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

We consider treecodes (N-body programs which use a tree data structure) from the standpoint of their worst-case behavior. That is, we derive upper bounds on the largest possible errors that are introduced into a calculation by use of various *multipole acceptability criteria* (MAC). We find that the conventional Barnes-Hut MAC can introduce potentially unbounded errors unless $\theta < 1/\sqrt{3}$, and that this behavior while rare, is demonstrable in astrophysically reasonable examples. We consider two other MACs closely related to the BH MAC. While they don't admit the same unbounded errors, they nevertheless require extraordinary amounts of CPU time to guarantee modest levels of accuracy.

We derive new error bounds based on some additional, easily computed moments of the mass distribution. These error bounds form the basis for four new MACs which can be used to limit the absolute or relative error introduced by each multipole evaluation, or, with the introduction of some additional data structures, the absolute or rms error in the final acceleration of each particle. Using the Sum Squares MAC to analytically place a 1% bound the rms error in a series of test models, we find that it significantly outperforms the $\theta = 0.65$ BH MAC in terms of both accuracy (mean, rms and maximum error) and performance (floating point operation count).

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

Submitted to: *Journal of Computational Physics*, July 31, 1992; revised, March 10, 1993

[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. Similar, so-called "all-pairs" problems involving the mutual interactions of N objects arise in molecular dynamics with charged species and in computational fluid dynamics. Until recently, it was believed that such problems required time proportional to N^2, because the long-range nature of the gravitational interaction (and the analogous Coulomb interaction and Biot-Savart law) requires that one consider the contribution of $N - 1$ terms in the update of 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 approximate interactions between all N bodies in a system in time proportional to $N \log N$ or even N. This qualitative change in the time-complexity has led to simulations much larger than were possible with direct-summation algorithms [9].

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 may 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 is satisfied, 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. The remainder of this paper is devoted to the analysis and optimization of various

MACs.

In Sec. 2, we introduce three simple MACs that could be easily adapted to work with almost any treecode. In Sec. 3 we argue that the MACs should be judged on the basis of worst-case (rather than mean, rms or median) accuracy, and we compare the three simple MACs in terms of performance at a specified level of accuracy. In Sec. 4 we introduce three new MACs which rely on additional "moments" of the mass distribution in a cell, and which delivers much tighter error-bounds than the three simple MACs. These can be used to guarantee limits on the absolute or relative errors introduced by each interaction, or limits on the total error. The "Sum Squares MAC" delivers a guaranteed level of accuracy but its performance is difficult to estimate *a priori*. We compare the Sum Squares MAC and the three simple MACs in three separate test cases and find that it not only guarantees far better accuracy, but it delivers superior average performance as well. Appendix A provides details of an example problem which behaves far differently from what one would expect based on previous semi-empirical treatments of errors in treecodes. Appendix B contains detailed derivations of the multipole expansions and error bounds used in the rest of the paper.

2 THREE SIMPLE MACS

2.1 The Barnes-Hut MAC

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 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.

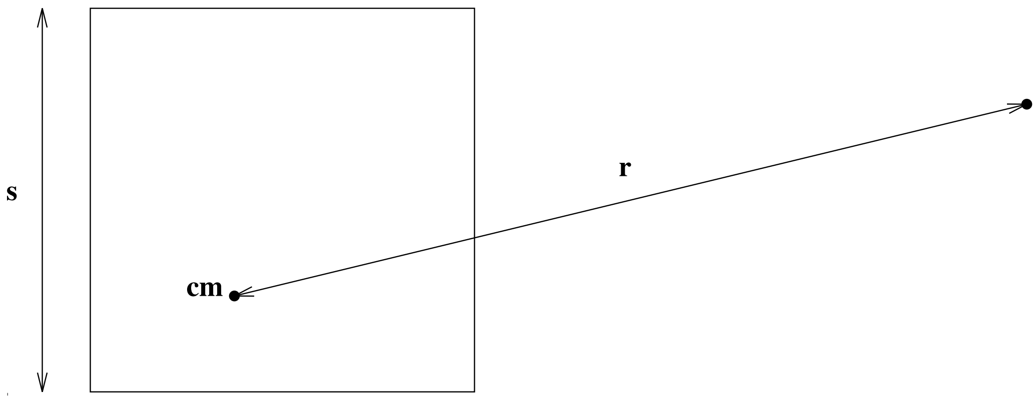

Figure 1: Geometry for BH MAC test. The multipole approximation is acceptable if and only if $s/r < \theta$.

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, $0.7 \le \theta \le 1.0$, and the multipole approximation is usually terminated after the quadrupole term. However, Appendix B shows that it is a simple matter to extend the multipole approximation to arbitrarily high order.

2.2 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. In Appendix A we find that the BH MAC can introduce surprisingly large errors which arise when a particle is near the edge of a cell, but far from its center-of-mass. This 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 2.

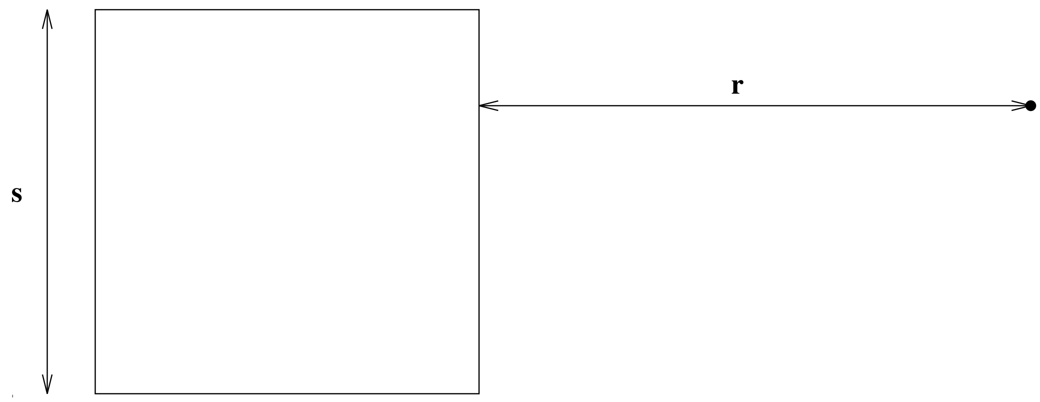

Figure 2: Geometry of for the MD MAC criterion. The multipole approximation is acceptable if and only if $s/r < \theta$.

One useful 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, 12], where one might wish to evaluate the MAC for cells whose data reside in another processor's memory.

2.3 The Bmax MAC.

Another MAC is motivated by Eq.(70), 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 3, allows that the multipole approximation is acceptable iff $b_{max}/r < \theta$. Like the MD MAC, it also eliminates the source of error that gives rise to the detonating galaxy in Appendix A.

3 Analysis of simple MACs

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. The ultimate goal is not controversial: one's simulations should run as fast as possible subject to the constraint that errors are not introduced which adversely affect the "physics." In practice, the question of how large an error is acceptable is a difficult one. We propose to

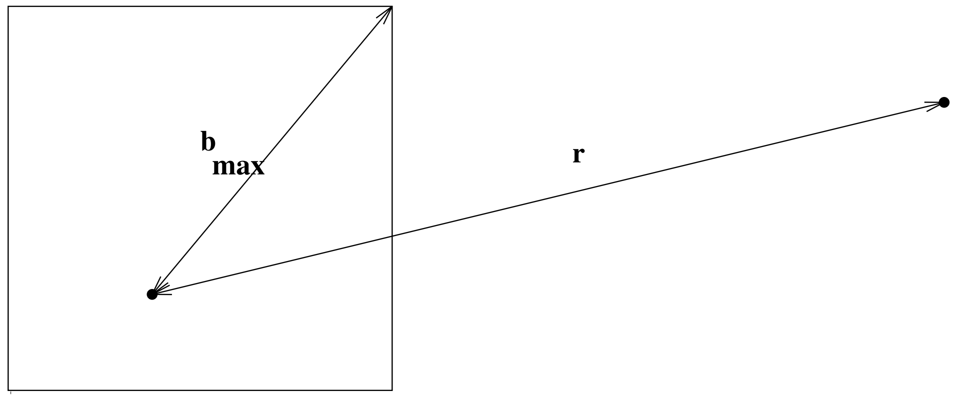

Figure 3: Geometry of for the Bmax MAC. The multipole approximation is acceptable if and only if $b_{max}/r < \theta$.

avoid the issue and leave it up to the "user" who is clearly better qualified to assess when errors of a given magnitude are acceptable and when are not. We cannot avoid the issue entirely, however, because we must determine the *kinds* of errors we are talking about.

The consensus in astrophysics [13, 14, 10] has been that it is acceptable to measure, e.g., rms errors, in a very small number of model systems. The measured errors are deemed "typical", and further simulations can be performed using the MAC and values of θ and p that have been certified by this semi-empirical validation. In contrast, the applied mathematics community prefers to rely on rigorous error bounds [4, 8], despite the fact that adherence to rigorous bounds has lead to extremely expensive methods, at least compared to the empirically "acceptable" MACs in use in astrophysics (but see Section 4). We are firmly on the side of the applied mathematicians on this question. We find the model presented in Appendix A to be compelling evidence that it is far too easy to be misled by semi-empirical methods. If a detonating galaxy like that in Appendix A can slip through the cracks of semi-empirical validation, then the sieve is leaky indeed.

We believe that 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. 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. not to mention examples from other disciplines where treecodes show promise such as molecular dynamics and fluid mechanics. 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 domain covered by the suite? The failure of previous empirical studies to detect a situation as simple as Figure 12 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 4, where we discuss an economical and mathematically defensible alternative MAC.

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 Appendix A 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 Appendix A requires just such a conspiracy, and it constitutes an entirely reasonable intermediate state of an interesting simulation.

3.1 Worst-case error analysis

We shall be computing dimensionless relative errors 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 force 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 4 and analytically by Eq.(2). The relationship between the values of x_1, x_2 and r depends on which MAC is under consideration,

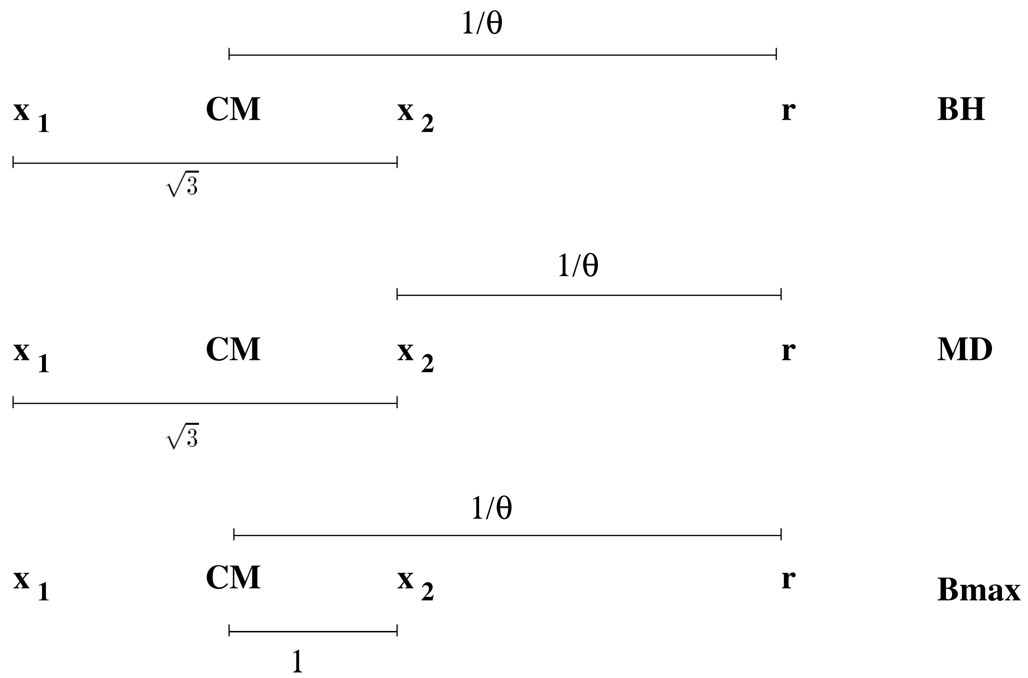

Figure 4: Relationships between positions of two point masses inside cell (at x_1 and x_2), and the point at which the acceleration is measured, r, for the three MACs.

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 $0 \le m_1 \le 1$ that gives rise to the largest possible value of e. Simple numerical techniques using parabolic interpolation [15] provide us with a maximum possible relative error for any (θ, p) pair and any MAC. The results are shown in Figure 5.

The errors displayed in Figure 5 are an order of magnitude or more worse than the empirical results obtained by a number of authors [13, 16, 14, 10]. We may account for this discrepancy by noting that the other authors report an rms or an average error. Errors approaching the bounds set by Figure 5 are atypical but not impossible.

Inspection of Figure 5 reveals that even modest accuracy requires either a very low value 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 Appendix A. 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 $\theta < 0.42$.

3.2 Performance considerations

Figure 5 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 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:

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

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

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

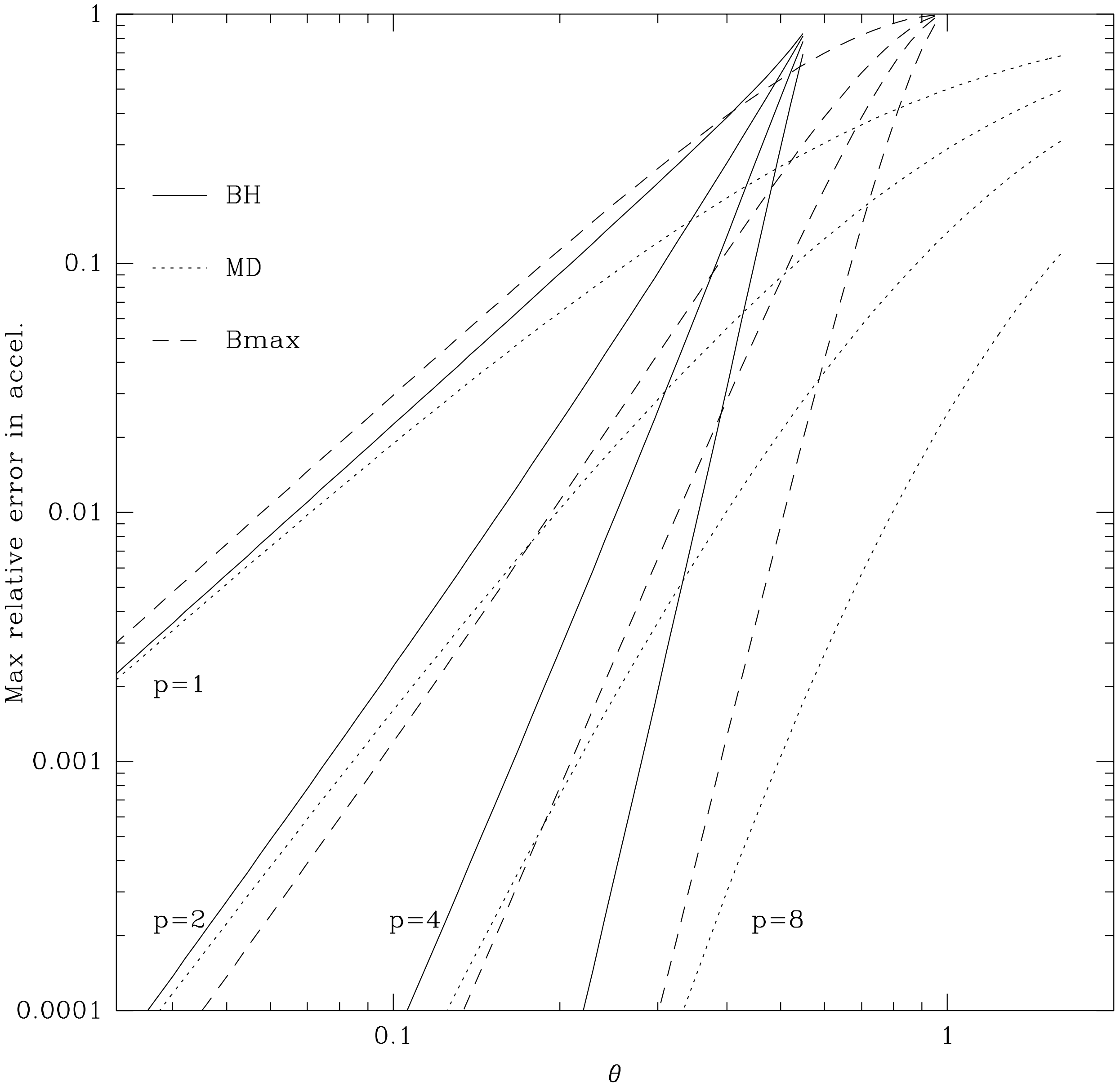

Figure 5: Plot of maximum fractional error in the acceleration, as a function of θ for values of the multipole order, $p = 1$, 2, 4 and 8, and the BH, MD and Bmax MACs. The different values of p may be distinguished on the plot by their asymptotic slope at small θ, for which $e \propto \theta^{(p+1)}$. 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.

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 and may also depend on such details as how recently nearby pieces of memory have been accessed. 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 roughly 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:[1]

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

Figure 6 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. Figure 6 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 6 to determine the optimal MAC and value of p, and finally, use Figure 5 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 6 tells us that the $p = 2$ and $p = 4$ MACs are approximately equivalent. Inspection of Figure 5 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 6 this would be about twice as expensive.

Figure 6 however, must be used with great caution. Eq.(6), (7) and (8) which are used to generate the abscissa of Figure 6 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_{max}^3) depends on the distribution of bodies in the simulation. Rather than treat the details of Figure 6 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 6. Then the procedure outlined above may be carried through with greater confidence.

4 A Data Dependent MAC

The MD MAC introduced in Section 2 is arguably superior to the BH MAC. In provides comparable worst-case errors at somewhat improved performance and it does not accept the particular configuration that gives rise to the detonating galaxy of

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

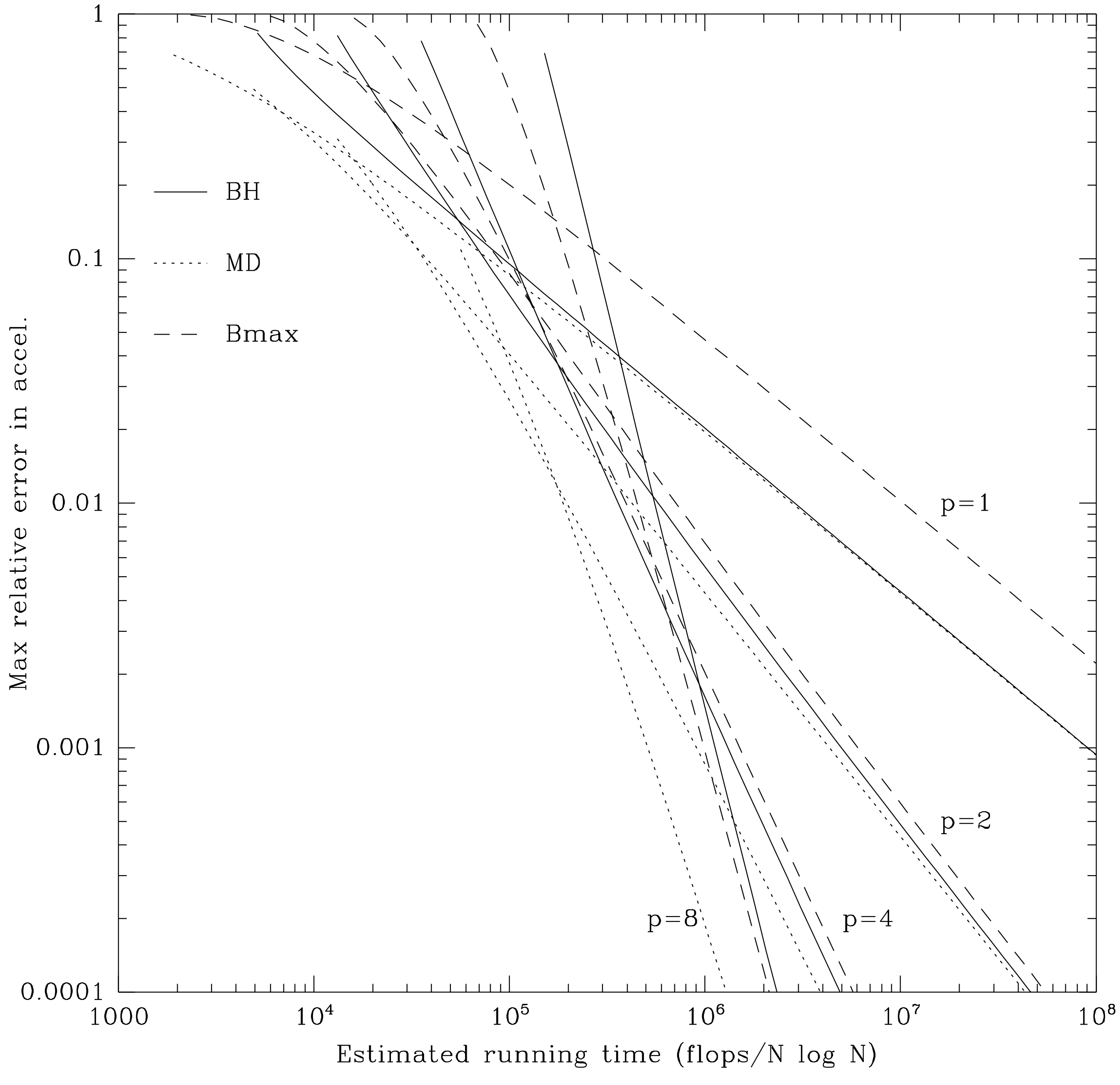

Figure 6: Plot of maximum fractional error in the force, as a function of computation time, estimated as $V_{MAC}(\theta) * f(p)$, for the BH, MD and Bmax MACs, and multipole orders, p=1, 2, 4 and 8. $V_{MAC}(\theta)$ and $f(p)$ are discussed in the text. The different values of p may be distinguished 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.

Appendix A. Nevertheless, Figures 5 and 6 imply that all three simple MACs require a great deal of computational resources to guarantee modest levels of accuracy. Accuracy of a few percent can only be guaranteed with values of θ and p that imply about 30 times as much computation as is currently expended with $p = 2, \theta = 0.75$. The observation that one cannot guarantee modest accuracy without an extraordinarily small θ is a disturbing one. Despite the dangers, one is tempted to return to a reliance on empirical studies which make clear that the worst-case behavior is a fairly rare event. If one were forced to use, e.g., $\theta = 0.25$, fewer simulations would be possible, and ultimately, less would be learned. There is another alternative, however. If we can find a tight, easily computed, analytic bound on the errors introduced by the multipole approximation, then we can simply compute the bound each time we evaluate a MAC and compare the bound with a desired accuracy. This procedure is applicable no matter what flavor of Body-Cell tree is in use [3, 2, 7]. In fact, with some additional analysis it is even applicable to adaptive Cell-Cell, i.e., O(N), treecodes.

To prevent confusion, and to avoid long-winded repetition, let us define some terminology. We make no claim that this terminology is in general use; 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 Body-Cell 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 Appendix B.

4.1 Absolute and Relative MACs

In Appendix B we show that the maximum possible partial error after p multipole terms have been added is expressible in terms of moments of the absolute value of mass distribution inside the cell. Eq.(62), which gives the maximum possible absolute error in a partial acceleration after p multipole terms have been added, is reproduced here:

$$\Delta a_{(p)}(r) \le \frac{1}{d^2} \frac{1}{(1 - \frac{b_{max}}{d})^2} \left((p+2) \left(\frac{\lceil B_{(p+1)} \rceil}{d^{p+1}} \right) - (p+1) \left(\frac{\lfloor B_{(p+2)} \rfloor}{d^{p+2}} \right) \right). \tag{9}$$

The moments, $B_{(n)}$ are defined in Eq.(61), and $d = |\vec{r} - \vec{r}_0|$ is the distance from r to the origin of the multipole expansion.

The obvious application of Eq.(9) is in the form of an "Absolute MAC", whereby the multipole approximation is acceptable if and only if

$$\lceil \Delta a_{partial} \rceil \le \Delta_{partial}, \tag{10}$$

where $\lceil \Delta a_{partial} \rceil$ is given by the right-hand-side of Eq.(9), and $\Delta_{partial}$ is a user-supplied parameter. Choice of $\Delta_{partial}$ is left to the discretion of the "user."

This formulation is extremely flexible. It is entirely possible for $\Delta_{partial}$ to vary from one particle to the next. One can use this to fine-tune the errors associated with different bodies if external considerations make it important to achieve high accuracy on only a subset of bodies, e.g., if an individual-timestep integrator is in use [17, 18, 19]. One can also achieve control of relative errors rather than absolute errors. There are two possibilities. If one has access to an estimate of the "exact" acceleration (partial or total), perhaps from the previous timestep, then one can simply arrange that $\Delta_{partial}$ be set to some chosen fraction of the magnitude of the exact acceleration. Alternatively, one can use the monopole approximation as an estimate of the exact acceleration, which leads to the "Relative MAC":

$$\frac{\Delta a_{(p)}(r)}{|a_{(0)}(r)|} \le \frac{1}{B_{(0)}} \frac{1}{(1 - \frac{b_{max}}{d})^2} \left((p+2) \left(\frac{\lceil B_{(p+1)} \rceil}{d^{p+1}} \right) - (p+1) \left(\frac{\lfloor B_{(p+2)} \rfloor}{d^{p+2}} \right) \right). \tag{11}$$

The Absolute and Relative MACs depend critically on the error bound set by Eq.(9). Since it is a strict analytic bound, 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 devoted to reducing an upper bound which is rarely approached in practice, or, on the other hand, the possibility of occasionally making unacceptably large errors. Figure 7 shows percentiles of the ratio of partial errors to the bound set by Eq.(9) for partials which were "accepted" by the Sum Square MAC in our 32768 body test model (see Table 1). Very small errors indicate that Eq.(9) is not providing a tight bound on 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 7 are large. The figure demonstrates that the bound set by Eq.(9) is very tight. In the mean, the actual error is about 20% of the bound, with occasional partials approaching the bound very closely.

Hernquist et al.[20] have asserted that the physics is best preserved when one controls relative partial errors. Their argument rests on the flawed assumption that errors are random and cancel one another out. While this is true of the N *terms* that give rise to the total acceleration when using the low-precision arithmetic in the GRAPE processor [21], it is certainly not true of the $O(\log N)$ partials that are added together in a treecode. One erroneous partial in a treecode can introduce errors far in excess of the unavoidable discreteness noise inherent in N-body simulations (as is clear from the detonating galaxy in Appendix A). We believe that it is far more "physical" to impose an upper bound on the *total error* rather than the partial errors. The correct conclusion from Hernquist et al.[20] is that, in the modeling of dissipationless systems, it is a waste of time to place a bound on the total error which is significantly smaller than the discreteness errors.

One could argue that a bound on the total error follows from limiting the relative error of each acceptable partial. This is effectively the strategy employed by the three simple MACs and the Relative MAC. 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 application of a MAC that limits relative partial errors 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 8 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 relative accuracy, e.g., 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, e.g., large θ, then one runs the risk of a single dominant interaction being computed with low accuracy. The Absolute MAC does not suffer from this dilemma. All partials are evaluated with an error below a fixed, prescribed tolerance. When added together, it is impossible for one outlying partial to dominate the total error. For this reason, we feel that the Absolute MAC is superior to the other four MACs discussed so far.

4.2 Optimizations

As written, the Absolute MAC requires evaluation of a fairly complicated and costly expression (Eq.(9)) each time it is called. It is likely that techniques like those in [22] can be used to greatly reduce the number of times the MAC is called, at the expense of a few additional interaction evaluations.

An additional optimization is extremely powerful if one can arrange that the same value of $\Delta_{partial}$ is used for every body. Then Eq.(9) implies the existence of d_{cut} satisfying:

$$\Delta a_{(p)}(r) \leq \Delta_{partial} \quad \text{iff} \quad d^2 > d^2_{cut} \tag{12}$$

When expressed in terms of d_{cut}, the Absolute MAC is just as fast as any of the "simple" MACs, yet by careful choice of d_{cut} it delivers a guaranteed level of accuracy. It is, of course, necessary to compute d_{cut} for each cell by solving Eq.(9). This is done in the tree-building phase of the computation, and is a negligible expense.

4.3 Sum MACS

The Absolute MAC, however beg the question of an appropriate value for $\Delta_{partial}$. It is conveniently left up to the "user". This is not entirely acceptable. After all, the value of $\Delta_{partial}$ depends not only on external numerical considerations (how large an error is acceptable), but also on how how many partials will contribute to the total acceleration (which itself depends

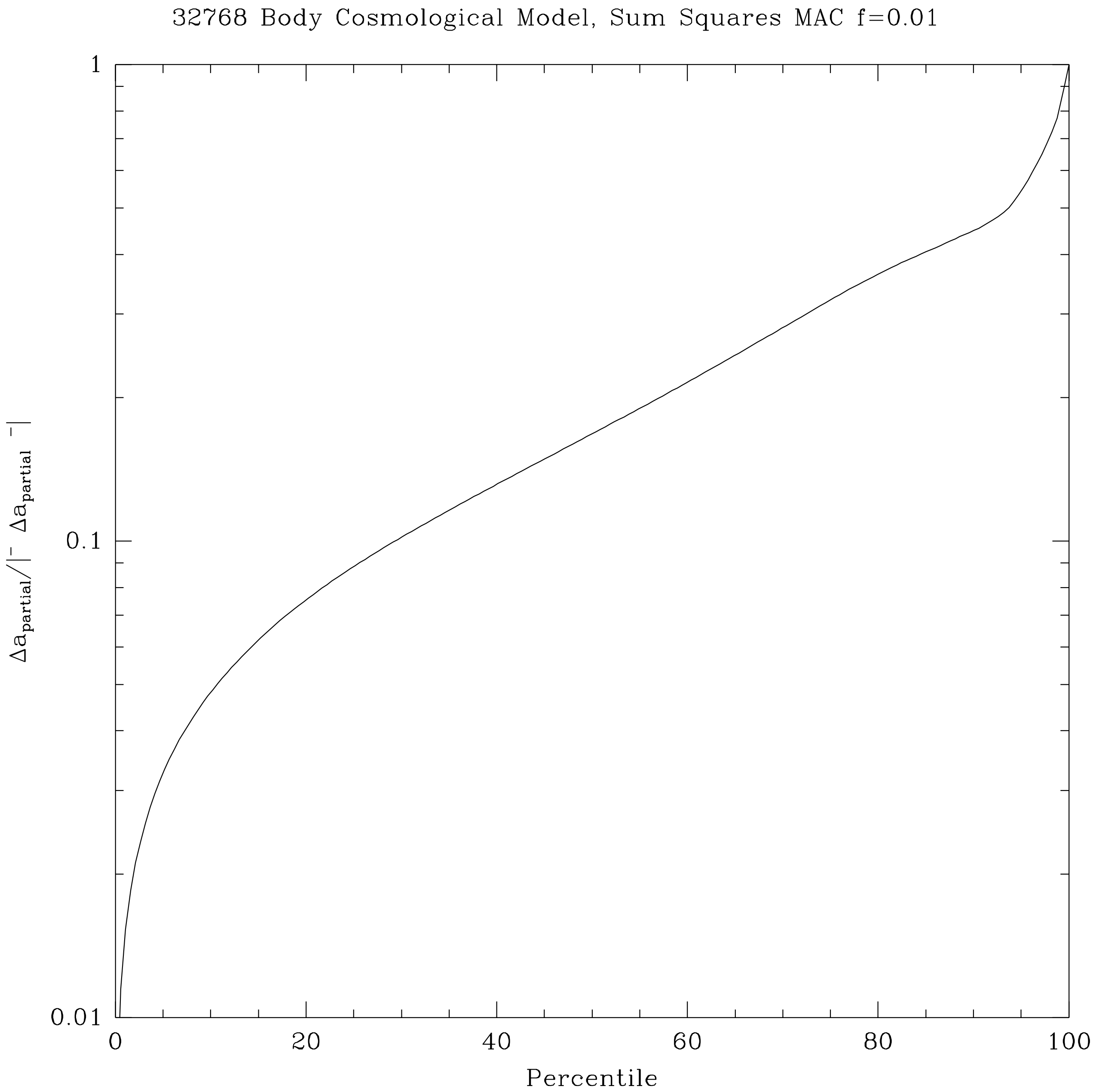

Figure 7: Percentiles of $|a_{partial}| \,/\, \lceil a_{partial} \rceil$ for the cosmological test model. The data consists of only those partial interactions which were "accepted" by the Sum Squares MAC with $f = 0.01$. The errors result from using the monopole approximation, and the bound is computed from Eq.(9) evaluated at $(p = 1)$. The mean ratio in this sample is 0.22. The rms ratio is 0.28.

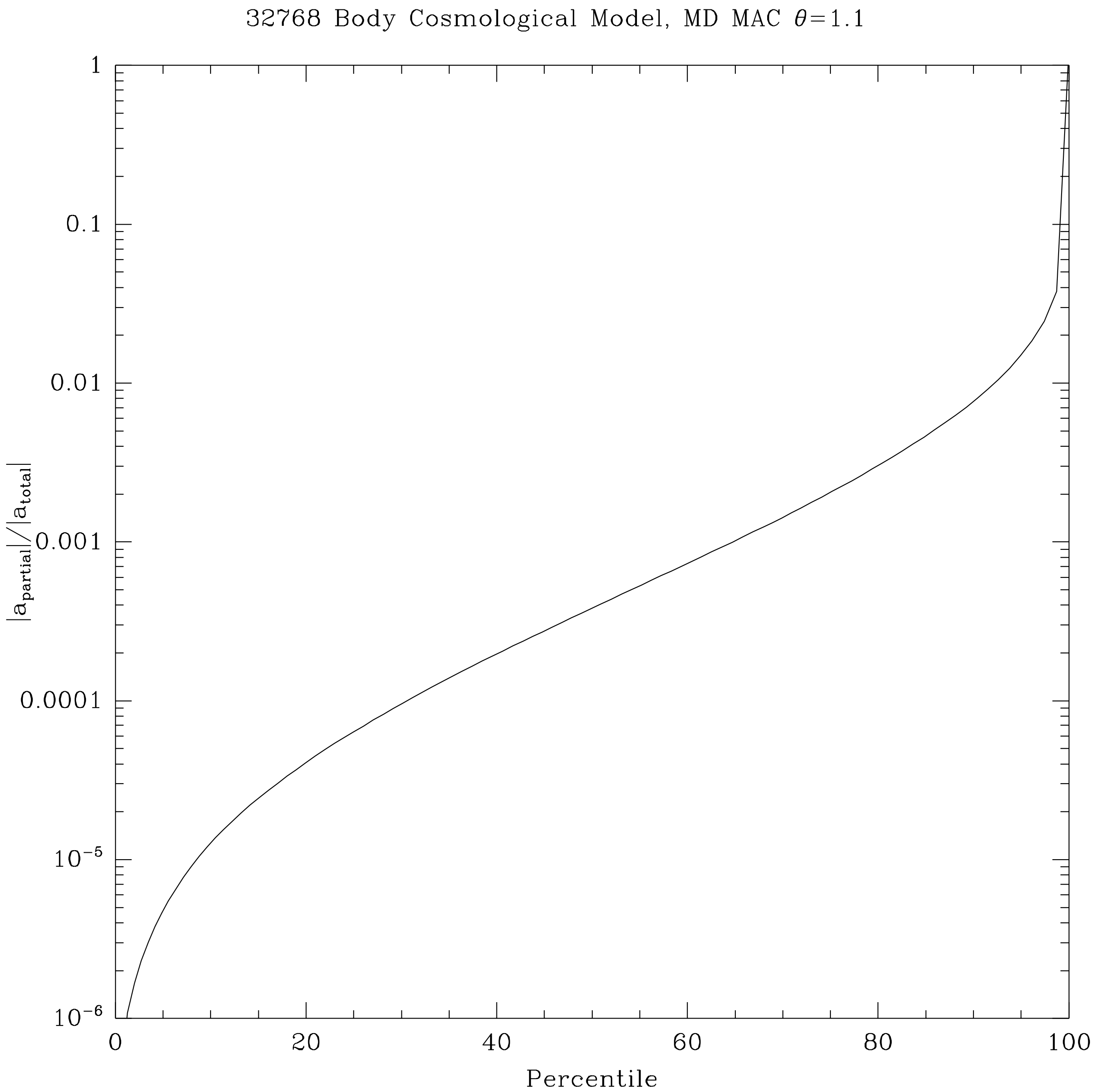

Figure 8: Percentiles of $|a_{partial}| / |a_{total}|$ for approximately 400000 partial accelerations computed using the MD MAC with $\theta = 1.1$ in a 32768 body system representing an evolved, clumpy cosmological simulation. The parameters of the system are described in Table 1.

on $\Delta_{partial}$). We need to develop a MAC that analytically guarantee:

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

The tolerance, Δ_{tot} has units of acceleration, and is determined by "physical" considerations beyond the scope of the treecode, i.e., questions of discreteness noise, integration method and the underlying physical system. It is properly in the domain of the "user", without requiring any intimate knowledge of the mechanics of the treecode. It is important to note that as far as the MAC is concerned, Δ_{tot} is a completely arbitrary value which would presumably be supplied by a sophisticated integrator operating on a per-particle basis. In the absence of a sophisticated integrator, we set $\Delta_{tot} = f\,|\vec{a}_{last}|$, where $\vec{a}_{last}$ is the acceleration of the body computed for the previous timestep. Alternatively, one can observe that discreteness noise in dissipationless systems scales as $N^{-1/2}$ [20], which suggests $\Delta_{tot} \propto N^{-1/2}$. The correct constant of proportionality must be governed by other physical/numerical considerations.

We begin with the inequality:

$$\Delta a_{total} \leq \sum_{partial} \lceil \Delta a_{partial} \rceil, \tag{14}$$

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

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

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

$$rms(a_{total}) \leq \left(\sum_{partial} \lceil \Delta a_{partial} \rceil^2 \right)^{\frac{1}{2}}, \tag{16}$$

which leads to the "Sum Squares MAC":

$$\sum_{partial} \lceil \Delta a_{partial} \rceil^2 \leq \Delta_{tot}^2. \tag{17}$$

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, and as such, it is exceedingly hard to solve exactly. The constraints are that the partials that are selected must account for every body exactly once and that the Eq.(15) or (17) be satisfied. The quantity to optimize is the number of partials. 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. [2] 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.(9)). 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, it is removed from the priority queue and values of $\lceil \Delta a_{partial} \rceil$ for its children are computed and inserted back into the priority queue. This process continues until Eq.(15) or Eq.(17) is satisfied. [3]

4.4 Optimizations

There are a number of possible ways to implement priority queues [23]. 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

[2] a priority queue [23] is a data structure into which objects may be inserted, and from which it is easy to extract the largest (or smallest) element.

[3] Minor difficulties arise because $\lceil \Delta a_{partial} \rceil$ is infinite when the body and cell are closer than b_{max}. Since one is keeping a running total of errors, it is obviously necessary to treat such situations separately.

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 with priority queues, the set of partials that remains satisfies a simple criterion:

$$\lceil \Delta a_{partial} \rceil \leq \Delta_{max} \tag{18}$$

where Δ_{max} is the largest key in the queue. Clearly, this is just 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 should not 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. If desired, Eq.(9) can be used to provide an *a posteriori* check of the sum square errors when the Absolute MAC is in use.

4.5 Numerical tests.

We have already expressed our skepticism of empirical numerical tests of MACs. However, despite the fact that a positive result from an empirical test cannot be relied upon as a certification of correctness, 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 plotting, for each MAC, 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. Cumulative probability plots convey far more information about possible large-error "tails" in the distribution than simple statistics like rms, mean, etc. We have restricted ourselves to relatively modest numbers of bodies in the test cases because it is important to be able to calculate exact accelerations for comparison with the approximations produced by the treecode. In addition, in each system we compute the accelerations exactly on only a subset of the bodies. Calculating the exact acceleration 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 A. Exact accelerations 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 [24]. It is triaxial and highly concentrated toward the middle. Exact accelerations 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 accelerations 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 the 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).

Figures 9 through 11 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 1.

Table 1 and Figures 9 through 11 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

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 1: 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.

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 1 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 simple 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 simple 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 [4]. Table 1 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.

4.6 *What value of p.*

It is tempting to add quadrupole terms to the Sum Squares Mac. Table 1 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.(17) is optimized not just by minimizing the number

[4] 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 $3.0\mu sec$.

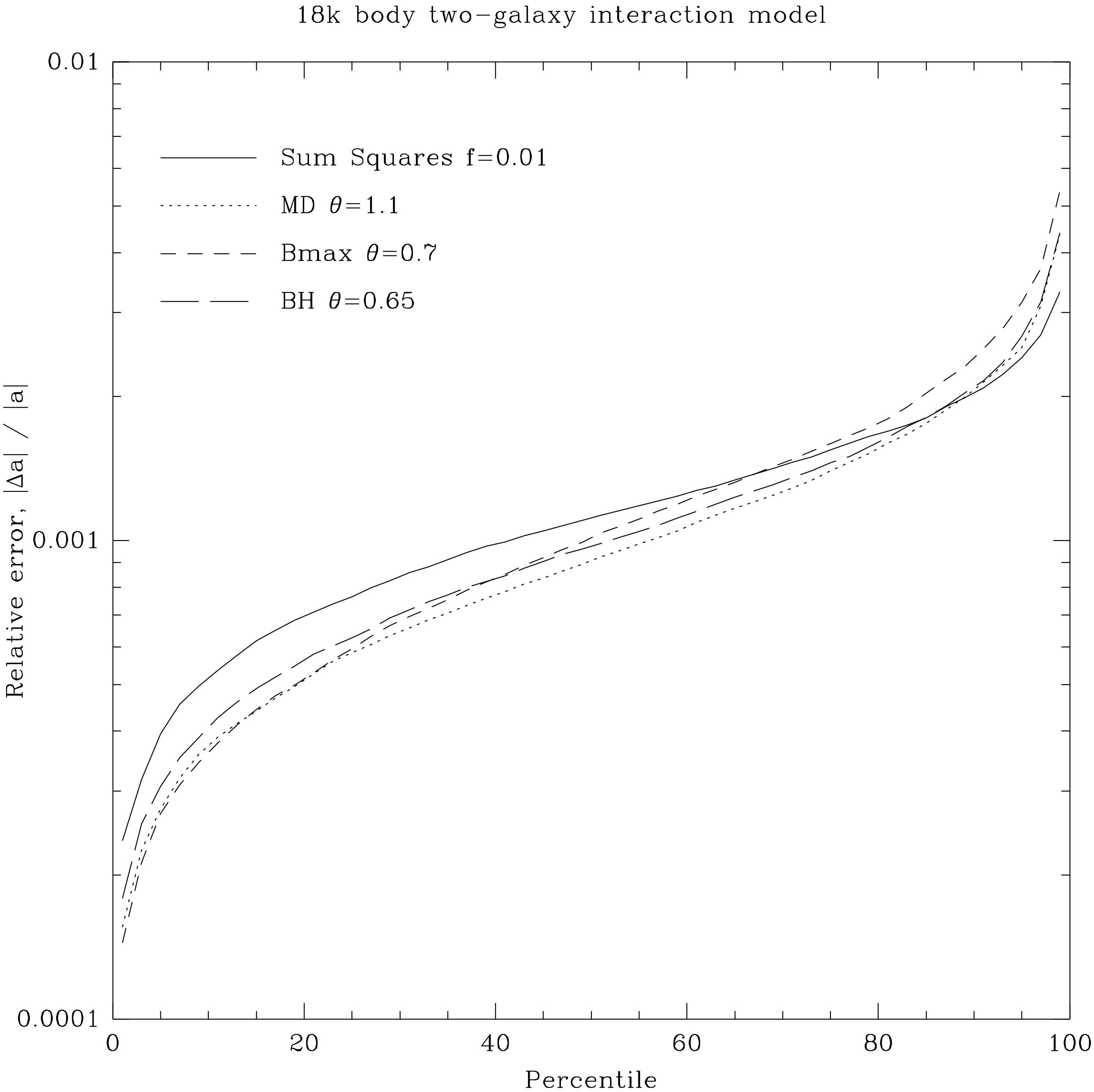

Figure 9: Percentiles of relative error in the total acceleration computed for 4443 bodies in a two-galaxy encounter with 18000 bodies.

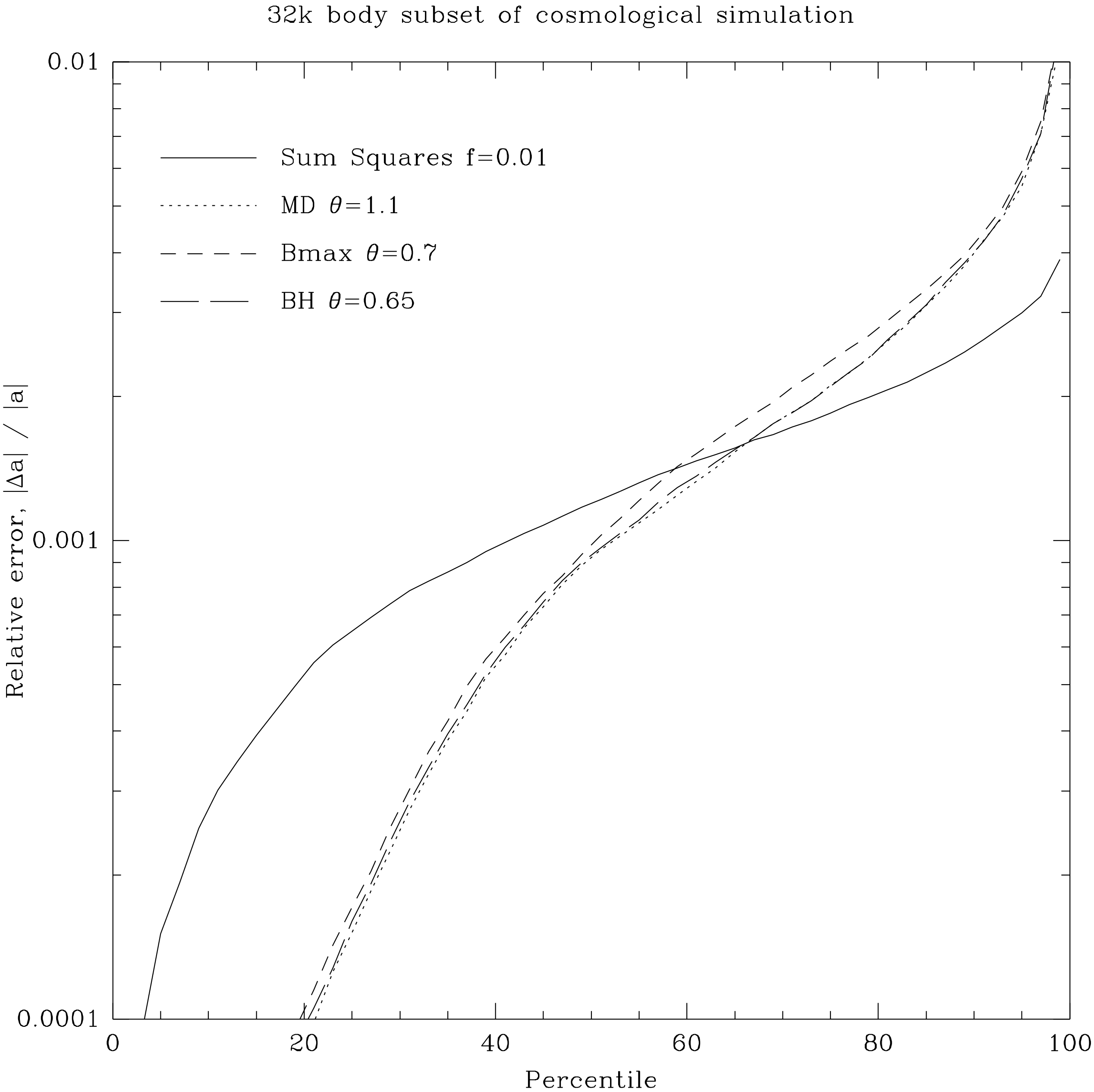

Figure 10: Percentiles of relative error in the total acceleration computed for 4624 bodies in a late-epoch cosmological simulation with 32768 bodies.

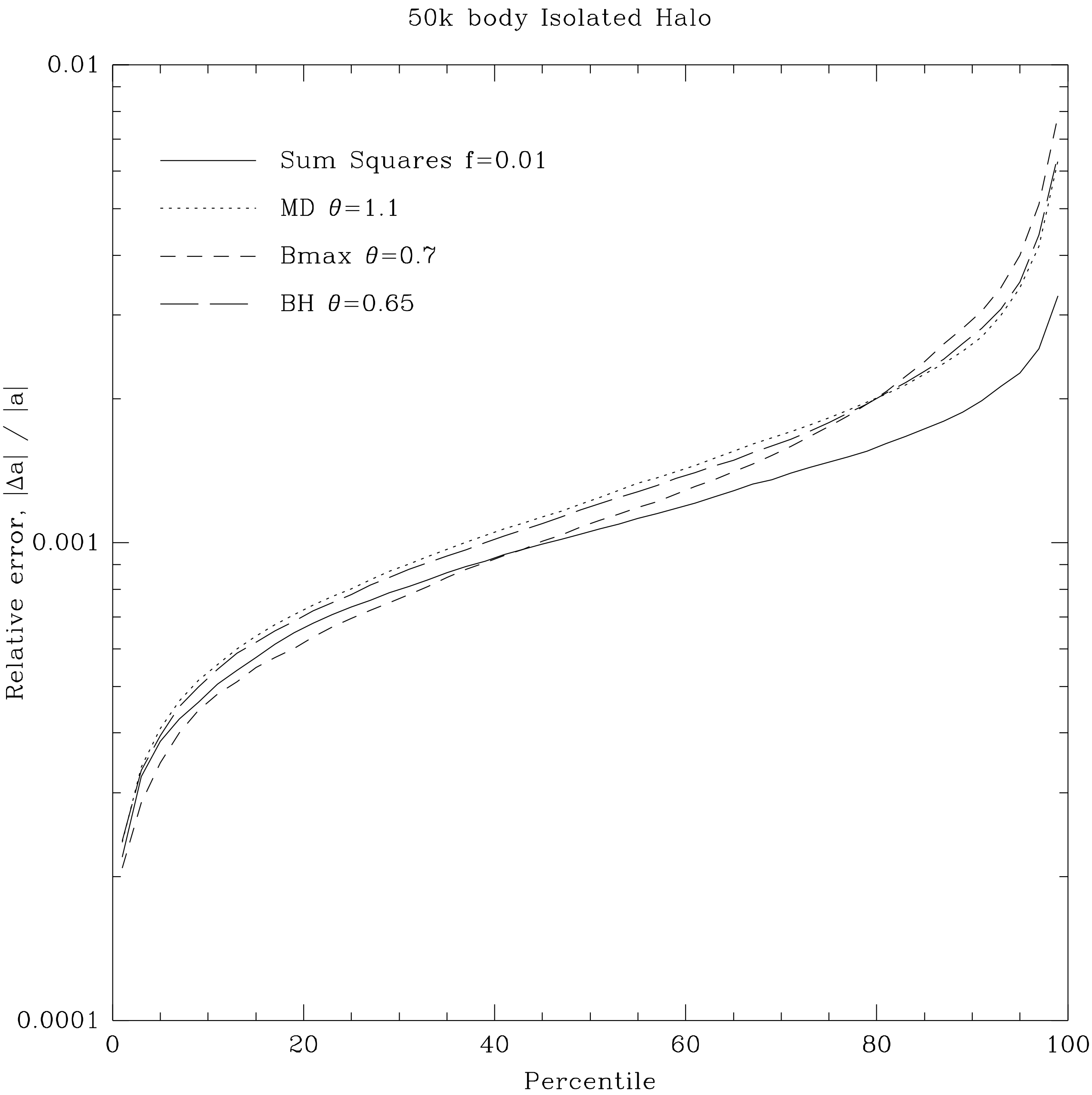

Figure 11: Percentiles of relative error in the total acceleration computed for 4942 bodies in a single triaxial halo.

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 1. 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 be, however, that an analogy with Figure 6 is appropriate, i.e., if much higher accuracy is required, then the use of higher order multipoles may be cost effective.

5 Conclusions

Of the three simple MACs considered (BH, MD, Bmax), the MD MAC appears to be optimal 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 (on the order of a few percent per interaction), no matter which simple MAC is used.

We instead propose a series of new MACs based on a much tighter analytic error bound derived Appendix B, that relies two additional moments of the distribution of matter within a cell. This error bound may be used alone to place strong, reliable limits on the error introduced by each multipole interaction. Alternatively, it can be combined with an alternative ordering of the tree traversal, which is itself based on the error bound, leading to the Sum Squares MAC, which outperforms all of the simple MACs, and guarantees a specified level of error in the total force (i.e., the final answer) 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. Since the Sum Squares MAC achieves this performance with only monopole interactions it enjoys 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 cell may be vanishingly small due to charge cancellation, but dipole and higher terms may be significant.

Finally, we observe that the idea of applying a tight, analytic error bound is applicable to Cell-Cell treecodes as well. We suspect that an adaptive Cell-Cell treecode using a MAC analogous to the Sum Squares MAC will provide superior asymptotic scaling for very large N, while still performing well with modest N because of its ability to use lower order multipole expansions.

Acknowledgments

We would like to thank the Concurrent Supercomputing Consortium and the Caltech Concurrent Supercomputing Facility for providing computational resources. JS acknowledges support from the Advanced Computing Division of the NSF, as well as the Center for Research on Parallel Computation. MW acknowledges support from IGPP. We are indebted to Grégoire Winckelmans for pointing out numerous typographical errors in the early manuscript, and to Peter Quinn and Wojciech Zurek for helpful suggestions.

A Detonating galaxies

It is tempting to dismiss the analytic worst-case error analysis of Section 3.1 as being far too pessimistic. One might argue that the worst-case will never arise in practice. Indeed, previous semi-empirical studies of errors in treecodes have never reported errors even remotely approaching those in Section 3.1. In this appendix we present a *reasonable* astrophysical simulation which encounters catastrophic errors when integrated using the conventional BH MAC. The point of this exercise is not to imply that all astrophysical simulations contain geometric configurations like those presented here. Nor is it relevant that the simulation presented here may abstract away crucial properties of real galaxies. The significance is that (we think) the non-physical detonating behavior will come as a great surprise to those familiar with prior, semi-empirical work on errors in tree codes [13, 14, 10]. In light of the "discovery" that errors in real simulations can far exceed the "typical" errors previously reported, we argue that conservative practitioners of N-body simulations should rely on analytic worst-case error analyses, or otherwise convince themselves that the worst-case bounds will not be approached. It is not sufficient to convince one's self that the particular configuration discussed here does not arise in a particular simulation. There may be other "undiscovered" configurations that also give rise to systematically erroneous acceleration evaluations with magnitude comparable to the error bounds of Figure 5.

Consider the situation in Figure 12. 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. Each of these galaxies is a self-gravitating collection of hundreds or thousands of individual bodies. Unless θ is very small (by the standards of normal astrophysical practice), 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 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 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.

Hernquist [13] has reported on a closely related source of error, and he suggests adding an additional test to the BH MAC to guarantee that bodies inside cell C cannot interact with themselves by interacting with cell C. His prescription, however, does nothing for points like X in Figure 12.

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 2 [5]. Both models are truncated at a radius of $10r_J$. The simulations were carried out using the BH MAC and oct-trees in which the root cell was a cube of size 24.0, centered on the origin. We expect a situation like Figure 12 to occur near $t = 4.0$, when the secondary passes through the corner of the cell at $\vec{x} = (6, 6, 6)$.

Figure 13 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-radii at $t = 4.0$ is the result of the error introduced by the BH MAC.

[5] This is not an entirely contrived example. A very similar system was evolved using the MD MAC (Sec. 2.2), and reported in Salmon et al[25].

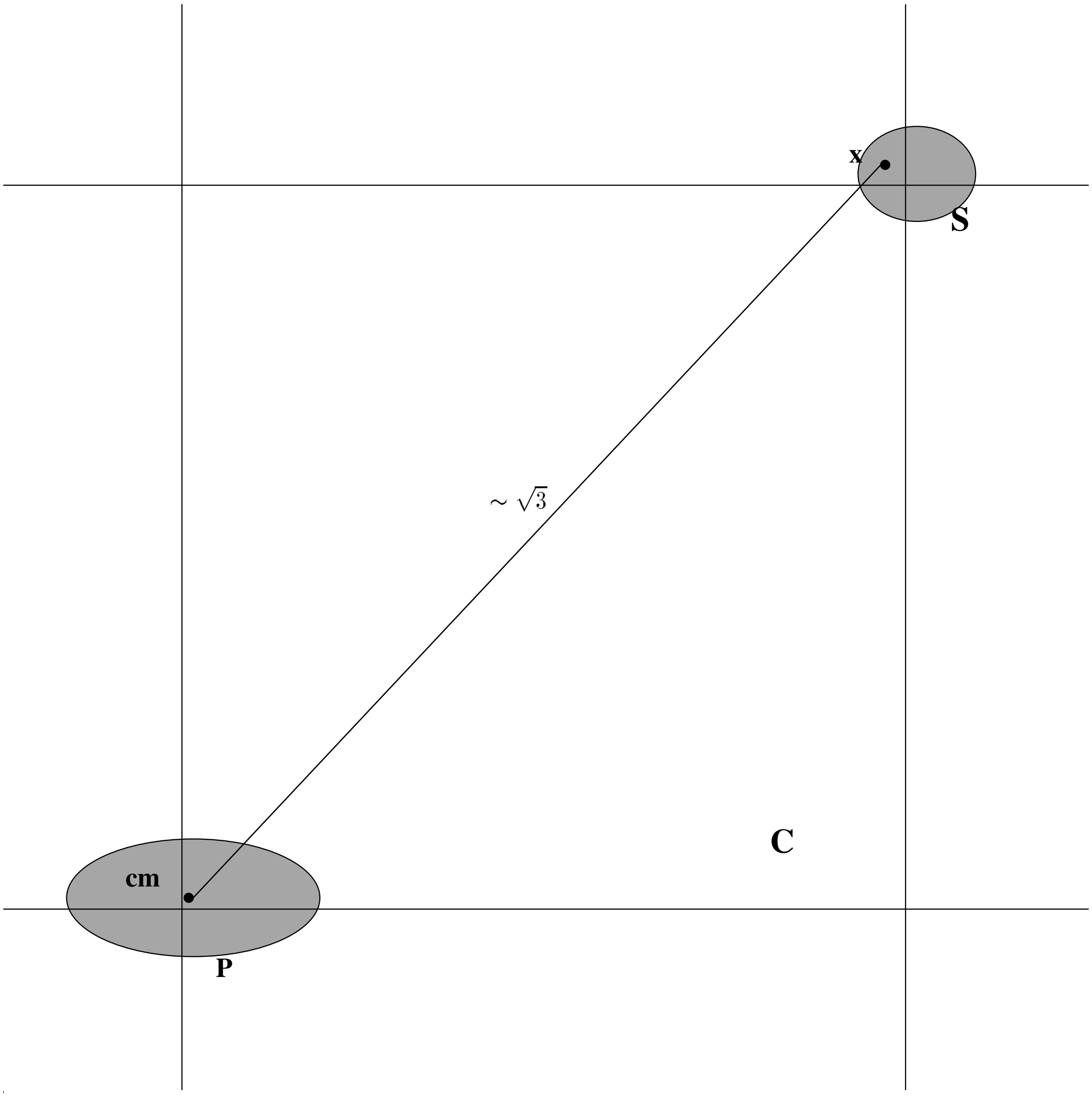

Figure 12: Geometry of "Detonating Galaxy" pathological situation for the BH MAC criterion.

	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 2: Dynamical parameters for two-galaxy simulation.

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. At $\theta \lessapprox 0.65$ the Hernquist criterion 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 13.)

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 14) seems to us to be too large to take chances with.

Figure 14 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 13), 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 13 (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 systems have been used to verify the accuracy of treecodes [14, 22], 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.

A number of authors have empirically measured errors introduced by the BH MAC over a range of values of θ [13, 16, 14, 10]. The situation here 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 errors here only affect a small fraction of the bodies for only a brief time but they does so in such a way as to significantly disrupt the physics of the secondary galaxy.

Even if the other papers had reported maximum relative errors, rather than rms or means, it is unlikely that they would have had cause to warn against $\theta \gtrapprox 1/\sqrt{3}$. Only certain, rather special geometrical arrangements of bodies give rise to errors of the magnitude of Figures 13 or 14. 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 12, 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*.

The two alternative MACs we propose in Section 2 are specifically chosen to avoid the source of error illustrated in Figure 12. In effect, we propose MACs which disallow use of the multipole approximation for points like X in Figure 12. The MD and Bmax MACs may be considered "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 4 are recommended as a more viable, long term solution.

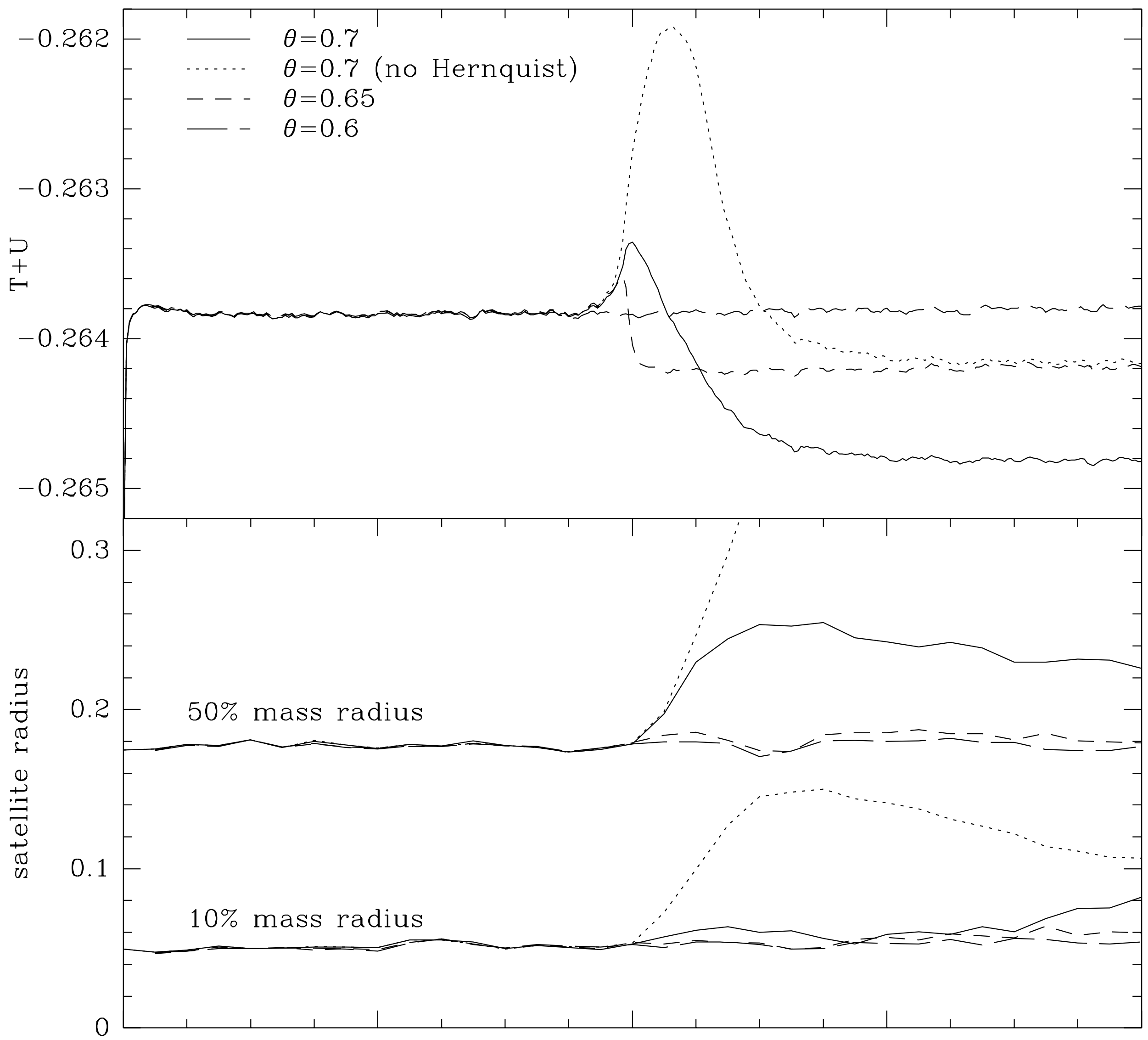

Figure 13: Total energy, T+U and selected mass-radii vs. t for a simulation consisting of a heavy primary and a light satellite (mass ratio, 20:1) on a headon orbit along the $x = y = z$ line.

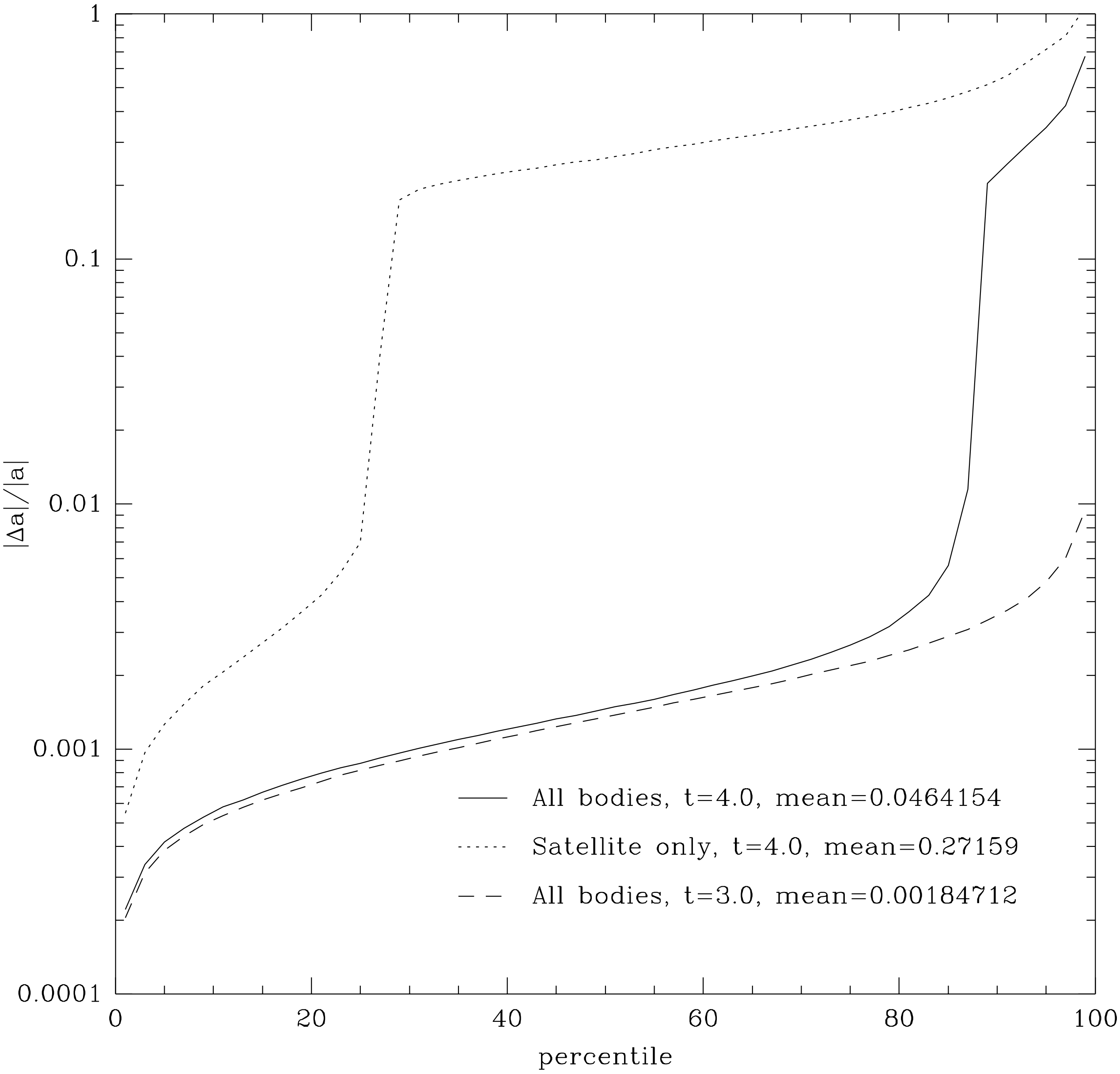

Figure 14: Cumulative probability distribution for relative errors in the acceleration for three samples from the $\theta = 0.7$ model shown in Figure 13. The errors are computed as $\frac{|\Delta\vec{a}|}{|\vec{a}_{exact}|}$, where $\vec{a}_{exact}$ is computed by direct, $O(N^2)$ summation.

B MULTIPOLE EXPANSIONS AND ERROR BOUNDS.

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. Thus, the analysis here applies equally well to the treecodes described in [1, 2, 7].

The potential at a point r is given by

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

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 is possible to apply treecodes to systems with entirely different Green's functions, such as may arise in chemistry [26] or fluid dynamics[27, 28].

The acceleration is obtained from the gradient of the potential:

$$\vec{a}(\vec{r}) = -\nabla\phi(r). \tag{20}$$

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:

$$\phi(r) = \int_{\mathcal{V}} G((\vec{r} - \vec{r}_0) - (\vec{x} - \vec{r}_0))\rho(\vec{x})d^3x. \tag{21}$$

The resulting formulae are:

$$\phi(r) = \sum_{n=0}^{p} \phi_{(n)}(r) + \Delta\phi_{(p)}(r) \tag{22}$$

$$\vec{a}(r) = \sum_{n=0}^{p} \vec{a}_{(n)}(r) + \Delta\vec{a}_{(p)}(r) \tag{23}$$

where

$$\phi_{(n)}(r) = \frac{(-1)^n}{n!} M_{(n)}^{i_1\ldots i_n} \partial_{i_1\ldots i_n} G|_{\vec{r}-\vec{r}_0} \tag{24}$$

$$\vec{a}_{(n)}(r) = -\frac{(-1)^n}{n!} M_{(n)}^{i_1\ldots i_n} \partial_{i_1\ldots i_n} \vec{\nabla} G|_{\vec{r}-\vec{r}_0} \tag{25}$$

The multipole moments are defined by,

$$M_{(n)}^{i_1\ldots i_n} = \int_{\mathcal{V}} d^3x (x - r_0)^{i_1} (x - r_0)^{i_2} \ldots (x - r_0)^{i_n} \rho(x), \tag{26}$$

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

$$M_{(n)}^{i_1\ldots i_n} = \sum_\alpha m_\alpha (x_\alpha - r_0)^{i_1} (x_\alpha - r_0)^{i_2} \ldots (x_\alpha - r_0)^{i_n}. \tag{27}$$

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:

$$\Delta\phi_{(p)}(r) = \int_{\mathcal{V}} d^3x K_{(p)}(r, x)\rho(x) \tag{28}$$

$$\Delta\vec{a}_{(p)}(r) = \int_{\mathcal{V}} d^3x \vec{\nabla}_r K_{(p)}(r, x)\rho(x) \tag{29}$$

where

$$K_{(p)}(r,x) = \frac{(-1)^{p+1}}{p!}(x-r_0)^{i_1}\cdots(x-r_0)^{i_{p+1}}\int_0^1 dt(1-t)^p\partial_{i_1\cdots i_{p+1}}G|_{\vec{r}-\vec{r}_0-t(\vec{x}-\vec{r}_0)} \tag{30}$$

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

$$G(\vec{r}) = -\frac{1}{|\vec{r}|}. \tag{31}$$

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:

$$G(\vec{r}) = -\frac{1}{\left(r^2+\epsilon^2\right)^{\frac{1}{2}}}. \tag{32}$$

Another alternative is to replace the Newtonian potential with a spline-softened potential [19]. 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:

$$R \equiv \left((\vec{r}-\vec{r}_0)^2+\epsilon^2\right)^{\frac{1}{2}} \tag{33}$$

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

$$\vec{h} \equiv \frac{\vec{r}-\vec{r}_0}{R} \tag{34}$$

and the "traced inner product":

$$\left\langle M_{(n)}\middle|\delta^{(l)}h^{(m)}\right\rangle^{i_{2l+m+1}\cdots i_n} = M_{(n)}^{i_1\cdots i_n}\delta_{i_1i_2}\cdots\delta_{i_{2l-1}i_{2l}}h_{i_{2l+1}}\cdots h_{i_{2l+m}} \tag{35}$$

Then Eq.(24) and (25), with the Plummer potential, become:

$$\phi_{(n)}(r) = -\frac{1}{R^{n+1}}\sum_{m=0}^{m\le n/2}\frac{(-1)^m(2n-2m-1)!!}{m!2^m(n-2m)!}\left\langle M_{(n)}\middle|\delta^{(m)}h^{(n-2m)}\right\rangle \tag{36}$$

and

$$\begin{aligned}\vec{a}_{(n)}(r) \;=\; & -\frac{1}{R^{n+2}}\sum_{m=0}^{m\le n/2}\frac{(-1)^m(2n-2m-1)!!}{m!2^m(n-2m)!} \\ & \times\left((2n-2m+1)\left\langle M_{(n)}\middle|\delta^{(m)}h^{(n-2m)}\right\rangle\vec{h}-(n-2m)\left\langle M_{(n)}\middle|\delta^{(m)}h^{(n-2m-1)}\right\rangle\right).\end{aligned} \tag{37}$$

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

$$\begin{aligned}\phi_{(0)} &= -R^{-1}M_{(0)}, \\ \phi_{(1)} &= -R^{-2}\left\langle M_{(1)}\middle|\delta^{(0)}h^{(1)}\right\rangle, \\ \phi_{(2)} &= -R^{-3}\frac{1}{2}\left(3\left\langle M_{(2)}\middle|\delta^{(0)}h^{(2)}\right\rangle-\left\langle M_{(2)}\middle|\delta^{(1)}h^{(0)}\right\rangle\right), \\ \phi_{(3)} &= -R^{-4}\frac{1}{6}\left(15\left\langle M_{(3)}\middle|\delta^{(0)}h^{(3)}\right\rangle-9\left\langle M_{(3)}\middle|\delta^{(1)}h^{(1)}\right\rangle\right), \\ \phi_{(4)} &= -R^{-5}\frac{1}{24}\left(105\left\langle M_{(4)}\middle|\delta^{(0)}h^{(4)}\right\rangle-90\left\langle M_{(4)}\middle|\delta^{(1)}h^{(2)}\right\rangle+9\left\langle M_{(4)}\middle|\delta^{(2)}h^{(0)}\right\rangle\right),\end{aligned} \tag{38}$$

$$
\begin{aligned}
\vec{a}_{(0)i} &= -R^{-2} M_{(0)} \vec{h}, \\
\vec{a}_{(1)i} &= -R^{-3}(3 \left\langle M_{(1)} | \delta^{(0)} h^{(1)} \right\rangle) \vec{h} - \left\langle M_{(1)} | \delta^{(0)} h^{(0)} \right\rangle), \\
\vec{a}_{(2)i} &= -R^{-4} \frac{1}{2} \big(15 \left\langle M_{(2)} | \delta^{(0)} h^{(2)} \right\rangle \vec{h} - 6 \left\langle M_{(2)} | \delta^{(0)} h^{(1)} \right\rangle \\
&\quad -3 \left\langle M_{(2)} | \delta^{(1)} h^{(0)} \right\rangle \vec{h}\big), \\
\vec{a}_{(3)i} &= -R^{-5} \frac{1}{6} \Big(105 \left\langle M_{(3)} | \delta^{(0)} h^{(3)} \right\rangle \vec{h} - 45 \left\langle M_{(3)} | \delta^{(0)} h^{(2)} \right\rangle \\
&\quad -45 \left\langle M_{(3)} | \delta^{(1)} h^{(1)} \right\rangle \vec{h} + 9 \left\langle M_{(3)} | \delta^{(1)} h^{(0)} \right\rangle \Big), \\
\vec{a}_{(4)i} &= -R^{-6} \frac{1}{24} \Big(945 \left\langle M_{(4)} | \delta^{(0)} h^{(4)} \right\rangle \vec{h} - 420 \left\langle M_{(4)} | \delta^{(0)} h^{(3)} \right\rangle \\
&\quad -630 \left\langle M_{(4)} | \delta^{(1)} h^{(2)} \right\rangle \vec{h} + 180 \left\langle M_{(4)} | \delta^{(1)} h^{(1)} \right\rangle \\
&\quad +45 \left\langle M_{(4)} | \delta^{(2)} h^{(0)} \right\rangle \vec{h} \Big).
\end{aligned} \tag{39}
$$

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 harmonic, i.e., a solution of Laplace's equation. Since derivatives may be taken in any order, we have,

$$
\delta^{i_\alpha i_\beta} \partial_{i_1} \partial_{i_2} \cdots \partial_{i_n} G(r) = 0; \qquad \vec{r} \neq 0; \quad \alpha, \beta \leq n. \tag{40}
$$

We define

$$
Q_{(n)} \equiv M_{(n)} - \delta \otimes C_{(n-2)} \tag{41}
$$

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

$$
\left(\delta \otimes A_{(n)}\right)^{i_1 \cdots i_{n+2}} = \underbrace{\delta^{i_1 i_2} A_{(n)}^{i_3 \cdots i_{n+2}} + \cdots}_{\binom{n+2}{2} \text{combinations}} \tag{42}
$$

Because of Eq.(40), the Kronecker delta that appears in Eq.(41) vanishes when contracted with the derivatives of G. Thus, from Eq.(24), (25), (40) and (41):

$$
\begin{aligned}
\phi_{(n)} &= \frac{(-1)^n}{n!} Q_{(n)}^{i_1 \ldots i_n} \partial_{i_1} \cdots \partial_{i_n} G|_{\vec{r}-\vec{r}_0}, && (43) \\
\vec{a}_{(n)} &= -\frac{(-1)^n}{n!} Q_{(n)}^{i_1 \ldots i_n} \partial_{i_1 \ldots i_n} \vec{\nabla} G|_{\vec{r}-\vec{r}_0} && (44)
\end{aligned}
$$

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.,

$$
\delta_{i_\alpha i_\beta} Q_{(n)}^{i_1 \cdots i_n} = 0 \tag{45}
$$

Then all but the leading (i.e., $m = 0$) terms in Eq.(36) and (37) vanish, and we have

$$
\begin{aligned}
\phi_{(n)}(r) &= -\frac{(2n-1)!!}{n!} \frac{1}{d^{n+1}} \left\langle Q_{(n)} | e^{(n)} \right\rangle, && (46) \\
\vec{a}_{(n)}(r) &= -\frac{(2n-1)!!}{n!} \frac{1}{d^{n+2}} \left((2n+1) \left\langle Q_{(n)} | e^{(n)} \right\rangle \hat{e} - n \left\langle Q_{(n)} | e^{(n-1)} \right\rangle \right), && (47)
\end{aligned}
$$

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.(41) 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.(41)) allows one to construct $Q_{(n)}$ from $M_{(n)}$ and its traces:

$$Q_{(n)}^{(\lfloor n/2 \rfloor)} = \delta^{(\lfloor n/2 \rfloor)} M_{(n)}, \tag{48}$$

$$Q_{(n)}^{(m)} = \delta^{(m)} M_{(n)} - \frac{1}{(m+1)(2n-2m-1)} \delta \otimes Q_{(n)}^{(m+1)}, \tag{49}$$

$$Q_{(n)} = Q_{(n)}^{(0)}, \tag{50}$$

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:

$$\begin{aligned}
Q_{(0)} &= M_{(0)}, \\
Q_{(1)} &= M_{(1)}, \\
Q_{(2)} &= M_{(2)} - \frac{1}{3}\delta \otimes \delta M_{(2)}, \\
Q_{(3)} &= M_{(3)} - \frac{1}{5}\delta \otimes \delta M_{(3)}, \\
Q_{(4)} &= M_{(4)} - \frac{1}{7}\delta \otimes \left(\delta M_{(4)} - \frac{1}{10}\delta \otimes (\delta^{(2)} M_{(4)})\right), \\
Q_{(5)} &= M_{(5)} - \frac{1}{9}\delta \otimes \left(\delta M_{(5)} - \frac{1}{14}\delta \otimes (\delta^{(2)} M_{(5)})\right), \\
Q_{(6)} &= M_{(6)} - \frac{1}{11}\delta \otimes \left(\delta M_{(6)} - \frac{1}{18}\delta \otimes \left(\delta^{(2)} M_{(6)} - \frac{1}{21}\delta \otimes (\delta^{(3)} M_{(6)})\right)\right).
\end{aligned} \tag{51}$$

B.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

$$K_{(p)}(r, x) = -\left(\frac{(p+1)}{r}\right) \alpha^{p+1} \int_0^1 \left(\frac{r}{r_t}\right)^{p+2} (1-t)^p P_{p+1}(\mu_t) dt \tag{52}$$

where P_n is the Legendre polynomial of degree n, unit-vectors are denoted $\hat{v} = \vec{v}/|\vec{v}|$, and

$$\begin{aligned}
\alpha &= \frac{|\vec{x}|}{r}, \\
\vec{r}_t &= \vec{r} - t\vec{x}, \\
\mu_t &= \hat{r}_t \cdot \hat{x}.
\end{aligned} \tag{53}$$

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

$$\nabla K_{(p)}(r, x) = \frac{1}{r^2} \alpha^{p+1} (p+1) \int_0^1 dt (1-t)^p \left(\frac{r}{r_t}\right)^{p+3} \left((p+2) P_{p+1}(\mu_t) \hat{r}_t - (\hat{x} - \mu_t \hat{r}_t) P'_{p+1}(\mu_t)\right). \tag{54}$$

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

$$\begin{aligned}
\frac{r}{r_t} &\leq \frac{1}{1 - \alpha t} \\
|P_n(\mu)| &\leq 1 \quad \text{for } |\mu| \leq 1 \\
(n+1)^2 P_n(\mu)^2 + (1-\mu^2) P'_n(\mu)^2 &\leq (n+1)^2 \quad \text{for } |\mu| \leq 1
\end{aligned} \tag{55}$$

from which we conclude

$$\left|K_{(p)}(r,x)\right| \leq \frac{1}{r}\frac{\alpha^{p+1}}{1-\alpha} \tag{56}$$

$$\left|\vec{\nabla}_r K_{(p)}(r,x)\right| \leq \frac{1}{r^2}\frac{\alpha^{p+1}}{(1-\alpha)^2}(p+2-\alpha(p+1)) \tag{57}$$

Using Eq.(57) we obtain:

$$\Delta a_{(p)}(r) \leq \frac{1}{r^2}\int_{\mathcal{V}} d^3x\,|\rho(x)|\,\frac{1}{(1-\alpha)^2}\left((p+2)\alpha^{p+1}-(p+1)\alpha^{p+2}\right) \tag{58}$$

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

$$\alpha \leq \frac{b_{max}}{d}, \tag{59}$$

we obtain:

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

where the moments:

$$B_{(n)} = \int_{\mathcal{V}} d^3x\,|\rho(x)|\,|\vec{x}-\vec{r}_0|^n = \sum_\beta |m_\beta|\,|\vec{x}_\beta-\vec{r}_0|^n \tag{61}$$

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 gravitational problems, but they are common in molecular-dynamics simulations with charged species.

B.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. In order to use these bounds in, e.g., molecular dynamics with charged species, it would be necessary to retain not only multipole moments of the charge distribution, q_α but also the multipole moments of the absolute value of the charge distribution, $|q_\alpha|$. Once $B_{(p+1)}$ and $B_{(p+2)}$ are found, the storage for the absolute value multipoles is no longer needed, and could be returned to the operating system. The latter is used to construct $B_{(n)}$, while the former is used to compute fields and gradients. In terms of upper and lower bounds on $B_{(n)}$, we have:

$$\Delta a_{(p)}(r) \leq \frac{1}{d^2}\frac{1}{(1-\frac{b_{max}}{d})^2}\left((p+2)\left(\frac{\lceil B_{(p+1)}\rceil}{d^{p+1}}\right)-(p+1)\left(\frac{\lfloor B_{(p+2)}\rfloor}{d^{p+2}}\right)\right). \tag{62}$$

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

$$B_{(n)} = \delta^{(n/2)}M_{(n)} \qquad \text{even } n. \tag{63}$$

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

$$B_{(n+k)} \leq b^k_{max}B_{(n)} \tag{64}$$

The following formulas, which follow from Hölder's inequality [29] are useful for even and odd p respectively:

$$B_{(p+2)} \geq \frac{B^2_{(p)}}{B_{(p-2)}} = \frac{\left(\delta^{(p/2)}M_{(p)}\right)^2}{\delta^{((p-2)/2)}M_{(p-2)}} \tag{65}$$

$$B^2_{(p+2)} \geq \frac{B^5_{(p-1)}}{B^3_{(p-3)}} = \frac{\left(\delta^{((p-1)/2)} M_{(p-1)}\right)^5}{\left(\delta^{((p-3)/2)} M_{(p-3)}\right)^3} \tag{66}$$

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 $B_{(3)}$:

$$B_{(2)} = \delta M_{(2)} \tag{67}$$

$$B^2_{(3)} \geq \frac{B^3_{(2)}}{B_{(0)}} = \frac{(\delta M_{(2)})^3}{M_{(0)}}. \tag{68}$$

B.3 Weak bounds

Eq.(62) 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 Eq.(59) and the fact that Eq.(56) and Eq.(57) are monotonically increasing functions of α:

$$\Delta\phi_{(p)} \leq \frac{B_{(0)}}{d} \frac{1}{1 - \frac{b_{max}}{d}} \left(\frac{b_{max}}{d}\right)^{p+1} \tag{69}$$

$$\Delta a_{(p)}(r) \leq \frac{B_{(0)}}{d^2} \frac{1}{(1 - \frac{b_{max}}{d})^2} \left(\frac{b_{max}}{d}\right)^{p+1} \left(p + 2 - \frac{b_{max}}{d}(p+1)\right). \tag{70}$$

The first term may be identified with the monopole interaction in the case of a non-negative ρ field. The maximum relative error is thus a monotonically increasing function of b_{max}/d, which is useful primarily for motivating the Bmax MAC.

Eq.(69) is similar to results obtained by Greengard[4], where it is used to select a suitable multipole order, p. Note, however, that the relative error in the acceleration is considerably larger than the relative error in the potential. Thus, if one bounds the potential with order p multipoles, the error in the acceleration may be almost p times as large, which can amount to an order of magnitude or more in practice.

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] M. S. Warren and J. K. Salmon. Astrophysical N-body simulations using hierarchical tree data structures. In *Supercomputing '92*, (IEEE Comp. Soc., Los Alamitos, 1992).

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

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

[15] 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).

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

[17] 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.

[18] 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.

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

[20] L. Hernquist, P. Hut, and J. Makino. Discreteness noise versus force errors in N-body simulations. *Ap. J.*, **402**, L85, (1993).

[21] J. Makino, T. Ito, and T. Ebisuzaki. Error analysis of the GRAPE-1 special purpose N-body machine. *Publ. Astron. Soc. Japan*, **42**, 717, (1990).

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

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

[24] M. S. Warren, P. J. Quinn, J. K. Salmon, and W. H. Zurek. Dark halos formed via dissipationless collapse: I. shapes and alignment of angular momentum. *Ap. J.*, **399**, 405, (1992).

[25] 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.

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

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

[28] 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).

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