

A new formulation for fast calculation of far field force in molecular dynamics simulations

Nguyen Hai Chau*

*Department of Information Technology, College of Technology, VNU
144 Xuan Thuy, Cau Giay, Hanoi, Vietnam*

Received 25 November 2006; received in revised form 7 August 2007

Abstract. We have developed a new formulation for fast calculation of far-field force of fast multipole method (FMM) in molecular dynamics simulations. FMM is a linear algorithm to calculate force for molecular dynamics simulations. GRAPE is a special-purpose computer dedicated to Coulombic force calculation. It runs 100-1000 times faster than normal computer at the same price. However FMM cannot be implemented directly on GRAPE. We have succeeded to implement FMM on GRAPE and developed a new formulation for far-field force calculation. Numerical tests show that the performance of FMM using our new formulation on GRAPE is approximately 2-5 times faster than that of FMM using conventional far field formulation.

1. Introduction

Molecular dynamics (MD) simulations often require high calculation cost. The most intensive part of MD is calculation of Coulombic force among particles (i.e. atoms and ions). In naive direct-summation algorithm, cost of the force calculation scales as $O(N^2)$, where N is the number of particles. In order to reduce the cost of force calculation, fast algorithms such as Barnes-Hut treecode [1] and fast multipole method [2] have been designed. Calculation cost of these algorithms are $O(N \cdot \log N)$ and $O(N)$, respectively. These fast algorithms are widely used in the field of MD simulation [3, 4].

Another approach to accelerate the force calculation is to use hardware dedicated to the calculation of inter-particle force. GRAPE (GRAvity Pipe) [5, 6] is one of the most widely used hardware of that kind. Figure 1 shows basic structure of a GRAPE system. It consists of a GRAPE processor board and a general-purpose computer (hereafter the host computer).

A typical GRAPE system performs force calculation 100-1000 times faster than conventional computers of the same price do. For small- N ($N \lesssim 10^5$) systems, combination of simple direct-summation algorithm and GRAPE is the fastest and simplest calculation scheme. However, for large- N systems, $O(N^2)$ direct-summation becomes expensive, even with GRAPE hardware. Combination of a fast algorithm and fast hardware will deliver extremely high performance for large N . Makino et al [7] have successfully implemented a modified treecode [8] on GRAPE, and achieved a factor of 30-50 speed up.

* Tel: 84-4-7547813.

E-mail: chaunh@vnu.edu.vn

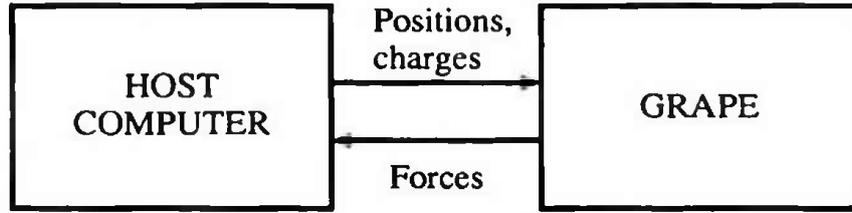


Figure 1. Basic structure of a GRAPE system.

Implementation of FMM on dedicated hardware of similar kind (MD-ENGINE) has been reported, but its performance is rather modest [9]. This is mainly because the hardware limitation. Since dedicated hardware can calculate the particle force only, they cannot handle multipole and local expansions. Therefore only a small fraction of the calculation procedure in the FMM can be performed on such hardware, and the speed up gain remains rather modest. An outstanding problem is how to perform a large or all fraction of FMM's calculation procedure on GRAPE.

We have implemented FMM on GRAPE and achieved significant speedup [10]. However we have not succeeded to put far field calculation part of FMM to GRAPE. This fact limits the performance of FMM on GRAPE.

In this paper we describe our new formulation to speed up far field force calculation – a significant calculation part of FMM on GRAPE. Remaining parts of the paper are organized as follows. In section 2 we give a summary of the FMM and related algorithms as well as describe the implementation of our FMM code and its limitation. Section 3 presents our new formulation. Results of numerical tests are shown in section 4. Section 5 summarizes.

2. FMM and its variant implementations

2.1. FMM

The FMM [2, 11] is an approximate algorithm to calculate force among particles. In the case of close-to-uniform distribution, its computation complexity is $O(N)$. This scaling is achieved by approximation of force using the multipole and local expansion technique.

Figure 2 shows schematic idea of force approximation in the FMM. The force from a group of distant particles are approximated by a multipole expansion. At an observation point, the multipole expansion is converted to local expansion. The local expansion is evaluated by each particle around the observation point. Hierarchical tree structure is used for grouping of the particles [2, 11].

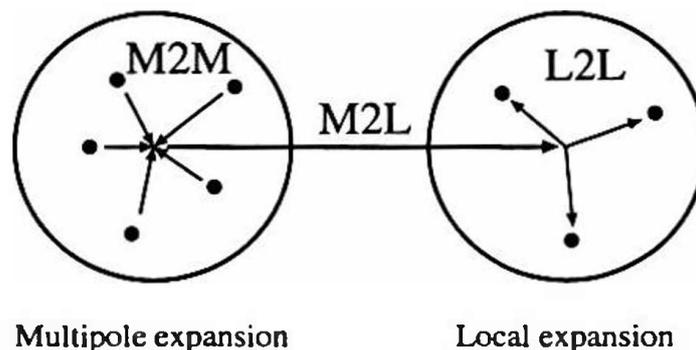


Figure 2. Schematic idea of force approximation in FMM.

2.2 Anderson's method

Anderson [12] proposed a variant of the FMM using a new formulation of the multipole and local expansions. His method is based on the Poisson's formulae. In order to use these formulae as replacements of the multipole and local expansions, Anderson proposed discrete versions of them as follows. When potential on the surface of a sphere of radius a is given, the potential Φ at position $\vec{r} = (r, \phi, \theta)$ is expressed as:

$$\Phi(\vec{r}) \approx \sum_{i=1}^K \sum_{n=0}^p (2n + 1) \left(\frac{a}{r}\right)^{n+1} P_n \left(\frac{\vec{s}_i \cdot \vec{r}}{r}\right) \Phi(a\vec{s}_i) w_i \tag{1}$$

for $r \geq a$ (outer expansion) and

$$\Phi(\vec{r}) \approx \sum_{i=1}^K \sum_{n=0}^p (2n + 1) \left(\frac{r}{a}\right)^n P_n \left(\frac{\vec{s}_i \cdot \vec{r}}{r}\right) \Phi(a\vec{s}_i) w_i \tag{2}$$

for $r \leq a$ (inner expansion). The function P_n denotes the n -th Legendre polynomial. Here w_i are constant weight values and p is the number of untruncated terms. Hereafter we refer p as expansion order. Anderson's method uses Eq. (1) and (2) for M2M and L2L transitions, respectively. The procedures of other stages are the same as that of the original FMM. Note that Anderson used spherical t -design [13] to obtain Eq. (1) and (2). Examples of spherical t -design is available at <http://www.research.att.com/njas/sphdesigns/>.

2.3 Pseudoparticle multipole method

Makino [14] proposed the pseudoparticle multipole method (P^2M^2). The advantage of his method is that the expansions can be evaluated using GRAPE.

Makino's idea is very similar to Anderson's. Both methods uses discrete quantity to approximate the potential field of the original distribution of the particles. The difference is that P^2M^2 uses the distribution of point charges, while the Anderson's method uses potential values. In the case of P^2M^2 , the potential is expressed by point charges as given below, and thus it can be evaluated using GRAPE.

$$Q_j = \sum_{i=1}^N q_i \sum_{l=0}^p \frac{2l + 1}{K} \left(\frac{r_i}{a}\right)^l P_l(\cos \gamma_{ij}), \tag{3}$$

where Q_j is charge of pseudoparticle, $\vec{r}_i = (r_i, \phi, \theta)$ is position of physical particle, γ_{ij} is angle between \vec{r}_i and position vector \vec{R}_j of the j -th pseudoparticle [14].

Implementation of the FMM on GRAPE In this section, we briefly describes our implementation on GRAPE [10]. The FMM consists of five stages, namely, tree construction, M2M transition, M2L conversion, L2L transition, and force evaluation. Force-evaluation stage consists of near field and far field evaluation parts.

In the case of original FMM, only the near field part of the force-evaluation stage can be performed on GRAPE. In our implementation (hereafter code A), we modified the original FMM so that GRAPE can handle M2L conversion stage, which is most time consuming. Table 1 summarizes mathematical expressions and operations used at each calculation stage. In the following we describe stages of the code A.

Table 1. Mathematical expressions and operations used in our implementation of the code A [10]. Bold parts run on GRAPE.

	Original [11]	Code A (section 2)
M2M	multipole expansion	P²M²
M2L	M2L conversion formula	evaluation of pseudoparticle potential
L2L	local expansion	Anderson's method
Near field force	evaluation of physical-particle force	
Far field force	evaluation of local expansion	Eq. (4)

The tree construction stage has no change. It is performed in the same way as in the original FMM.

At the M2M transition stage, we compute positions and charges of pseudoparticles, instead of forming multipole expansion as in the original FMM. This process is totally done on the host computer.

The M2L conversion stage is done on GRAPE. Difference from the original FMM is that we do not use the formula to convert multipole expansion to local expansion. We directly calculate potential values due to pseudoparticles.

The L2L transition is done in the same way as Anderson has done using Eq. (2).

The near field contribution is directly calculated by evaluating the particle-particle force. GRAPE handles this part.

Using Eq. (2), we obtain the far field potential on a particle at position \vec{r} . Consequently, far field force is calculated using derivative of Eq. (2):

$$-\nabla\Phi(\vec{r}) = \sum_{i=1}^K \sum_{n=0}^p \left(n\vec{r}P_n(u) + \frac{u\vec{r} - \vec{s}_i r}{\sqrt{1-u^2}} \nabla P_n(u) \right) (2n+1) \frac{r^{n-2}}{a^n} g(a\vec{s}_i) w_i, \quad (4)$$

where $u = \vec{s}_i \cdot \vec{r}/r$. All the calculation at this stage is done on the host computer.

With the modification to original FMM described above, we have succeeded to put the bottleneck, namely, the M2L conversion stage, on GRAPE. The overall calculation of the FMM is significantly accelerated. Now the most expensive part is the far field force evaluation. A new bottleneck appears. Eq. (4) is complicated and evaluation of it takes rather big fraction of the overall calculation time [10].

If we can convert a set of potential values into a set of pseudoparticles at marginal calculation cost, force from those pseudoparticles can be evaluated on GRAPE, and the new bottleneck will disappear. In order for this conversion, we have newly developed a conversion procedure (hereafter A2P conversion) presented in section 3.

3. A new formulation for fast calculation of far field force

Eq. (3) gives solution for outer expansion of P²M². Using a similar approach, we obtained solution for inner expansion as:

$$Q_j = \sum_{i=1}^N q_i \sum_{l=0}^p \frac{2l+1}{K} \left(\frac{a}{r_i} \right)^{l+1} P_l(\cos \gamma_{ij}). \quad (5)$$

In the following we give derivation procedure for Eq. (5). The local expansion of the potential $\Phi(\vec{r})$ is expressed as

$$\Phi(\vec{r}) = 4\pi \sum_{l=0}^p \sum_{m=-l}^l \beta_l^m r^l Y_l^m(\theta, \phi). \tag{6}$$

Here, $Y_l^m(\theta, \phi)$ is the spherical harmonics and β_l^m is the expansion coefficient. In order to approximate the potential field due to the distribution of N particles, the coefficients should satisfy

$$\beta_l^m = \frac{1}{2l+1} \sum_{i=1}^N q_i \frac{1}{r_i^{l+1}} Y_l^{m*}(\theta_i, \phi_i), \tag{7}$$

where q_i and $\vec{r}_i = (r_i, \theta_i, \phi_i)$ are the charges and positions of the particles, and * denotes the complex conjugate.

In order to reproduce the expansion $\Phi(\vec{r})$ up to p -th order, the charges Q_j and the positions $\vec{R}_j = (R_j, \theta_j, \phi_j)$ of pseudoparticles must satisfy

$$\beta_l^m = \frac{1}{2l+1} \sum_{j=1}^K Q_j \frac{1}{R_j^{l+1}} Y_l^{m*}(\theta_j, \phi_j) \tag{8}$$

for all $(p+1)^2$ combinations of l and m in the range of $0 \leq l \leq p$ and $-l \leq m \leq l$. Here K is the number of pseudo particles.

Following Makino's approach [14], we restrict the distribution of pseudoparticles to the surface of a sphere centered at the origin. With this restriction, the coefficients of local expansion generated by the pseudoparticles are expressed as

$$\beta_l^m = \frac{1}{(2l+1)b^{l+1}} \sum_{j=1}^K Q_j Y_l^{m*}(\theta_j, \phi_j), \tag{9}$$

where b is the radius of the sphere. If we consider the limit of infinite K , Eq. (9) is replaced by

$$\beta_l^m = \frac{1}{(2l+1)b^{l+1}} \int_S \rho(a, \theta, \phi) Y_l^{m*}(\theta, \phi) ds. \tag{10}$$

Here S is the surface of a unit sphere, and ρ is the continuous charge representation of pseudoparticle. In this limit, the charge distribution is obtained by the inverse transform of spherical harmonics expansion as follows:

$$\rho(a, \theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l (2l+1)b^{l-1} \beta_l^m Y_l^m(\theta, \phi). \tag{11}$$

We can discretize ρ using the spherical t -design. In other words, the spherical t -design gives a distribution of pseudoparticles over which numerical integration retains the orthogonality of spherical harmonics up to p -th order. The charges of the pseudoparticles are then obtained as

$$Q_j = \frac{4\pi}{K} \sum_{l=0}^p \sum_{m=-l}^l (2l+1)b^{l+1} \beta_l^m Y_l^m(\theta_j, \phi_j). \tag{12}$$

This equation gives the charges Q_j of pseudoparticles from the expansion coefficients of physical particles β_l^m . In practice, we can directly calculate Q_j from the charges q_i and the positions \vec{r}_i of physical particles.

Combining Eq. (7) and Eq. (12), Q_j is expressed as

$$Q_j = \frac{4\pi}{K} \sum_{l=0}^p \sum_{m=-l}^l \sum_{i=1}^N q_i \left(\frac{b}{r_i}\right)^{l+1} Y_l^m(\theta_j, \phi_j) Y_l^{m*}(\theta_i, \phi_i). \quad (13)$$

Using the addition theorem of spherical harmonics, we can simplify this equation and obtain the formula to give Q_j from q_i and \vec{r}_i :

$$Q_j = \sum_{i=1}^N q_i \sum_{l=0}^p \frac{2l+1}{K} \left(\frac{b}{r_i}\right)^{l+1} P_l(\cos \gamma_{ij}). \quad (14)$$

Using the new formula (14), we have implemented yet another version of FMM (hereafter code B). Table 2 describes stages in code B. In the code B, we use A2P conversion to obtain a distribution of pseudoparticles that reproduces the potential field given by Anderson's inner expansion. Once the distribution of pseudoparticles is obtained, L2L stage can be performed using inner-P²M² formula (Eq. (5)), and then the force evaluation stage is totally done on GRAPE (see table 2). Procedure of A2P conversion is as follows.

Table 2. Mathematical expressions and operations used in the code B. Bold parts run on GRAPE.

	Original [11]	Code B (section 3)
M2M	multipole expansion	P ² M ²
M2L	M2L conversion formula	evaluation of pseudoparticle potential
L2L	local expansion	P ² M ²
Near field force	evaluation of physical-particle force	
Far field force	evaluation of local expansion	evaluation of pseudoparticle force

At the first step, we distribute pseudoparticles on the surface of a sphere with radius b using the spherical t -design. Here, b should be larger than the radius of the sphere a on which Anderson's potential values $\Phi(a\vec{s}_i)$ are defined. According to Eq. (5), it is guaranteed that we can adjust the charge of the pseudoparticles so that $\Phi(a\vec{s}_i)$ are reproduced. Therefore, the relation

$$\sum_{j=1}^K \frac{Q_j}{|\vec{R}_j - a\vec{s}_i|} = \Phi(a\vec{s}_i) \quad (15)$$

should be satisfied for all $i = 1..K$. Using a matrix $\mathcal{R} = \{1/|\vec{R}_j - a\vec{s}_i|\}$ and vectors $\vec{Q} = {}^T[Q_1, Q_2, \dots, Q_K]$ and $\vec{P} = {}^T[\Phi(a\vec{s}_1), \Phi(a\vec{s}_2), \dots, \Phi(a\vec{s}_K)]$, we can rewrite Eq. (15) as

$$\mathcal{R}\vec{Q} = \vec{P}. \quad (16)$$

In the next step, we solve the linear Eq. (16) to obtain charges Q_j . By numerical experiment, we found that appropriate value of radius b is about 6.0, for particles inside a cell with side length 1.0. Anderson specified in his paper [12] that a should be about 0.4.

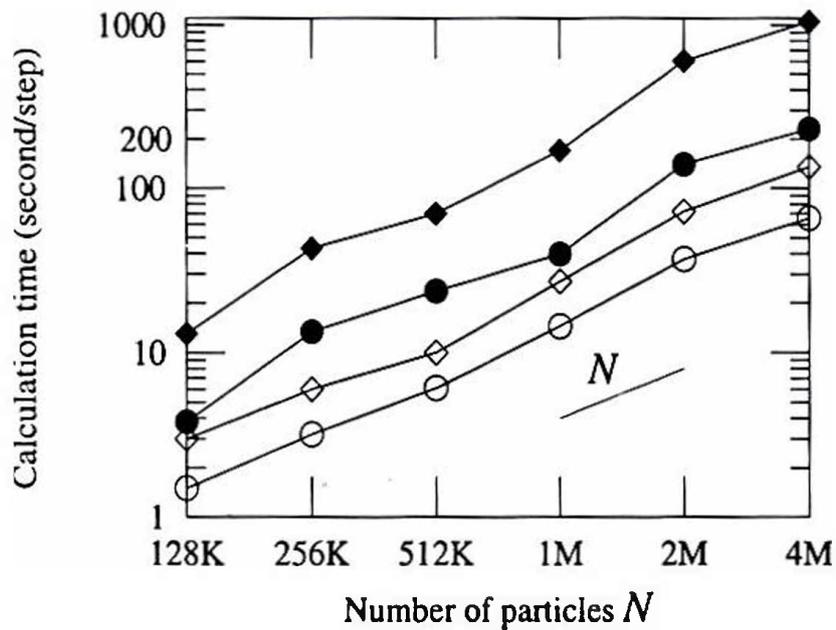


Figure 3. Comparison of the code A and B. Squares are performance of code A on MDGRAPE-2. Circles are that of code B. Open and filled symbols are for low ($p = 1$) and high accuracy ($p = 5$), respectively.

4. Numerical results

Here we show the performance of the FMM code B and compare performance of the code A and B measured on MDGRAPE-2 [15]. MDGRAPE-2 is one of the latest hardware of the GRAPE series. It is developed for MD simulation.

Our test system consists of one MDGRAPE-2 board (16 pipelines, 48GFlops) and a host computer Pentium 4 2.2GHz, Intel D850 motherboard.

In the tests, we distributed particles uniformly within a unit cube centered at origin, and evaluated force on all particles. We measured the calculation time at high ($p = 5$) and low ($p = 1$) accuracy, with and without GRAPE. The finest refinement level l_{max} is set to $l_{max} = 4$ and 5, for runs with and without GRAPE, respectively. These values are chosen so that the overall calculation time is minimized. Result is shown in figure 3. Notation K and M on the figures are 1024 and 1024*1024, respectively.

In figure 3 we compare the performance of code A and code B on our test computer system. Since code B uses the A2P conversion procedure, it runs approximately faster than code A 2 times for low accuracy and 5 times for high accuracy.

5. Summary

We have developed a new formulation and a new calculation procedure to speed up the calculation of far field force in FMM implementation on special-purpose hardware GRAPE. Employing the new formulation, our new code (code B) is of higher performance than the treecode at high accuracy. The numerical results show that the code B performs approximately 2-5 times faster than the code A [10] which uses conventional formulation of calculation.

Acknowledgements. This work is supported by Advanced Computing Center, Institute of Physical and Chemical Research (RIKEN), Japan; Institute of Information Technology, Vietnam National University, Hanoi under QCT.05.07 project; and College of Technology, Vietnam National University, Hanoi under QC.05.01 project.

References

- [1] J.E. Barnes, P. Hut, A hierarchical $O(N \log N)$ force-calculation algorithm, *Nature* 324 (1986) 446.
- [2] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, *Journal of Computational Physics* 73 (1987) 325.
- [3] P. Lakshminarasimhulu, J.D. Madura, A cell multipole based domain decomposition algorithm for molecular dynamics simulation of systems of arbitrary shape, *Computer Physics Communications* 144 (2002) 141.
- [4] J.A. Lupo, Z.Q. Wang, A.M. McKenney, R. Pachter, W. Mattson, A large scale molecular dynamics simulation code using the fast multipole algorithm (FMD): performance and application, *Journal of Molecular Graphics and Modelling* 21 (2002) 89.
- [5] D. Sugimoto, Y. Chikada, J. Makino, T. Ito, T. Ebisuzaki, M. Umemura, A special-purpose computer for gravitational many-body problems, *Nature* 345 (1990) 33.
- [6] J. Makino, M. Taiji, *Scientific Simulations with Special-Purpose Computers - The GRAPE Systems* (Chichester: John Wiley and Sons, 1998).
- [7] J. Makino, Treecode with a special-purpose processor, *Publ. Astron. Soc. Japan* 43 (1991) 621.
- [8] J.E. Barnes, A modified tree code: Don't laugh; It runs, *Journal of Computational Physics* 87 (1990) 161.
- [9] T. Amisaki, S. Toyoda, H. Miyagawa, K. Kitamura, Development of hardware accelerator for molecular dynamics simulations: a computation board that calculates nonbonded interactions in cooperation with fast multipole method. *Journal of Computational Chemistry* 24 (2003) 582.
- [10] N.H. Chau, A. Kawai, T. Ebisuzaki, Implementation of fast multipole algorithm on special-purpose computer MDGRAPE-2, *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics SCI2002*, (Orlando, Colorado, USA, July 14-18, 2002) 477.
- [11] L. Greengard, V. Rokhlin, A new version of the fast multipole method for the Laplace equation in three dimensions, *Acta Numerica* 6 (1997) 229.
- [12] C.R. Anderson, An implementation of the fast multipole method without multipoles, *SIAM J. Sci. Stat. Comput.* 13 (1992) 923.
- [13] R.H. Hardin, N.J.A. Sloane, McLaren's improve snub cube and other new spherical design in three dimensions, *Discrete and Computational Geometry* 15 (1996) 429.
- [14] J. Makino, Yet another fast multipole method without multipoles - pseudoparticle multipole method, *Journal of Computational Physics* 151 (1999) 910.
- [15] R. Susukita, T. Ebisuzaki, B.G. Elmegreen, H. Furusawa, K. Kato, A. Kawai, Y. Kobayashi, T. Koishi, G.D. McNiven, T. Narumi, K. Yasuoka, Hardware accelerator for molecular dynamics: MDGRAPE-2, *Computer Physics Communications* 155 (2003) 115.