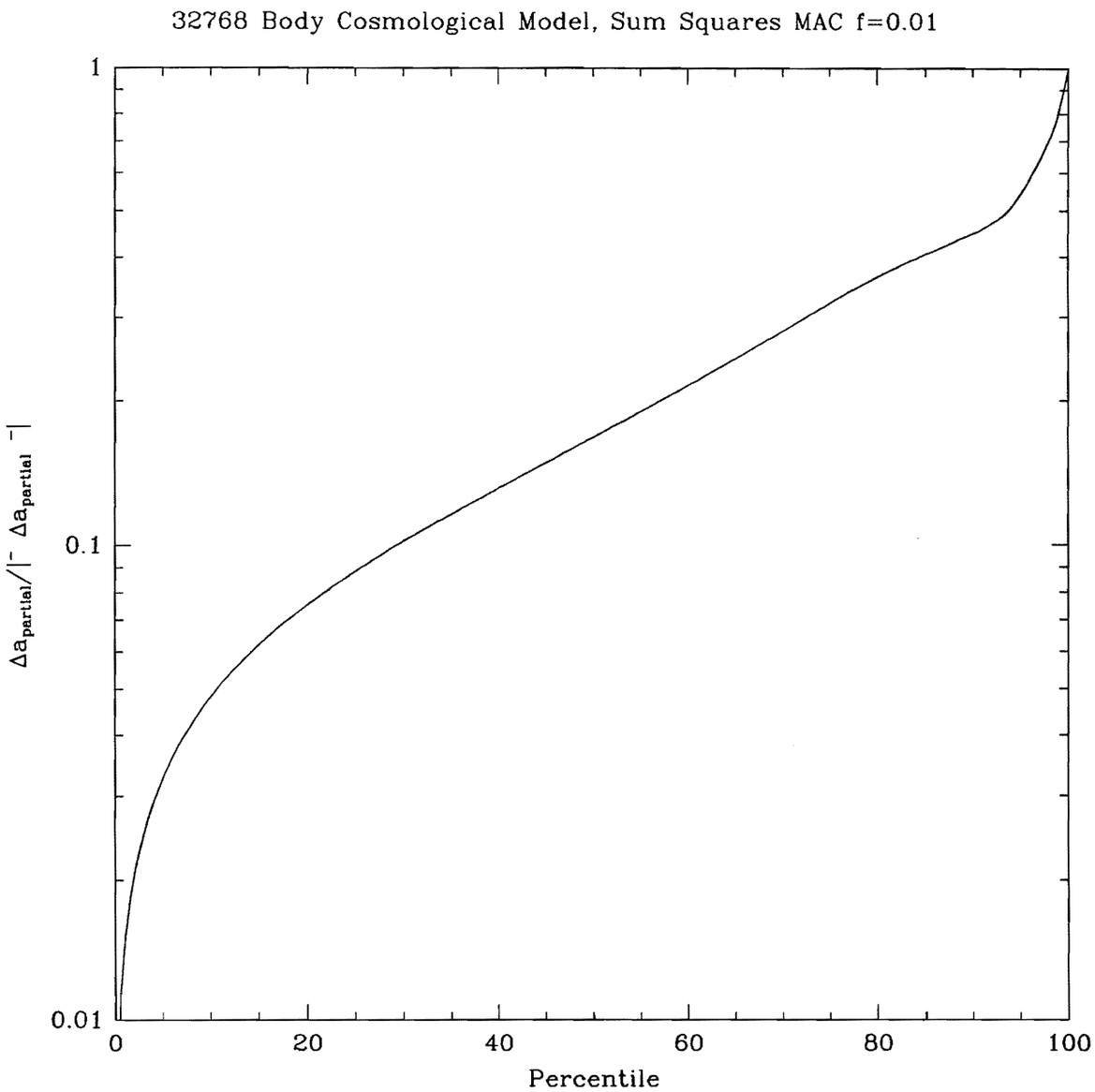

Figure 11: Percentiles of $|a_{partial}| / \lceil a_{partial} \rceil$ for the cosmological test model. The data consists of only those partial interactions which were "accepted" by the Sum Squares MAC with $f = 0.01$. The errors result from using the monopole approximation, and the bound is computed from Eq.(10) evaluated at ($p = 1$). The mean ratio in this sample is 0.22. The rms ratio is 0.28.

Calculating the exact force on N_{exact} bodies in a system of N bodies requires space proportional to N and time proportional to NN_{exact}. We treated three separate test cases.

- The initial state of the 18000 body simulation used in Section 3. Exact forces were computed for 4443 bodies.
- A single halo model with 54720 bodies. This halo was extracted from from a cosmological simulation with 1.1 million bodies reported in [22]. It is triaxial and highly concentrated toward the middle. Exact forces were computed for 4942 bodies.
- A randomly selected subset of a very large cosmological simulation with 8.8 million bodies. The model displays significant clumpiness, and contains approximately 100 identifiable "halos" (isolated regions of significantly enhanced density) with more than 30 bodies. The subset contains 32768 bodies. Exact forces were computed for 4624 bodies.

The models using the Sum Squares MAC were run with Δ_{tot} set to a constant fraction, $f = 0.01$, of the exact acceleration (which was pre-computed by direct summation). That is, the rms error computed using Sum Squares MAC is guaranteed to be less than 1% of the total acceleration on each particle. This choice of Δ_{tot} makes it easy to generate meaningful statistics about the errors. Since Δ_{tot} is a fixed fraction of the total acceleration, we expect that the total *relative* errors, i.e., the absolute errors divided by the magnitude of the exact total acceleration, incurred by each particle should be approximately equal (and within a small constant factor of f). It is important to note that the Sum Squares MAC can compute forces with arbitrary values of Δ_{tot}, which would presumably be supplied by a sophisticated integrator operating on a per-particle basis. In the absence of a sophisticated integrator, it is reasonable to set $\Delta_{tot} = f\,|\vec{a}_{last}|$, where $\vec{a}_{last}$ is the acceleration of the body computed for the previous timestep.

Figures 12 through 14 show percentiles of relative error in the acceleration for selected bodies in the test models. The parameters used in the opening criteria, and some statistics related to performance and accuracy are shown in Table 2.

MAC	Model	N_{mono}	N_{quad}	$2N_{quad} + N_{mono}$	Mean Err	Rms Err	Max Err
	18k Merger	117.3	590.0	1287	0.00117	0.00147	0.0135
BH $\theta = 0.65$	32k Cosmology	74.6	337.6	750	0.00167	0.00319	0.0701
	50k Halo	73.8	503.9	1082	0.00150	0.00203	0.0400
	18k Merger	101.1	498.4	1098	0.00112	0.00141	0.0121
MD $\theta = 1.1$	32k Cosmology	69.1	319.2	708	0.00164	0.00317	0.0662
	50k Halo	74.8	440.4	956	0.00151	0.00199	0.0224
	18k Merger	122.1	538.9	1200	0.00127	0.00165	0.0142
Bmax $\theta = 0.7$	32k Cosmology	88.6	346.7	782	0.00178	0.00340	0.0706
	50k Halo	73.1	488.4	1050	0.00152	0.00216	0.0240
	18k Merger	829.2	0	829	0.00122	0.00138	0.00455
Sum Sq $f = 0.01$	32k Cosmology	585.2	0	585	0.00132	0.00160	0.00571
(monopole only)	50k Halo	1101.1	0	1101	0.00116	0.00131	0.00466
	18k Merger	750.9	19.9	791	0.00113	0.00127	0.00453
Sum Sq $f = 0.01$	32k Cosmology	521.3	15.9	553	0.00120	0.00143	0.00536
(variable order)	50k Halo	963.4	32.1	1028	0.00105	0.00119	0.00473

Table 2: Table of statistics related to relative errors in test models. All errors are relative, i.e., normalized to the exact acceleration on each body. The normalization is applied before statistics are computed. The N_{mono} and N_{quad} columns list the mean number of the given type of interaction per body. The $2N_{quad} + N_{mono}$ column is roughly proportional to running time, assuming that quadrupole interactions are twice as expensive as monopole interactions.

Table 2 and Figures 12 through 14 confirm the hope that the Sum Square MAC outperforms the other MACs (at least on the three test cases). The curves in the figures corresponding to the Sum Square MAC are markedly flatter than the others. This implies that the errors are more narrowly distributed, which is desirable since very small errors on some bodies do not counteract the effect of large errors on other bodies. Expending additional effort to reduce a few errors is not profitable unless all errors are reduced equally. Furthermore, when one compares the errors reported in Table 2 across models, the Sum Square MAC errors are much less sensitive to the model. Setting a tolerance using the Sum Square MAC appears to imply a certain level of overall accuracy (whether measured by mean error, rms error or maximum error) independent of the distribution of bodies. In contrast, setting a value of θ in one of the traditional MACs implies an overall accuracy that is strongly dependent on the distribution of bodies.

Finally, we note that the Sum Squares MAC was run using only monopole interactions, while the traditional MACs were run with quadrupole interactions. Quadrupole interactions are about twice as costly as monopole interactions in terms of CPU usage and also require about twice as much storage [5]. Table 2 reports the quantity $2N_{quad} + N_{mono}$ which is roughly proportional to running time under the assumption that quadrupole interactions are twice as expensive as monopole. With this assumption, the Sum Squares MAC is significantly cheaper than the other MACs. Nevertheless, it achieves marginally superior average errors, and worst-case errors that are superior by factors ranging from three to ten.

5.3 What value of p.

It is tempting to add quadrupole terms to the Sum Squares Mac. Table 2 shows results for the Sum Squares MAC using the monopole approximation for partials. Computing quadrupole terms would considerably add to the expense in terms of both time and memory. Furthermore, we find that adding quadrupole interactions is less cost-effective than simply lowering the Δ_{tot} accuracy parameter, which has the effect of evaluating additional monopole interactions. The reason is that computing high-order approximations when low-order will suffice can be just as wasteful as computing small-magnitude partials to high absolute accuracy. Once one considers MACs that vary on a per-particle basis, it is natural to ask if the multipole order, p, might also vary on a per-particle, or even per-partial, basis.

In some sense, the optimal scheme is one in which the sum in Eq.(15) is optimized not just by minimizing the number of partials, but also by allowing different partials to be evaluated to different orders, and assigning an appropriate cost. This optimization problem is even harder than the original, but an approximate solution is still tractable.

Suppose that multipole moments have been stored with cells in the tree up to some order p_{max}. When we traverse the tree, we can still maintain a priority queue of unopened cells keyed by the magnitude of the possible error, but now we also store a value of p with each unopened cell. When the cell with the largest error is popped from the priority queue, we check its value of p. If it is less than p_{max}, then we simply increment p, recompute the new error appropriate for the higher order approximation and re-insert it in the queue. If p is equal to p_{max}, then we insert its children on the queue, each with $p = 1$. With this scheme, high order interactions are computed only when they are useful in reducing the total error.

We have done some limited experiments with this scheme and $p_{max} = 2$. We ran the three representative models using the Sum Squares MAC and $f = 0.01$ obtaining errors that are essentially indistinguishable from those that result from using only monopole interactions. The average number of interactions of various types that were computed for each model are shown in Table 2. Notice that the vast majority of the interactions are still monopole. The savings that can be attributed to using quadrupole interactions is at best about 7%. This saving must be weighed against the additional complexity involved in coding the quadrupole interaction, as well as the memory and time required to store the quadrupole moments themselves. Our limited experience suggests that it is not worth the effort to store and use quadrupole moments in conjunction with the Sum Squares MAC. It may

[5] These factors, of course, depend strongly on implementation details. Carefully tuned assembly language for the i860 microprocessor achieves approximately one monopole interaction in $1.5\mu sec$ and one quadrupole interaction in $3.0\mu sec$.

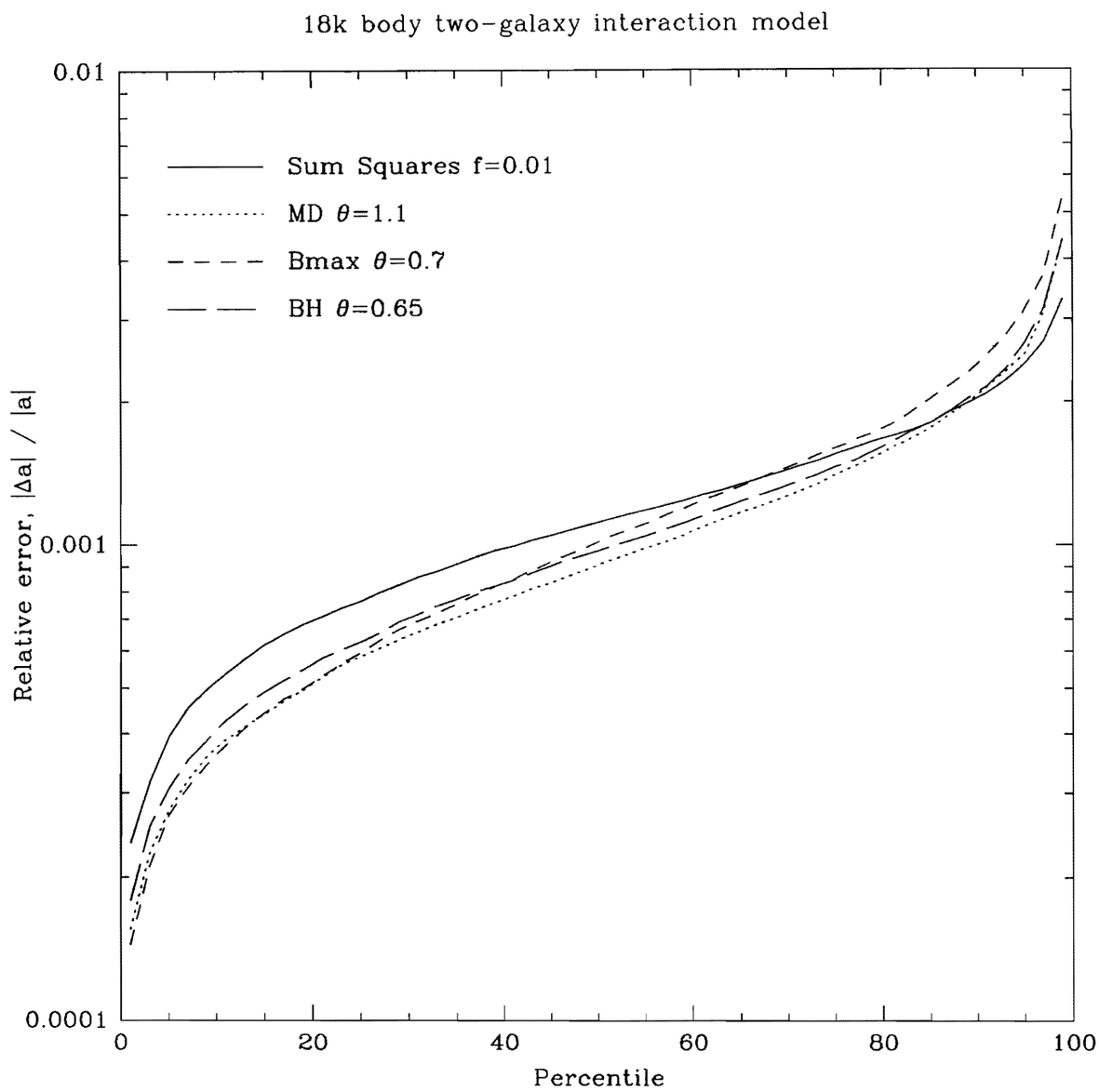

Figure 12: Percentiles of relative error in the total acceleration computed for 4443 bodies in a two-galaxy encounter with 18000 bodies.

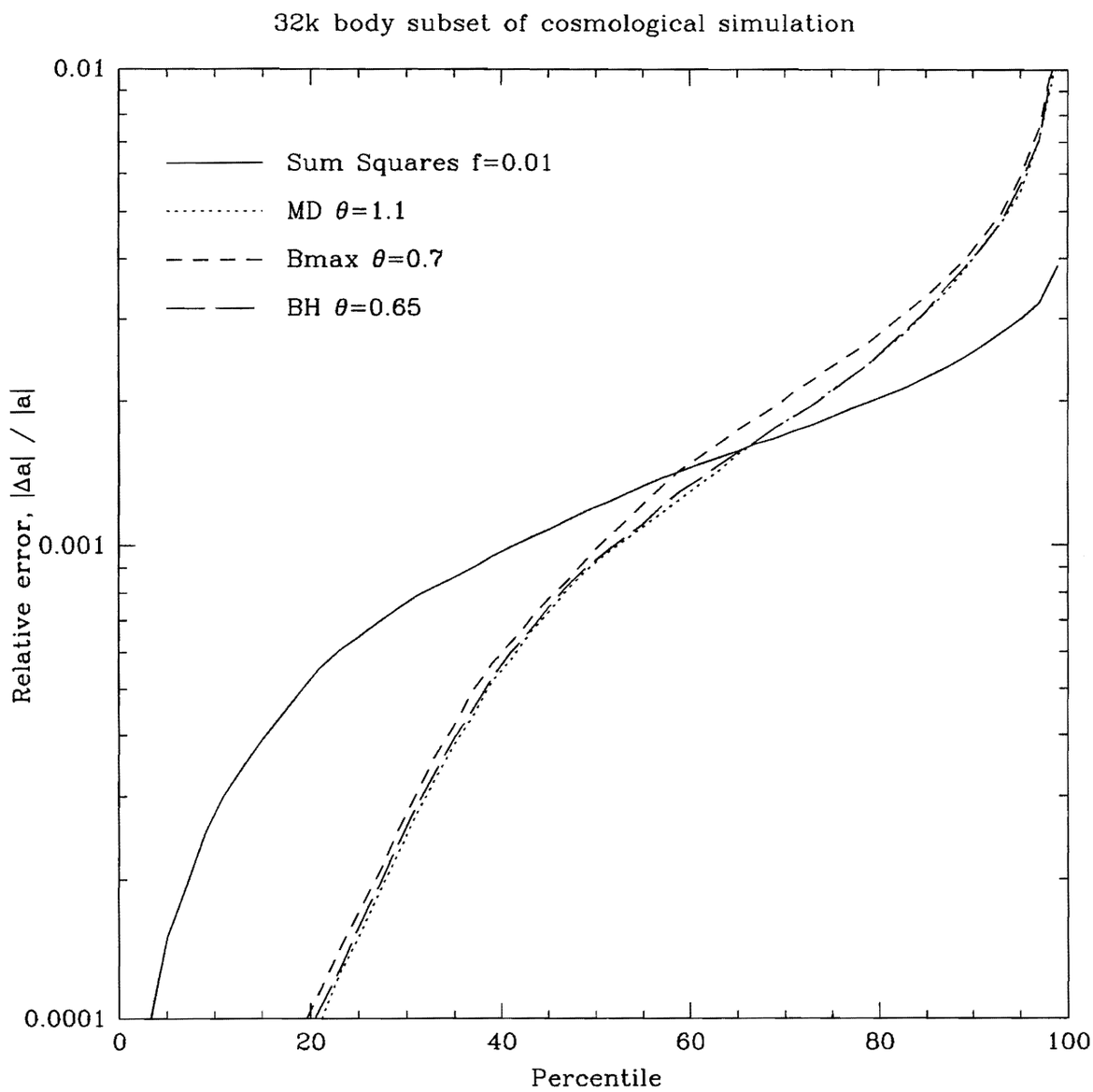

Figure 13: Percentiles of relative error in the total acceleration computed for 4624 bodies in a late-epoch cosmological simulation with 32768 bodies.

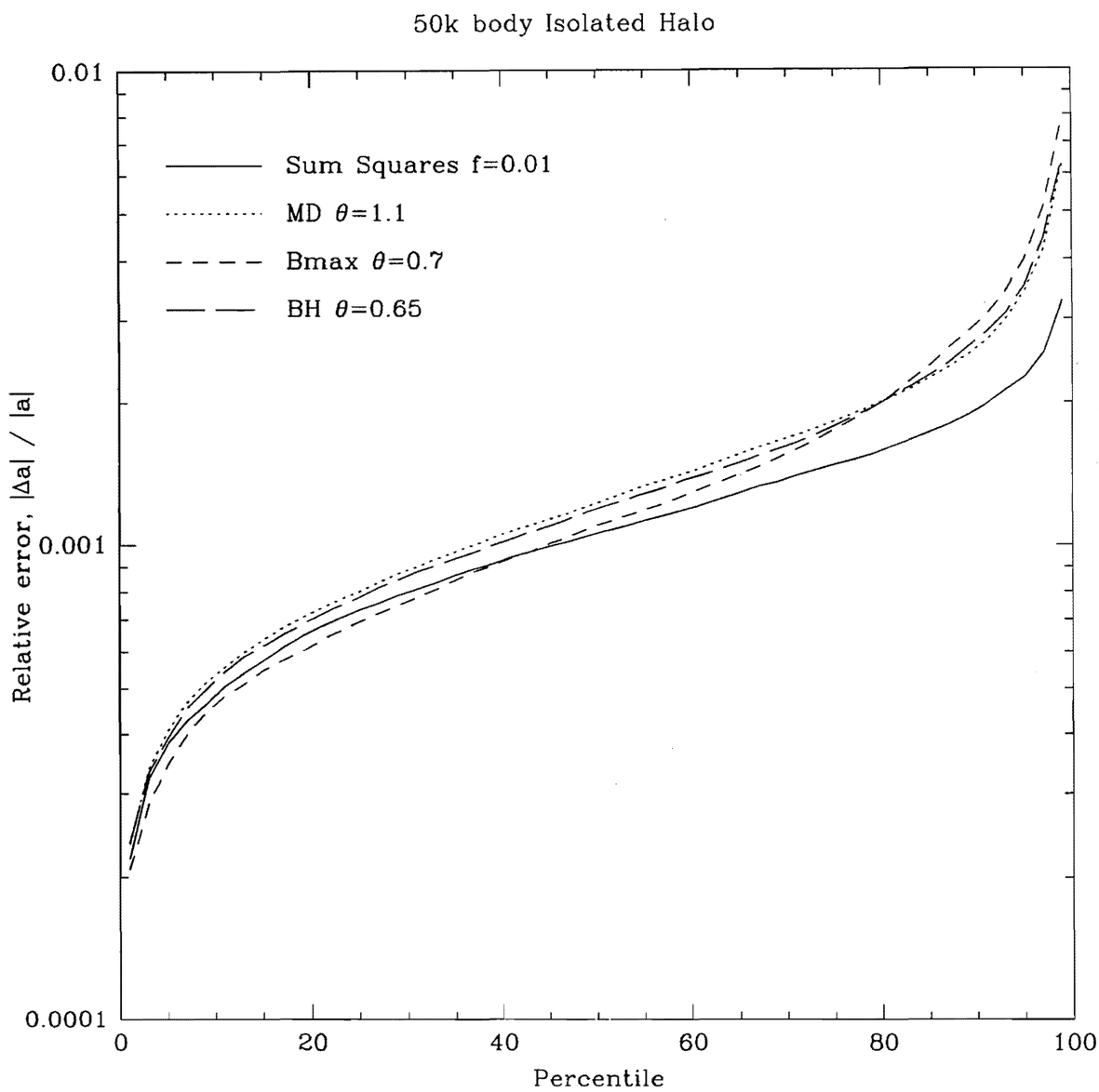

Figure 14: Percentiles of relative error in the total acceleration computed for 4942 bodies in a single triaxial halo.

be, however, that an analogy with Figure 9 is appropriate, i.e., if much higher accuracy is required, then the use of higher order multipoles may be cost effective.

6 Conclusions

We have shown that conventional treecodes are susceptible to a large and disruptive source of error when run with commonly accepted values of θ. A merger simulation with plausible initial conditions was integrated with a treecode using the BH MAC and $\theta = 0.7$, and underwent a significant and unphysical disruption.

We suggest that in light of this discovery, which comes after a number of researchers have empirically studied and used the BH MAC, empirical tests of the accuracy of treecodes are highly suspect. It is imperative that treecodes and MACs be justified on *analytic*, rather than empirical grounds. Furthermore, the analysis must consider *worst-case* behavior, now that such behavior has been observed in practice. Analyses that assume uniform distributions of matter within cells, or otherwise restrict the phase-space in which errors are sought should be considered highly suspect.

We have carried out such an analysis of the BH MAC and two other MACs which we introduce specifically to address the particular source of errors encountered in the merger example. We strongly suggest that treecode users adopt either the MD MAC or the Bmax MAC to eliminate the potential for disastrous errors. Of the three simple MACs (BH, MD, Bmax), the MD MAC appears to be best in the sense of best worst-case behavior for a given amount of CPU resources. It is superficially equivalent to the BH MAC, except that the distance from a body to a cell is computed between the body and the nearest face of the cell, rather than the center-of-mass of the cell. Our analysis suggests, however, that a disturbingly large amount of CPU resources is required to guarantee a reasonable level of accuracy, no matter which simple MAC is used.

We instead propose a new MAC based on a much tighter error bound that can be derived analytically, but that relies two additional moments of the distribution of matter within a cell. This error bound, when combined with an alternative ordering of the tree traversal, which is itself based on the error bounds, leads to the Sum Squares MAC, which outperforms all of the simple MACs, and guarantees reasonable accuracy at reasonable cost. In fact, by comparison with the $p = 2, \theta = 0.65$ BH MAC, the monopole-only Sum Squares MAC achieves empirical rms errors that range from comparable to twice as good, and empirical worst-case errors that range from three to twelve times as good. The CPU requirements range from approximately equal to one third less for the Sum Squares MAC. The theoretical worst-case error for the Sum Squares MAC under consideration is 1%, which can be guaranteed only by $p = 4$ multipoles and $\theta = 0.25$ using the BH MAC. The Sum Squares MAC achieves this performance with only monopole interactions. This results in significant savings in memory, CPU time and code complexity, which often has dramatic non-linear effects on modern RISC architectures.

The Sum Squares MAC allows much finer control over the magnitudes of errors associated with individual particles. This presents an opportunity for sophisticated integrators to request accelerations with errors specified on a per-particle basis. To our knowledge, none of the currently popular "individual timestep" integrators are so configured, so this is a technique which remains to be exploited. The fact that errors are bounded in absolute rather than relative terms may also be significant for other types of N-body simulation, notably molecular dynamics and vortex dynamics, in which the monopole contribution of a large region may be vanishingly small due to charge cancelation, but dipole and higher terms may be significant.

Acknolwedgments

We would like to the Concurrent Supercomputing Consortium and the Caltech Concurrent Supercomputing Facility for providing computational resources. JS wishes to acknowledge support from the Advanced Computing Division of the NSF, as well as the Center for Research on Parallel Computation. MW wishes to acknowledge support from IGPP and AFOSR. We are indebted to Grégoire Winckelmans for pointing out numerous typographical errors.

A MULTIPOLE MOMENTS AND MULTIPOLE EXPANSIONS.

We restrict our attention to methods which compute only Body-Cell or Body-Body interactions. The analysis of Cell-Cell interactions (and hence all $O(N)$ algorithms) is deferred to another paper.

Consider a distribution of matter in a "cell", $\mathcal{V}$, and its effect on the potential and acceleration field outside $\mathcal{V}$. In the specific case of the BH algorithm, the cells are cubical, with sides of length L, $L/2$, $L/4$, etc, but for now, these facts are irrelevant.

The potential at a point r is given by

$$\phi(r) = \int_{\mathcal{V}} G(\vec{r} - \vec{x})\rho(\vec{x})d^3x. \tag{17}$$

where ρ is the mass density of matter in the volume, and G is the Green's function, which we leave general for the moment. It is noteworthy that the multipole expansion, and hence the entire machinery of treecodes, may be formulated for arbitrary Green's functions. It may be possible to apply treecodes to systems with entirely different Green's functions, such as may arise in chemistry [23] or fluid dynamics[24, 25].

The acceleration is obtained from the gradient of the potential:

$$\vec{a}(\vec{r}) = -\nabla\phi(r). \tag{18}$$

The multipole expansion is obtained by selecting a particular point, (usually the center-of-mass of the cell), $\vec{r}_0$, and performing a Taylor expansion of:

$$\phi(r) = \int_{\mathcal{V}} G((\vec{r} - \vec{r}_0) - (\vec{x} - \vec{r}_0))\rho(\vec{x})d^3x. \tag{19}$$

The resulting formulae are:

$$\phi(r) = \sum_{n=0}^{p} \phi_{(n)}(r) + \Delta\phi_{(p)}(r) \tag{20}$$

$$\vec{a}(r) = \sum_{n=0}^{p} \vec{a}_{(n)}(r) + \Delta\vec{a}_{(p)}(r) \tag{21}$$

where

$$\phi_{(n)}(r) = \frac{(-1)^n}{n!} M_{(n)}^{i_1\ldots i_n} \partial_{i_1\ldots i_n} G|_{\vec{r}-\vec{r}_0} \tag{22}$$

$$\vec{a}_{(n)}(r) = -\frac{(-1)^n}{n!} M_{(n)}^{i_1\ldots i_n} \partial_{i_1\ldots i_n} \vec{\nabla} G|_{\vec{r}-\vec{r}_0} \tag{23}$$

The multipole moments are defined by,

$$M_{(n)}^{i_1\ldots i_n} = \int_{\mathcal{V}} d^3x (x - r_0)^{i_1}(x - r_0)^{i_2}\ldots(x - r_0)^{i_n}\rho(x), \tag{24}$$

or, in the case of a collection of point masses m_α at positions $\vec{x}_\alpha$,

$$M_{(n)}^{i_1\ldots i_n} = \sum_\alpha m_\alpha (x_\alpha - r_0)^{i_1}(x_\alpha - r_0)^{i_2}\ldots(x_\alpha - r_0)^{i_n}. \tag{25}$$

The multipole moments represent the distribution of mass within the cell, and are independent of the point, r, at which the potential or acceleration is computed. The error terms are:

$$\Delta\phi_{(p)}(r) = \int_{\mathcal{V}} d^3x K_{(p)}(r, x)\rho(x) \tag{26}$$

$$\Delta\vec{a}_{(p)}(r) = \int_{\mathcal{V}} d^3x \vec{\nabla}_r K_{(p)}(r, x)\rho(x) \tag{27}$$

where

$$K_{(p)}(r,x) = \frac{(-1)^{p+1}}{p!}(x-r_0)^{i_1}\cdots(x-r_0)^{i_{p+1}}\int_0^1 dt(1-t)^p\partial_{i_1\cdots i_{p+1}}G|_{\vec{r}-\vec{r}_0-t(\vec{x}-\vec{r}_0)} \tag{28}$$

With a particular Green's function in mind, it is possible to explicitly evaluate the derivatives and summations that appear in the above equations. For Newtonian gravity, the Green's function is

$$G(\vec{r}) = -\frac{1}{|\vec{r}|}. \tag{29}$$

The singularity at the origin of the Newtonian potential can lead to difficulties with time integration, so it is common to replace the Newtonian potential with the Plummer potential, which lacks the singularity:

$$G(\vec{r}) = -\frac{1}{(r^2+\epsilon^2)^{\frac{1}{2}}}. \tag{30}$$

Another alternative is to replace the Newtonian potential with a spline-softened potential [20]. Care must be taken when applying the multipole approximation to a spline-softened potential because the Taylor series which underlies the multipole expansion has limited radius of convergence near the points where the spline function changes form. Lacking a very careful analysis, it is probably best to use direct summation whenever there is the possibility that an interaction may be spline-softened. This is easily accomplished by inserting a simple comparison into the MAC.

We treat the more general Plummer case first. It is convenient to introduce some notation. Define the "Plummer distance", R:

$$R \equiv \left((\vec{r}-\vec{r}_0)^2+\epsilon^2\right)^{\frac{1}{2}} \tag{31}$$

the "pseudo-unit vector", $\vec{h}$:

$$\vec{h} \equiv \frac{\vec{r}-\vec{r}_0}{R} \tag{32}$$

and the "traced inner product":

$$\left\langle M_{(n)}|\delta^{(l)}h^{(m)}\right\rangle^{i_{2l+m+1}\cdots i_n} = M_{(n)}^{i_1\cdots i_n}\delta_{i_1i_2}\cdots\delta_{i_{2l-1}i_{2l}}h_{i_{2l+1}}\cdots h_{i_{2l+m}} \tag{33}$$

Then Eq.(22) and 23, with the Plummer potential, become:

$$\phi_{(n)}(r) = -\frac{1}{R^{n+1}}\sum_{m=0}^{m\le n/2}\frac{(-1)^m(2n-2m-1)!!}{m!2^m(n-2m)!}\left\langle M_{(n)}|\delta^{(m)}h^{(n-2m)}\right\rangle \tag{34}$$

and

$$\begin{aligned}\vec{a}_{(n)}(r) &= -\frac{1}{R^{n+2}}\sum_{m=0}^{m\le n/2}\frac{(-1)^m(2n-2m-1)!!}{m!2^m(n-2m)!} \\ &\times\left((2n-2m+1)\left\langle M_{(n)}|\delta^{(m)}h^{(n-2m)}\right\rangle\vec{h} - (n-2m)\left\langle M_{(n)}|\delta^{(m)}h^{(n-2m-1)}\right\rangle\right).\end{aligned} \tag{35}$$

For reference, the summations are expanded, and combinatoric factors evaluated for the first few values of n:

$$\begin{aligned}\phi_{(0)} &= -R^{-1}M_{(0)}, \\ \phi_{(1)} &= -R^{-2}\left\langle M_{(1)}|\delta^{(0)}h^{(1)}\right\rangle, \\ \phi_{(2)} &= -R^{-3}\frac{1}{2}\left(3\left\langle M_{(2)}|\delta^{(0)}h^{(2)}\right\rangle - \left\langle M_{(2)}|\delta^{(1)}h^{(0)}\right\rangle\right), \\ \phi_{(3)} &= -R^{-4}\frac{1}{6}\left(15\left\langle M_{(3)}|\delta^{(0)}h^{(3)}\right\rangle - 9\left\langle M_{(3)}|\delta^{(1)}h^{(1)}\right\rangle\right), \\ \phi_{(4)} &= -R^{-5}\frac{1}{24}\left(105\left\langle M_{(4)}|\delta^{(0)}h^{(4)}\right\rangle - 90\left\langle M_{(4)}|\delta^{(1)}h^{(2)}\right\rangle + 9\left\langle M_{(4)}|\delta^{(2)}h^{(0)}\right\rangle\right),\end{aligned} \tag{36}$$

$$
\begin{aligned}
\vec{a}_{(0)i} &= -R^{-2} M_{(0)} \vec{h}, \\
\vec{a}_{(1)i} &= -R^{-3} (3 \left\langle M_{(1)} | \delta^{(0)} h^{(1)} \right\rangle) \vec{h} - \left\langle M_{(1)} | \delta^{(0)} h^{(0)} \right\rangle), \\
\vec{a}_{(2)i} &= -R^{-4} \frac{1}{2} (15 \left\langle M_{(2)} | \delta^{(0)} h^{(2)} \right\rangle \vec{h} - 6 \left\langle M_{(2)} | \delta^{(0)} h^{(1)} \right\rangle \\
&\qquad -3 \left\langle M_{(2)} | \delta^{(1)} h^{(0)} \right\rangle \vec{h}), \\
\vec{a}_{(3)i} &= -R^{-5} \frac{1}{6} \Big(105 \left\langle M_{(3)} | \delta^{(0)} h^{(3)} \right\rangle \vec{h} - 45 \left\langle M_{(3)} | \delta^{(0)} h^{(2)} \right\rangle \\
&\qquad -45 \left\langle M_{(3)} | \delta^{(1)} h^{(1)} \right\rangle \vec{h} + 9 \left\langle M_{(3)} | \delta^{(1)} h^{(0)} \right\rangle \Big), \\
\vec{a}_{(4)i} &= -R^{-6} \frac{1}{24} \Big(945 \left\langle M_{(4)} | \delta^{(0)} h^{(4)} \right\rangle \vec{h} - 420 \left\langle M_{(4)} | \delta^{(0)} h^{(3)} \right\rangle \\
&\qquad -630 \left\langle M_{(4)} | \delta^{(1)} h^{(2)} \right\rangle \vec{h} + 180 \left\langle M_{(4)} | \delta^{(1)} h^{(1)} \right\rangle \\
&\qquad +45 \left\langle M_{(4)} | \delta^{(2)} h^{(0)} \right\rangle \vec{h} \Big).
\end{aligned} \tag{37}
$$

It is possible to set ϵ equal to zero in the above formula and obtain correct results for the Newtonian potential. However, an additional simplification is applicable whenever the Green's function is a solution of Laplace's equation. Since derivatives may be taken in any order, we have,

$$
\delta^{i_\alpha i_\beta} \partial_{i_1} \partial_{i_2} \cdots \partial_{i_n} G(r) = 0; \qquad \vec{r} \neq 0; \quad \alpha, \beta \leq n. \tag{38}
$$

We define

$$
Q_{(n)} \equiv M_{(n)} - \delta \otimes C_{(n-2)} \tag{39}
$$

where $C_{(n)}$ is any fully symmetric rank n tensor whatsoever, and the operator $\otimes$ means a symmetrized outer product, i.e.,

$$
\left(\delta \otimes A_{(n)}\right)^{i_1 \cdots i_{n+2}} = \underbrace{\delta^{i_1 i_2} A_{(n)}^{i_3 \cdots i_{n+2}} + \cdots}_{\binom{n+2}{2}\text{combinations}} \tag{40}
$$

Because of Eq.(38), the Kronecker delta that appears in Eq.(39) vanishes when contracted with the derivatives of G. Thus, from Eq.(22), 23, 38 and 39:

$$
\phi_{(n)} = \frac{(-1)^n}{n!} Q_{(n)}^{i_1 \ldots i_n} \partial_{i_1} \cdots \partial_{i_n} G|_{\vec{r} - \vec{r}_0}, \tag{41}
$$

$$
\vec{a}_{(n)} = -\frac{(-1)^n}{n!} Q_{(n)}^{i_1 \ldots i_n} \partial_{i_1 \ldots i_n} \vec{\nabla} G|_{\vec{r} - \vec{r}_0} \tag{42}
$$

Since $C_{(n)}$ is arbitrary, we may choose it so that $Q_{(n)}$, the "reduced multipole tensor", is completely trace-free, on any pair of indices, i.e.,

$$
\delta_{i_\alpha i_\beta} Q_{(n)}^{i_1 \cdots i_n} = 0 \tag{43}
$$

Then all but the leading (i.e., $m = 0$) terms in Eq.(34) and 35 vanish, and we have

$$
\phi_{(n)}(r) = -\frac{(2n-1)!!}{n!} \frac{1}{d^{n+1}} \left\langle Q_{(n)} | e^{(n)} \right\rangle, \tag{44}
$$

$$
\vec{a}_{(n)}(r) = -\frac{(2n-1)!!}{n!} \frac{1}{d^{n+2}} \left((2n+1) \left\langle Q_{(n)} | e^{(n)} \right\rangle \hat{e} - n \left\langle Q_{(n)} | e^{(n-1)} \right\rangle \right), \tag{45}
$$

where d is the distance, $|\vec{r} - \vec{r}_0|$ and $\hat{e}$ is the unit-vector in the direction $\vec{r} - \vec{r}_0$..

Since $M_{(n)}$ is symmetric under interchange of indices, and Eq.(39) is manifestly symmetric by construction, $Q_{(n)}$ is a completely trace-free symmetric tensor of rank n. The following recursion relation (which follows from taking the trace of both sides of Eq.(39)) allows one to construct $Q_{(n)}$ from $M_{(n)}$ and its traces:

$$Q_{(n)}^{(\lfloor n/2 \rfloor)} = \delta^{(\lfloor n/2 \rfloor)} M_{(n)}, \tag{46}$$

$$Q_{(n)}^{(m)} = \delta^{(m)} M_{(n)} - \frac{1}{(m+1)(2n-2m-1)} \delta \otimes Q_{(n)}^{(m+1)}, \tag{47}$$

$$Q_{(n)} = Q_{(n)}^{(0)}, \tag{48}$$

where $\delta^{(m)} M_{(n)}$ means m powers of the Kronecker delta contracted with the tensor $M_{(n)}$, or equivalently, the m-fold trace of $M_{(n)}$.

For reference, the first few tensors $Q_{(n)}$ are:

$$\begin{aligned}
Q_{(0)} &= M_{(0)}, \\
Q_{(1)} &= M_{(1)}, \\
Q_{(2)} &= M_{(2)} - \frac{1}{3}\delta \otimes \delta M_{(2)}, \\
Q_{(3)} &= M_{(3)} - \frac{1}{5}\delta \otimes \delta M_{(3)}, \\
Q_{(4)} &= M_{(4)} - \frac{1}{7}\delta \otimes \left(\delta M_{(4)} - \frac{1}{10}\delta \otimes (\delta^{(2)} M_{(4)})\right), \\
Q_{(5)} &= M_{(5)} - \frac{1}{9}\delta \otimes \left(\delta M_{(5)} - \frac{1}{14}\delta \otimes (\delta^{(2)} M_{(5)})\right), \\
Q_{(6)} &= M_{(6)} - \frac{1}{11}\delta \otimes \left(\delta M_{(6)} - \frac{1}{18}\delta \otimes \left(\delta^{(2)} M_{(6)} - \frac{1}{21}\delta \otimes (\delta^{(3)} M_{(6)})\right)\right).
\end{aligned} \tag{49}$$

A.1 The error terms.

The error term associated with the Plummer potential is expressible in terms of elementary functions and integrals [11] but the precise form is unwieldy and not terribly useful. The error derived from a Newtonian potential is somewhat more manageable. We assume, without loss of generality, that the center-of-mass, $\vec{r}_0$, is at the origin. Then

$$K_{(p)}(r,x) = -\left(\frac{(p+1)}{r}\right) \alpha^{p+1} \int_0^1 \left(\frac{r}{r_t}\right)^{p+2} (1-t)^p P_{p+1}(\mu_t) dt \tag{50}$$

where P_n is the Legendre polynomial of degree n, and

$$\begin{aligned}
\alpha &= \frac{|\vec{x}|}{r}, \\
\vec{r}_t &= \vec{r} - t\vec{x}, \\
\hat{x} &= \frac{\vec{x}}{|\vec{x}|}, \\
\hat{e}_t &= \frac{\vec{r}_t}{|\vec{r}_t|}, \\
\mu_t &= \hat{e}_t \cdot \hat{x}.
\end{aligned} \tag{51}$$

The gradient of $K_{(p)}$ is given by

$$\nabla K_{(p)}(r,x) = \frac{1}{r^2} \alpha^{p+1} (p+1) \int_0^1 dt (1-t)^p \left(\frac{r}{r_t}\right)^{p+3} \left((p+2) P_{p+1}(\mu_t) \hat{e}_t - (\hat{x} - \mu_t \hat{e}_t) P'_{p+1}(\mu_t)\right). \tag{52}$$

We can use the following inequalities to place upper bounds on the magnitudes of $K_{(p)}$ and its gradient:

$$\begin{aligned} \frac{r}{r_t} &\leq \frac{1}{1-\alpha t} \\ |P_n(\mu)| &\leq 1 \quad \text{for } |\mu| \leq 1 \\ (n+1)^2 P_n(\mu)^2 + (1-\mu^2) P_n'(\mu)^2 &\leq (n+1)^2 \quad \text{for } |\mu| \leq 1 \end{aligned} \tag{53}$$

from which we conclude

$$\left|K_{(p)}(r,x)\right| \leq \frac{1}{r}\frac{\alpha^{p+1}}{1-\alpha} \tag{54}$$

$$\left|\vec{\nabla}_r K_{(p)}(r,x)\right| \leq \frac{1}{r^2}\frac{\alpha^{p+1}}{(1-\alpha)^2}(p+2-\alpha(p+1)) \tag{55}$$

Using Eq.(55) we obtain:

$$\Delta a_{(p)}(r) \leq \frac{1}{r^2}\int_{\mathcal{V}} d^3x\, |\rho(x)| \frac{1}{(1-\alpha)^2}\left((p+2)\alpha^{p+1} - (p+1)\alpha^{p+2}\right) \tag{56}$$

Returning to the general case, where $d = |\vec{r} - \vec{r}_0|$, and using the fact that

$$\alpha \leq \frac{b_{max}}{d}, \tag{57}$$

we obtain:

$$\Delta a_{(p)}(r) \leq \frac{1}{r^2}\frac{1}{(1-\frac{b_{max}}{d})^2}\left((p+2)\left(\frac{B_{(p+1)}}{d^{p+1}}\right) - (p+1)\left(\frac{B_{(p+2)}}{d^{p+2}}\right)\right), \tag{58}$$

where the moments:

$$B_{(n)} = \int_{\mathcal{V}} d^3x\, |\rho(x)|\, |\vec{x} - \vec{r}_0|^n = \sum_\beta m_\beta\, |\vec{x}_\beta - \vec{r}_0|^n \tag{59}$$

depend solely on the distribution of matter in $\mathcal{V}$. Note that these expressions hold even in systems where the "mass-density", $\rho(x)$ is allowed to be negative. Such systems, of course, do not arise in astrophysics, but they are common in molecular-dynamics simulations with charged species.

A.2 Bounds on $B_{(n)}$

It is a straightforward, for a treecode to compute exact values of $B_{(n)}$ as needed by direct summation over all bodies within a cell. However, for large cells with many bodies this procedure can be costly. In this section we derive upper and lower bounds on $B_{(n)}$ which can be used in lieu of direct summation. Note, however, that these additional bounds do rely on the non-negativity of the density field, and hence are inapplicable in molecular dynamics simulations. In terms of upper and lower bounds on $B_{(n)}$, we have:

$$\Delta a_{(p)}(r) \leq \frac{1}{d^2}\frac{1}{(1-\frac{b_{max}}{d})^2}\left((p+2)\left(\frac{\lceil B_{(p+1)}\rceil}{d^{p+1}}\right) - (p+1)\left(\frac{\lfloor B_{(p+2)}\rfloor}{d^{p+2}}\right)\right). \tag{60}$$

Even moments may be computed exactly from traces of the unreduced multipole tensors:

$$B_{(n)} = \delta^{(n/2)} M_{(n)} \qquad \text{even } n. \tag{61}$$

Furthermore, if the maximum extent, b_{max} is known (or bounded), then upper bounds may be placed on high-order moments by:

$$B_{(n+k)} \leq b_{max}^k B_{(n)} \tag{62}$$

The following formulas, which follow from Hölder's inequality [26] are useful for even and odd p respectively:

$$B_{(p+2)} \geq \frac{B^2_{(p)}}{B_{(p-2)}} = \frac{\left(\delta^{(p/2)} M_{(p)}\right)^2}{\delta^{((p-2)/2)} M_{(p-2)}} \tag{63}$$

$$B^2_{(p+2)} \geq \frac{B^5_{(p-1)}}{B^3_{(p-3)}} = \frac{\left(\delta^{((p-1)/2)} M_{(p-1)}\right)^5}{\left(\delta^{((p-3)/2)} M_{(p-3)}\right)^3} \tag{64}$$

The particular case of $p = 1$ deserves special consideration. For this case, simple expressions involving only $B_{(0)}$ and b_{max} do not afford sufficiently tight bounds on the $B_{(2)}$ and $B_{(3)}$ moments. Thus, we recommend accumulating the diagonal elements of the $M_{(2)}$ moment exactly, even though $M_{(2)}$ will not be used in the multipole approximation, and using the following alternative bounds for $B_{(2)}$ and $B_{(3)}$:

$$B_{(2)} = \delta M_{(2)} \tag{65}$$

$$B^2_{(3)} \geq \frac{B^3_{(2)}}{B_{(0)}} = \frac{(\delta M_{(2)})^3}{M_{(0)}}. \tag{66}$$

Eq.(60) tells us the maximum possible error in terms of the moments $B_{(n)}$. Existing tree-codes, however, do not record values (or estimates) of $B_{(n)}$. Lacking any knowledge of $B_{(n)}$, we can still obtain a weak bound using:

$$B_{(p+1)} \leq b^{p+1} M_{(0)}, \tag{67}$$

$$B_{(p+2)} \geq 0, \tag{68}$$

from which we derive

$$\left|\vec{A}_{(p)}\right|(r) \leq \left|\vec{a}_{(0)}\right| \frac{\left(\frac{b_{max}}{r}\right)^{p+1}}{\left(1 - \frac{b_{max}}{r}\right)^2} \left(p + 2 - \frac{b_{max}}{r}(p+1)\right). \tag{69}$$

The maximum error is thus a monotonically increasing function of b_{max}/r, which is useful primarily for motivating the Bmax MAC.

References

[1] A. W. Appel. An efficient program for many-body simulation. *SIAM J. Sci. Stat. Comp.*, **6**, 85, (1985).

[2] J. G. Jernigan and D. H. Porter. A tree code with logarithmic reduction of force terms, hierarchical regularization of all variables and explicit accuracy controls. *Ap. J. Suppl.*, **71**, 871, (1989).

[3] J. E. Barnes and P. Hut. A hierarchical O(NlogN) force-calculation algorithm. *Nature*, **324**, 446, (1986).

[4] L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems.* PhD thesis, Yale University, (1987).

[5] L. Greengard and V. I. Rokhlin. A fast algorithm for particle simulations. *J. Comp. Phys.*, **73**, 325, (1987).

[6] F. Zhao. An O(N) algorithm for three-dimensional N-body simulations. Master's thesis, Massachusetts Institute of Technology, (1987).

[7] W. Benz, R. L. Bowers, A. G. W. Cameron, and W. H. Press. Dynamic mass exchange in doubly degenerate binaries i. 0.9 and 1.2 msolar stars. *Ap. J.*, **348**, 647, (1990).

[8] C. R. Anderson. An implementation of the fast multipole method without multipoles. *SIAM J. Sci. Stat. Comp.*, **13**, 923, (1992).

[9] J. Makino and P. Hut. Performance analysis of direct N-body simulations. *Ap. J. Suppl.*, **68**, 833, (1988).

[10] J. Makino. Comparison of two different tree algorithms. *J. Comp. Phys.*, **88**, 393, (1990).

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

[12] J. E. Barnes and P. Hut. Error analysis of a tree code. *Ap. J. Suppl.*, **70**, 389, (1989).

[13] L. Hernquist. Performance characteristics of tree codes. *Ap. J. Suppl.*, **64**, 715, (1987).

[14] 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*, edited by R. Wielen, (Springer-Verlag, New York, 1990), p.216.

[15] J. E. Barnes. A modified tree code: Don't laugh; it runs. *J. Comp. Phys.*, **87**, 161, (1990).

[16] F. R. Bouchet and L. Hernquist. Cosmological simulations using the hierarchical tree method. *Ap. J. Suppl.*, **68**, 521, (1988).

[17] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing.* (Cambrigde University Press, New York, 1988).

[18] S. J. Aarseth. Direct methods for N-body simulations. In *Multiple Time Scales*, edited by J. U. Brackbill and B. I.Cohen, (Academic Press, New York, 1985), p.377.

[19] W. H. Press. Techniques and tricks for N-body computation. In *The Use of Supercomputers in Stellar Dynamics*, edited by P. Hut and S. McMillan, (Springer-Verlag, New York, 1986), p.184.

[20] L. Hernquist and N. Katz. TREESPH - a unification of SPH with the hierarchical tree method. *Ap. J. Suppl.*, **70**, 419, (1989).

[21] D. E. Knuth. *The Art of Computer Programming: Sorting and Searching*, volume 3. (Addison Wesley, Reading, Mass., 1973).

[22] 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. *Ap. J.*, (1992). (accepted for publication).

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

[24] A. Leonard. Computing three-dimensional incompressible flows with vortex elements. *Annual Rev. Fluid Mech.*, **17**, 523, (1985).

[25] G. S. Winckelmans. *Topics in Vortex Methods for the Computation of Three- and Two- Dimensional Incompressible Unsteady Flows.* PhD thesis, California Institute of Technology, (1989).

[26] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions.* (Dover Publications, New York, 1972).