

High-Accuracy Treecode Based on Pseudoparticle Multipole Method

Atsushi Kawai

*Computational Science Division, Advanced Computing Center, RIKEN,
2-1 Hirosawa, Wako-shi, Saitama, 351-0198, Japan*

Junichiro Makino

*Department of Astronomy, School of Science, University of Tokyo, 7-3-1
Hongo, Bunkyo-ku, Tokyo 113-0033, Japan*

Abstract. We invented the pseudoparticle multipole method (P^2M^2), a method to express multipole expansion by a distribution of pseudoparticles. We can use this distribution of particles to calculate high order terms in both the Barnes-Hut treecode and FMM. The primary advantage of P^2M^2 is that it works on GRAPE. Although the treecode has been implemented on GRAPE, we could handle terms only up to dipole, since GRAPE can calculate forces from point-mass particles only. Thus the calculation cost grows quickly when high accuracy is required. With P^2M^2 , the multipole expansion is expressed by particles, and thus GRAPE can calculate high order terms. Using P^2M^2 , we realized arbitrary-order treecode on MDGRAPE-2. Timing result shows MDGRAPE-2 accelerates the calculation by a factor between 20 (for low accuracy) to 150 (for high accuracy). We parallelized the code so that it runs on MDGRAPE-2 cluster. The calculation speed of the code shows close-to-linear scaling up to 16 processors for $N \gtrsim 10^6$.

1. Introduction

The calculation of the gravitational force is the most expensive part of almost all N -body simulations. The Barnes-Hut treecode (Barnes & Hut 1986) is a widely used algorithm to reduce the cost of the force calculation. In the treecode, forces on a particles from distant particles are replaced by multipole expansions of groups of particles. More distant particles are organized into larger groups, so that the truncation error of the expansion is similar everywhere. Hierarchical tree structure is used to form grouping efficiently. The calculation cost is reduced from $O(N^2)$ to $O(N \log N)$.

Even with the treecode, the cost of force calculation is still high, and it dominates the total calculation cost. In order to accelerate the treecode further, we can use GRAPE (Sugimoto et al 1990; Makino & Taiji 1998). GRAPE is a special-purpose hardware for the calculation of gravitational force between particles. It works in cooperation with a general-purpose computer (host computer). The host computer does everything except for the force calculation. The

application of GRAPE to the treecode is introduced by Makino (1991), who implemented Barnes' (1990) modified algorithm on GRAPE-1A (Fukushige et al. 1991). Athanassoula et al. (1998) and Kawai et al. (2000) reported its performance on GRAPE-3 and GRAPE-5, respectively. The speedup factor they obtained is in the range of 30 to 50.

Although GRAPE can significantly accelerate the treecode, its application has been limited to simulations where the requirement for the accuracy is modest. Since GRAPE can only calculate forces from point mass particles, we have not been able to handle terms of the multipole expansion higher than dipole. Thus, the calculation cost grows quickly when high accuracy is required.

We have developed the pseudoparticle multipole method (P^2M^2) which makes it possible to evaluate higher-order expansions on GRAPE. Using P^2M^2 , multipole expansions are represented by a small number of pseudoparticles. The masses and positions of pseudoparticles are determined so that they have the same expansion coefficients as the corresponding physical particles up to the specified order. With the P^2M^2 , we can use GRAPE to evaluate high order terms, since they are expressed by particles. Thus, for high accuracy the advantage of our method over dipole code with GRAPE is large. The crossover is achieved at rather low accuracy (around 0.5% for relative force error). So our method is useful for a wide range of applications.

In the following, we first describe the procedure to express multipole expansion using pseudo particles (§ 2). Then we give the result of numerical tests on MDGRAPE-2 (§ 3), and summarize this proceeding (§ 4).

2. The Pseudoparticle Multipole Method

In P^2M^2 , we distribute the pseudoparticles so that they exactly reproduce the coefficients of the multipole expansion of real (physical) particles up to a given order. Conceptually, in order to obtain such a distribution, first we calculate the expansion coefficients from the distribution of physical particles. We then solve the inverse problem to obtain the masses and positions of pseudoparticles. In § 2.1, we describe a practical procedure to obtain the distribution of the pseudoparticles which can be used up to quadrupole. In § 2.2, we describe a procedure to obtain the distribution for higher order.

2.1. Quadrupole Moment with Pseudoparticles

In Cartesian coordinates, the multipole expansion up to quadrupole of the potential due to N particles is expressed as

$$\Phi(\mathbf{r}) = \sum_{i=1}^N m_i \left\{ \frac{1}{r} + \frac{\mathbf{r}\mathbf{r}_i}{r^3} + \frac{1}{r^5} \left[\frac{3}{2}(\mathbf{r}\mathbf{r}_i)^2 - \frac{1}{2}r^2r_i^2 \right] \right\}. \quad (1)$$

The mass M_j and the position \mathbf{R}_j of pseudoparticles must be determined so that they give the same Φ up to a given order p . In general, the expansion up to the p -th order has $(p+1)^2$ independent terms. Since each pseudoparticle has four degrees of freedom (one for mass and three for position), theoretically we can reproduce the expansion using $K_{\min}(p) = \lceil (p+1)^2/4 \rceil$ pseudoparticles. Here $\lceil x \rceil$ denotes the minimum integer not smaller than x .

In order to express multipole expansion of order $p = 0$, we need $K_{\min}(0) = 1$ pseudoparticle. We must put the particle so that M_1 and \mathbf{R}_1 reproduce the first term (monopole term) of the right hand side of equation (1). For this purpose we can set the mass $M_1 = M$, where M is the total mass of physical particles. At least formally, the position of the pseudoparticle can be anywhere. In figure 1 we place them at the origin as example. In practice, we would never use zeroth order expansion since we can achieve the first order accuracy with one particle, as we will see below.

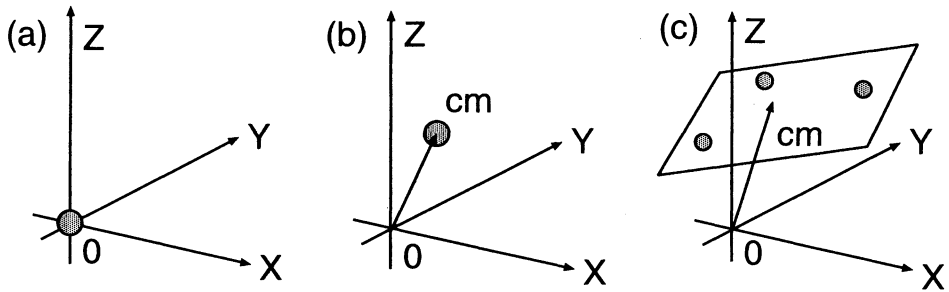


Figure 1. The positions of pseudoparticles which reproduce the multipole expansion up to monopole (left), dipole (middle), and quadrupole (right).

For $p = 1$, M_j and \mathbf{R}_j must reproduce the first and the second term (dipole term) of the right hand side of equation (1). We can satisfy this condition by placing a single pseudoparticle with mass M at the center of mass of physical particles, \mathbf{r}_{cm} (see figure 1b), as is done with the original Barnes-Hut treecode.

For $p = 2$, the minimum number of pseudoparticles necessary is $K_{\min}(2) = 3$. In the following, we'll see whether we can actually construct the distribution of three pseudoparticles which reproduces the multipole expansion up to quadrupole.

In order to reproduce the first and the second terms, the total mass of the pseudoparticles should be M , and their center of mass should be located at \mathbf{r}_{cm} . This is achieved by making their total mass M and center of mass to be the same as that of physical particles. In the following, we use a coordinate system shifted so that $\mathbf{r}_{cm} = \mathbf{0}$.

Pseudoparticles should have the same quadrupole tensor

$$\mathcal{A} = \frac{3}{2} \sum_{i=1}^N m_i \begin{bmatrix} x_i x_i & x_i y_i & x_i z_i \\ y_i x_i & y_i y_i & y_i z_i \\ z_i x_i & z_i y_i & z_i z_i \end{bmatrix} - \frac{1}{2} \sum_{i=1}^N m_i r_i^2 \tag{2}$$

as physical particles to reproduce the third term in equation (1).

By definition, \mathcal{A} is symmetric and traceless. Therefore we can choose the coordinate axe so that \mathcal{A} is diagonalized. In this coordinate system, \mathcal{A} is expressed as:

$$\mathcal{A} = \begin{bmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & -(a_1 + a_2) \end{bmatrix}. \tag{3}$$

We can choose a_1 and a_2 , so that the relation

$$a_1 \geq a_2 \geq -(a_1 + a_2) \quad (4)$$

is satisfied.

Obviously, all three pseudoparticles should be on the x-y plane. Now our problem is reduced to determining the masses and positions of three particles on the x-y plane so that they give the quadrupole moment tensor of the form (3).

There are a variety of ways to satisfy this requirement. Here we give just one example.

We still have three extra degrees of freedom, since we can change masses and positions of two particles on the x-y plane freely, and we have only three constraint. In our procedure, we set the masses as $M_1 = M_2 = M_3 = M/3$. These masses reproduce the first term of the equation (1). Now we have only one extra degree of freedom left. We use it by setting x component of \mathbf{R}_1 to 0. Now we can determine the position vectors as follows:

$$\mathbf{R}_1 = \begin{bmatrix} 0 \\ 2\beta \\ 0 \end{bmatrix}, \mathbf{R}_2 = \begin{bmatrix} \alpha \\ -\beta \\ 0 \end{bmatrix}, \mathbf{R}_3 = \begin{bmatrix} -\alpha \\ -\beta \\ 0 \end{bmatrix}, \quad (5)$$

where α and β are defined as

$$\alpha \equiv \sqrt{(2a_1 + a_2)/M}, \quad \beta \equiv \sqrt{(a_1 + 2a_2)/(3M)}. \quad (6)$$

Note that both α and β are guaranteed to be real numbers. As we have already mentioned, this solution is not unique. For example, if we set y , instead of x , component of \mathbf{R}_1 to 0, we obtain another solution that attains the same order of accuracy.

2.2. Higher Order Generalization

Here we describe the generalization of P^2M^2 for multipole expansion of higher order. The procedure we described in the previous section depends closely on the nature of quadrupole-moment tensor, and it is difficult to extend this procedure to higher orders.

Makino (1999) proposed a different approach based on the orthogonality of spherical harmonics. His approach gives a systematic procedure to obtain the distribution for an arbitrary order.

The key idea of Makino's approach is to fix the positions of pseudoparticles. To obtain the distribution of pseudoparticles, we need to invert the multipole expansion formula. Direct inversion is not easy for high order expansion, since the formula is nonlinear for the positions of pseudoparticles. However, if we fix the positions, the equation becomes linear, since the formula is linear for the masses of pseudoparticles. In this case we can invert the formula relatively easily. On the other hand, the necessary number of pseudoparticles increases, since we can adjust only the masses, and the degree of freedom assigned to each particle is reduced from four to one. Because of the rather large number of pseudoparticles, his procedure is not efficient when the required accuracy is modest. But it may be useful for calculations that require very high accuracy. For the formulation detail of the generalized P^2M^2 , see Makino (1999).

3. Numerical Tests

Using P^2M^2 , we implemented an arbitrary-order treecode (hereafter referred to as P^2M^2 treecode) on GRAPE and a UNIX workstation. Also we parallelized the code on a distributed memory system consists of workstations and GRAPE processor boards attached to each of them. The parallelization scheme we used are basically the same as Warren & Salmon's (1993) Hashed Oct-Tree algorithm. The source code for both serial and parallel implementations are available upon request.

In the following we show the results of numerical tests. In § 3.1 and § 3.2, we show the accuracy of instantaneous force calculation and that of dynamical simulation, respectively. For these measurement we used the serial code on one MDGRAPE-2 (Narumi et al. 2000) processor board (192 Gflops). The host computer for MDGRAPE-2 is a COMPAQ DS20E with Alpha 21264 processor, running at 667 MHz. In § 3.3, we show the the calculation speed of the parallel code. For this measurement we used an MDGRAPE-2 cluster, which consists of 16 processor boards of MDGRAPE-2 and 16 DS20Es connected via Myrinet.

3.1. Speed and Accuracy of Force Calculation

Here we show the accuracy and the speed of instantaneous force calculation.

We uniformly distributed 262144 equal-mass particles within a unit sphere. Then we calculated the force on each particle with P^2M^2 treecode, and measured relative error e and calculation time T for various values of the opening angle θ . The error e is defined as r.m.s. relative difference from the exact force. As the exact force, we used the force calculated with direct summation algorithm on the host computer using IEEE-754 standard 64-bit arithmetic. Barnes' (1990) vectorization parameter, $n_{\text{crit}} = 8000$ and 1 are used for runs with and without MDGRAPE-2, respectively. These values are close to optimal for our hardware configuration.

Figures 2 and 3, and table 1 show the results. In figure 2, the errors for P^2M^2 treecode of orders $p = 1$ through 4 are plotted as functions of θ . We can see that the error roughly scales as $\theta^{p+1.5}$. This behavior agrees well with the theoretical estimate given by Makino (1990). The saturation of the accuracy at $e \approx 10^{-8}$ is due to the hardware limitation of MDGRAPE-2.

In figure 3, the force error e is plotted as functions of the calculation time T . We can see that P^2M^2 treecode of order $p = 2$ is the fastest for a wide range of accuracy ($10^{-6} \lesssim e \lesssim 10^{-3}$). For low accuracy ($e \gtrsim 10^{-3}$), the treecode of order $p = 1$ is the fastest. For extremely high accuracy ($e \lesssim 10^{-7}$), P^2M^2 treecode of order $p = 4$ is the fastest. We performed the same test without MDGRAPE-2, and found that MDGRAPE-2 accelerates the calculation by a factor of 20 (for $e \approx 10^{-2}$) to 150 (for $e \approx 10^{-6}$). We performed the same test also on GRAPE-5 to confirm that our method works well on low-accuracy hardwares. The results are similar to those on MDGRAPE-2, except that the accuracy saturate around $e \approx 4 \times 10^{-4}$ due to the hardware limitation.

In table 1, the breakdown of calculation time are given for P^2M^2 treecode of order $p = 2$ (quadrupole) and the treecode of order $p = 1$ (dipole). We can see that the force calculation using GRAPE is the most time-consuming part. The time to set up the pseudoparticles is about 10% of the total time.

When we compare the timings for $p = 1$ and 2, we can see that the calculation time is only 30% longer for $p = 2$, though each node in $p = 2$ calculation consists of 3 pseudoparticles. This is because we used the algorithm optimized for GRAPE, in which the forces from nearby particles are calculated directly Barnes (1990). Since this part dominates the total cost, the increase in the calculation cost of the forces from nodes has rather small effect on the calculation cost.

Table 1. BREAKDOWN OF COMPUTATION TIME ($N = 262144, \theta = 0.5$)

	$p = 1$ (s/step)	$p = 2$ (s/step)
Host Computer		
Tree Construction	0.27	0.20
Pseudoparticles Set up	0.20	0.36
Interaction List Construction	0.55	0.62
MDGRAPE-2	0.63	0.92
Data Transfer	0.64	0.90
Total	2.29	3.00

3.2. An Example of Simulation

Here we discuss the overall accuracy of the simulation. We have already seen our code attains high accuracy in calculation of instantaneous force, but this high accuracy does not necessarily guarantee the accuracy of the total simulation. For example, if the force errors in consecutive timesteps are strongly correlated, they accumulate. As a result, overall accuracy of the orbits of stars might be worse, compared to the case with weaker correlation.

We performed a simulation of a collision of two identical galaxies. We chose the system of units so that the total mass of each galaxy and the gravitational constant are both unity and the total energy of each galaxy is $-1/4$. Galaxy model we used is the Hernquist model Hernquist (1990) with 65536 equal-mass particles. We cut off the distribution at radius 20. We place the galaxies at initial separation 3.0, and gave initial velocities so that they would follow a parabolic head on collision. We integrated this system up to $t = 16$ with time step $\Delta t = 1/200$ and softening parameter $\epsilon = 1/100$. We performed the same simulation using three different codes, namely, P²M² treecode of order $p = 2$, the treecode of order $p = 1$, and the direct summation algorithm. For treecode simulations, we used the opening angle $\theta = 0.5$.

Figure 4 shows the results of simulations. The relative error of the total energy, $dE(t)/E(0)$, is plotted as a function of time t . For all runs, we can see three bumps around $t = 2, 7$, and 8, which correspond to close encounters of the two galaxies. These bumps comes from the time-integration error, since they can be seen in the result of the direct summation algorithm with highly accurate force calculation. We can regard the deviation from the result of the direct summation as being caused by the error of the force calculated with the

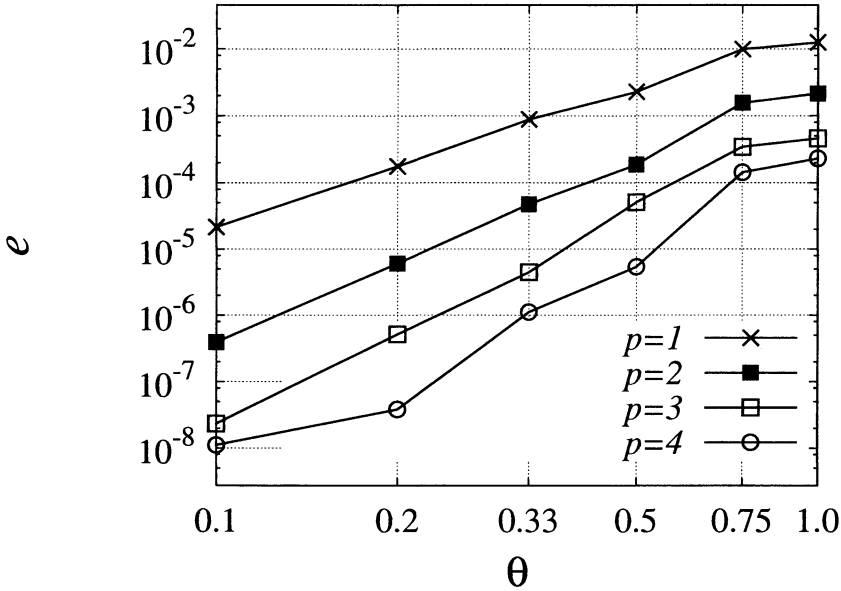


Figure 2. The r.m.s. relative force error e plotted against the opening angle θ . Curves with crosses, filled squares, open squares, and circles are results for treecodes with $p = 1, 2, 3$ and 4 , respectively.

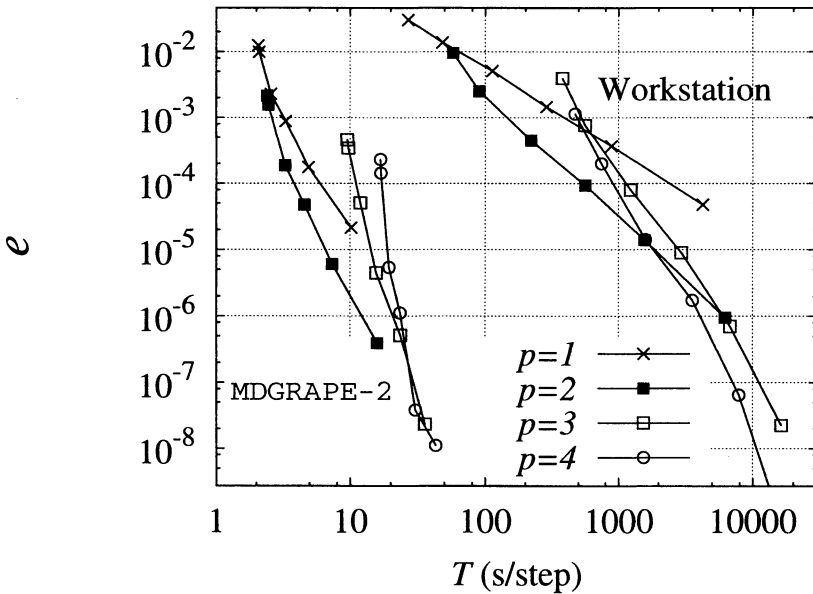


Figure 3. The r.m.s. relative force error e plotted against the calculation time T . Meaning of the symbols are the same as those in figure 2.

treecodes. For the treecode with $p = 2$, the maximum deviation from the direct summation is 5×10^{-5} , while for $p = 1$ it is 4×10^{-4} . This a factor of 10 difference is consistent with the difference of the accuracy of the instantaneous force shown in figure 2. The simulations took 1.7 and 2.4 hours for $p = 1$ and $p = 2$, respectively. Thus we can conclude that the high accuracy in the instantaneous force offered by our new P^2M^2 does improve the overall accuracy of the time integration, with modest increase of calculation time.

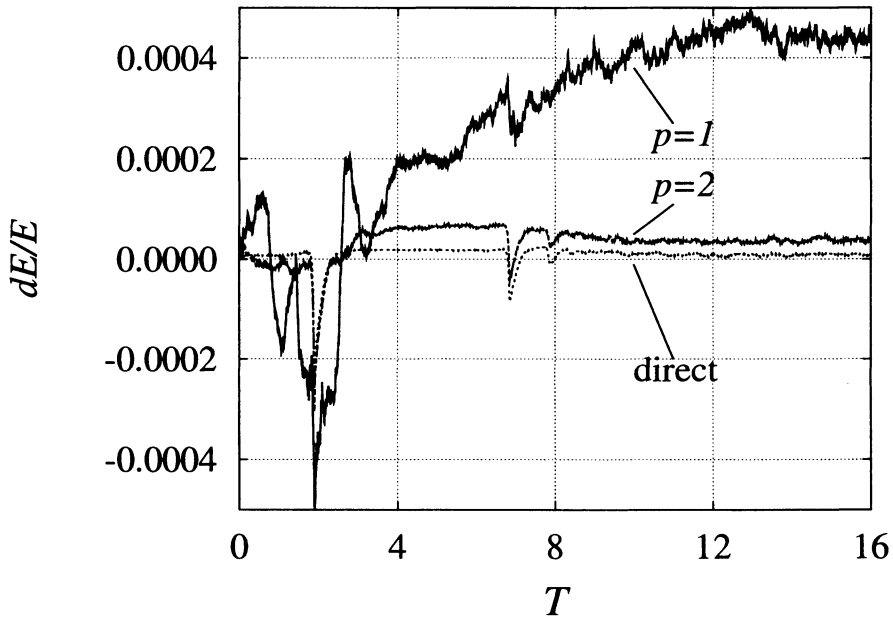


Figure 4. The relative error of the total energy plotted as a function of time. Solid, long-dashed and short-dashed curves are for treecodes with $p = 1$, $p = 2$, and direct summation algorithm, respectively.

3.3. Scalability of the Parallel Code

Here we show the calculation speed of our parallelized P^2M^2 treecode.

We uniformly distributed various numbers of equal-mass particles within a unit sphere, and measured the calculation time using different numbers of workstations (processor elements). Each processor element used one MDGRAPE-2 processor board. The opening angle used was $\theta = 0.5$.

Figure 5 and 6 show the results. In figure 5, the calculation time T measured with the parallel code of order $p = 1$ and $p = 2$ are plotted as functions of the number of processor elements, N_{pe} . The number of particles used for this measurement was $N = 4 \times 10^6$. We can see that the calculation time T scales almost linearly, for both $p = 1$ and $p = 2$.

In figure 6, the calculation time T measured with the code of order $p = 1$ and $p = 2$ are plotted as functions of the number of particles, N . The number

of processor elements used for this measurement was $N_{\text{pe}} = 16$. As theoretically expected, the calculation time T roughly scales as $N \log N$, except for rather small N ($N \lesssim 10^6$),

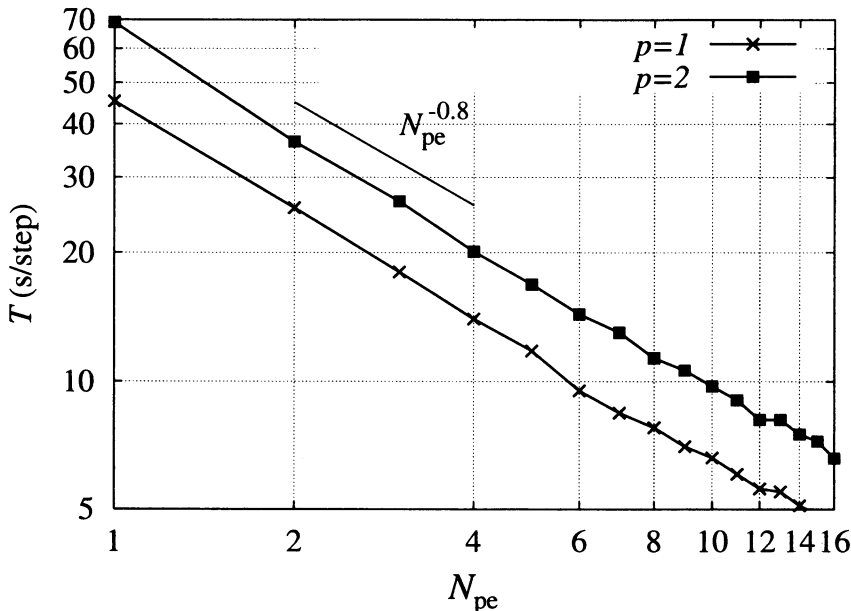


Figure 5. The calculation time T of the parallel code plotted against the number of processor elements N_{pe} . The results are for number of particles $N = 4 \times 10^6$. Meaning of the symbols are the same as those in figure 2.

4. Summary

We proposed P^2M^2 , a method to express multipole expansion by a distribution of pseudoparticles. Using this method, we implemented an arbitrary-order treecode on MDGRAPE-2. With this code MDGRAPE-2 accelerates the calculation by a factor between 20 (for $e \sim 10^{-2}$) to 150 (for $e \sim 10^{-6}$). We performed a dynamical simulation using this code and confirmed that it improves not only the instantaneous force calculation but the overall accuracy of the simulation. We parallelized the code and implemented it on an MDGRAPE-2 cluster, and found that the performance of our parallel code shows good scalability. We can conclude that the combination of GRAPE and the P^2M^2 treecode is an attractive choice for large N , high-accuracy simulations.

References

- Athanassoula, E., Bosma, A., Lambert, J.-C., Makino, J. 1998, MNRAS, 293, 369

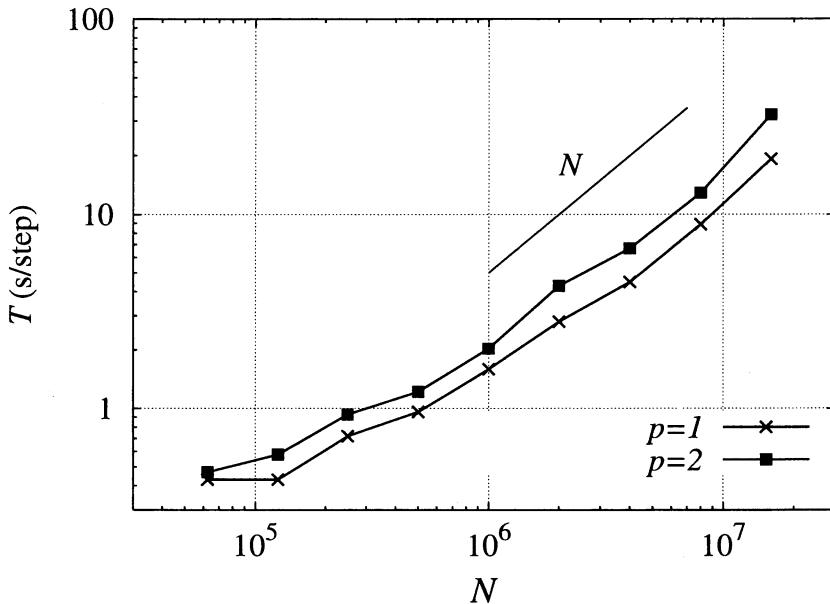


Figure 6. The calculation time T of the parallel code plotted against the number of particles N . The results are for 16 processor elements. Meaning of the symbols are the same as those in figure 2.

Barnes, J. E. 1990, *J. Comput. Phys.*, 87, 161

Barnes, J. E., Hut, P. 1986, *Nature*, 324, 446

Fukushige, T., Ito, T., Makino, J., Ebisuzaki, T., Sugimoto, D., Umemura, M. 1991, *PASJ*, 43, 841

Hernquist, L. 1990, *ApJ*, 356, 359

Hardin, R. H., Sloane, N. J. 1996, *Discrete and Computational Geometry*, 15, 429

Kawai, A., Fukushige, F., Makino, J., Taiji, M. 2000, *PASJ*, 52, 659

Makino, J. 1990, *J. Comput. Phys.*, 88, 393

Makino, J. 1991, *PASJ*, 43, 621

Makino, J., Taiji, M., Ebisuzaki, T., Sugimoto, D. 1997, *ApJ*, 480, 432

Makino, J., Taiji, M. 1998, *Scientific Simulations with Special-Purpose Computers — The GRAPE Systems* (Chichester: John Wiley and Sons)

Makino, J. 1999, *J. Comput. Phys.*, 151, 910

Narumi, T., Susukita, R., Koishi, T., Yasuoka, K., Furusawa, H., Kawai, A., & Ebisuzaki, T. 2000, in proceedings of SC2000 (ACM, in CD-ROM)

Sugimoto, D., Chikada, Y., Makino, J., Ito, T., Ebisuzaki, T., Umemura, M. 1990, *Nature*, 345, 33