Fast, accurate molecular dynamics simulations for surfaces and membranes

Masaaki Kawata* and Umpei Nagashima**

Grid Technology Research Center,
National Institute of Advanced Industrial Science and Technology,
AIST Tsukuba Central 2, Tsukuba, 305-8568 JAPAN
* m.kawata@aist.go.jp, ** u.nagashima@aist.go.jp

ABSTRACT

We developed a fast, accurate method for calculating Coulomb interactions for three-dimensional systems with two-dimensional (2D) periodicity (quasi-2D systems) and implemented this method into a molecular dynamics (MD) calculation. The algorithm is based on a computationally-efficient method we previously developed referred to as the 2D-PME method. We present details of this new 2D-PME method, focusing on the techniques for increasing computational speed, and implementation into an MD calculation. We also discuss optimization of the new 2D-PME method for parallelization. For the same accuracy, the new 2D-PME method is about 40 % faster than the previous 2D-PME method. Compared with the previous algorithm, the new algorithm has the advantage that it is optimized for parallel calculations. The new method is therefore useful for extending the frontier of MD calculations for surfaces and membranes to realistic nano-scale systems.

Keywords: Molecular dynamics, Coulomb interactions, surface, membrane, simulation.

1 INTRODUCTION

Numerical simulations are useful for investigating phenomena in nano-scale structures.[1] Although atomistic simulations, such as molecular dynamics (MD) simulations, have been done for bulk materials, i.e., three-dimensional (3D) systems with 3D periodicity, the time and length scales possible with atomistic simulations for quasi two-dimensional (2D) systems, i.e., 3D systems with 2D periodicity, has been limited.[2] Such quasi-2D systems are model systems for surfaces, self-assembled monolayer membranes, and biological membranes (see Fig. 1).

The size of systems treated in MD calculations and the accuracy of the calculations depend on the computational efficiency and accuracy of calculating the Coulomb interactions in the systems. Improper calculation of Coulomb interactions, such as the use of "cutoff" for the long-range Coulomb interactions, decreases the accuracy of the MD calculations; in the worst case simulating physically unrealistic behavior of the system. To overcome the artifacts associated with the finite size of the simulation box, Ewald summations have been widely used to approximate Coulomb interactions.[3]

For bulk materials, fast algorithms for evaluating the Ewald summations have been developed and are widely used in MD simulations, such as the particle-mesh Ewald (3D-PME) method,[4] whereas for quasi-2D systems, although accurate, approximate formulations for calculating the Coulomb interactions have been developed,[5] prohibitive computational costs or limited range of validity of the inherent approximations have limited the applications to MD simulations. We therefore developed a computationally efficient algorithm to implement the Ewald summations for quasi-2D systems, which maintains acceptable accuracy and range of application.[6] We refer to this new method as the 2D-PME method. In this work, we present an improved algorithm, i.e., faster and optimized for parallel calculations, for the 2D-PME method and details of the implementation of the method into the MD calculations. We show this implementation for both for single- and for parallel-processor calculations.

2 NUMERICAL METHOD

In this section we summarize the 2D-PME method and its implementation into the MD calculation. For the details of the MD components other than Coulomb interactions (i.e., van der Waals and bonded interactions), see Ref.[1].

2.1 2D particle-mesh Ewald (2D-PME) method

For a quasi-2D system of N charged particles, $\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_N$ satisfying the neutrality condition, $q_1 + q_2 + \cdots + q_N = 0$, the Ewald summations of the Coulomb potential energy, Φ, can be expressed as a summation of a rapid convergence sums in real-space, Φ^r, and in reciprocal-space without the $\mathbf{k} = \mathbf{0}$ term, $\Phi^k_{\mathbf{k}\neq\mathbf{0}}$, with the $\mathbf{k} = \mathbf{0}$ term, $\Phi^k_{\mathbf{k}=\mathbf{0}}$, and with the self-correction term, Φ^s, as

$$\Phi = \frac{1}{2} \sum_{\mathbf{h}_x, \mathbf{h}_y} {}' \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{q_i q_j}{\left| \mathbf{r}_i - \mathbf{r}_j + \mathbf{h}_x + \mathbf{h}_y \right|}$$
$$\approx \Phi^r + \Phi^k_{\mathbf{k}\neq\mathbf{0}} + \Phi^k_{\mathbf{k}=\mathbf{0}} + \Phi^s, \qquad (1)$$

where a Gaussian charge distribution is used, such as $\rho(\mathbf{r}_i) = q_i \alpha^3 \pi^{-3/2} \exp(-\alpha^2 |\mathbf{r}_i|^2)$ with a screening parameter of the charge distribution α. The prime in the first summation of Eq. (1) indicates that terms with $i = j$ are omitted when $\mathbf{h}_x + \mathbf{h}_y = \mathbf{0}$. $\mathbf{h}_x$ and $\mathbf{h}_y$ are the components of the 2D lattice vectors in the x and y direction, respectively, and $\mathbf{k}$ is the 2D reciprocal vector. In Eq. (1). Φ^r, $\Phi^k_{\mathbf{k}\neq\mathbf{0}}$, $\Phi^k_{\mathbf{k}=\mathbf{0}}$, and Φ^s are defined as

$$\Phi^r = \frac{1}{2} \sum_{\mathbf{h}_x, \mathbf{h}_y}{}' \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{h}_x + \mathbf{h}_y|} \operatorname{erfc}\left(\alpha |\mathbf{r}_{ij} + \mathbf{h}_x + \mathbf{h}_y|\right), \quad (2)$$

$$\Phi^k_{\mathbf{k}\neq\mathbf{0}} = \frac{1}{|\mathbf{h}_x \times \mathbf{h}_y|} \sum_{t_x=0}^{G_x-1} \sum_{t_y=0}^{G_y-1} \sum_{t_z=-\infty}^{\infty} Q(t_x, t_y, t_z) \times \Psi \bullet Q(t_x, t_y, t_z) \quad (3)$$

$$\Phi^k_{\mathbf{k}=\mathbf{0}} = \sum_{i=1}^{N} q_i B\left[\Phi^k_{\mathbf{k}=\mathbf{0}\ g}, z_i\right], \quad (4)$$

$$\Phi^s = -\frac{\alpha}{\sqrt{\pi}} \sum_{i=1}^{N} q_i^2 . \quad (5)$$

In Eq. (2), $\mathbf{r}_{ij} \equiv \mathbf{r}_i - \mathbf{r}_j$ and erfc represents the complimentary error function. In Eq. (3), $Q(t_x, t_y, t_z)$ is defined as

$$Q(t_x, t_y, t_z) \equiv \sum_{j=1}^{N} q_j \left\{ \sum_{l_x=-\infty}^{\infty} M_{n_x}\left[(\mathbf{u}_j)_x - t_x - l_x G_x\right] \times \sum_{l_y=-\infty}^{\infty} M_{n_y}\left[(\mathbf{u}_j)_y - t_y - l_y G_y\right] \times M_{n_z}\left[z_j - t_z\right] \right\}, \quad (6)$$

and $\Psi(t_x, t_y, t_z)$ is defined as

$$\Psi(t_x, t_y, t_z) \equiv \sum_{m_x=0}^{G_x-1} \sum_{m_y=0}^{G_y-1} \int_{-\infty}^{\infty} dh \left|C(h, n_z)\right|^2 \times \left|C\left(\frac{2\pi m_x}{G_x}, n_x\right)\right|^2 \left|C\left(\frac{2\pi m_y}{G_y}, n_y\right)\right|^2 \times \frac{1}{4\pi^2\left(m_x^2 + m_y^2\right) + h^2} \exp\left\{-\frac{1}{4\alpha^2}\left[4\pi^2\left(m_x^2 + m_y^2\right) + h^2\right]\right\} \times \exp\left(\mathrm{i}\frac{2\pi m_x t_x}{G_x} + \mathrm{i}\frac{2\pi m_y t_y}{G_y} + \mathrm{i}\, h t_z\right), \quad (7)$$

where G_x and G_y are the number of grid points for the fast Fourier transform (FFT), i.e., the dimensions of the FFT, in the x and y directions, respectively, and $\bullet$ means convolution. $\mathbf{u}_j \quad (j = 1, \ldots, N)$ represents fractional coordinates in the (x, y) plane, which is defined as

$$\mathbf{u}_j = \left(G_x \mathbf{m}_x{}^T \cdot \mathbf{r}'_j, \quad G_y \mathbf{m}_y{}^T \cdot \mathbf{r}'_j\right), \quad (8)$$

where $\mathbf{r}'_j$ is the projection of $\mathbf{r}_j$ onto the (x, y) plane, $\mathbf{m}_\lambda \equiv \mathbf{k}_\lambda / 2\pi$ with $\lambda = x, y$, M_w is the B-spline function, $C(v, w)$ is defined as

$$C(v, w) \equiv \frac{\exp\left[\mathrm{i}\,(w-1)\,v\right]}{\sum_{t=0}^{w-2} M_w(t+1) \exp\left(\mathrm{i}\ v\ t\right)}, \quad (9)$$

and n_λ is the order of the B-splines in the λ direction.[7]

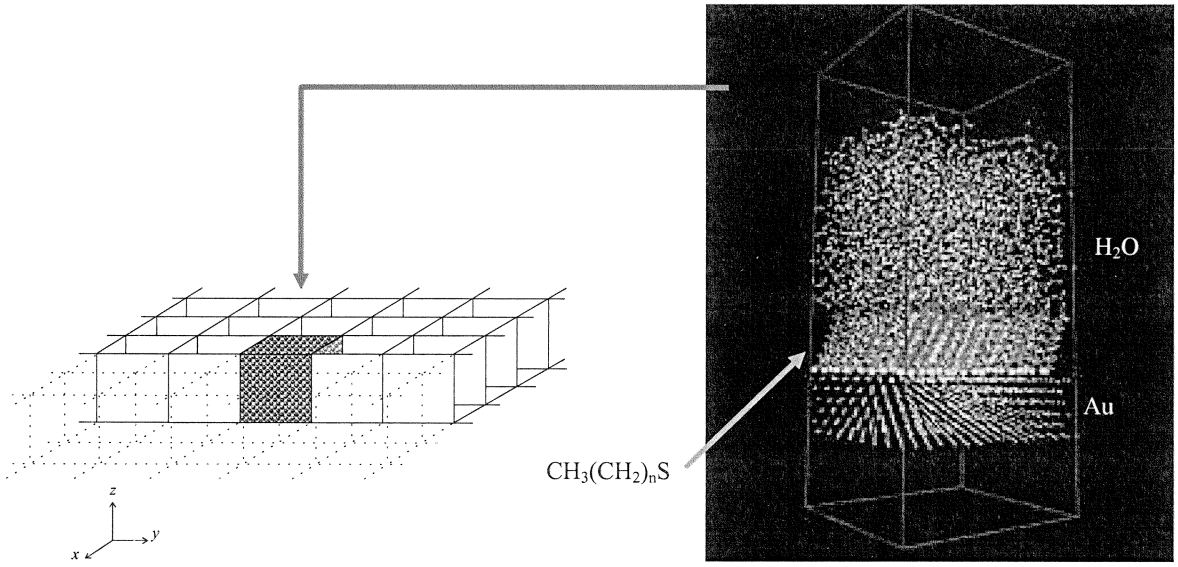

Figure 1: (Left) Quasi two-dimensional simulation box, i.e., three-dimensional box with two-dimensional periodicity in the (x, y) directions and with non-periodicity in the z direction. Original particles are contained in the central box with sides of length L_x, L_y, and L_z in the x, y, and z directions, respectively. (Right) Self-assembled monolayer membrane system. Images of the simulation box are repeated in the (x,y) directions.

In Eq. (4), $B\left[\Phi^{k}_{\mathbf{k}=\mathbf{0}g}, z_i\right]$ represents the Coulomb potential energy at position z_i, interpolated with a B-spline interpolation by using the values of $\Phi^{k}_{\mathbf{k}=\mathbf{0}1},\ldots,\Phi^{k}_{\mathbf{k}=\mathbf{0}L}$ on the gth grid point $(g=1,\ldots,L)$ in the z direction, which can be expressed as

$$\Phi^{k}_{\mathbf{k}=\mathbf{0}\ g} = \frac{\pi}{2\left|\mathbf{h}_x \times \mathbf{h}_y\right|}\sum_{j=1}^{N} q_j \Lambda\left(\alpha, z_{gj}\right), \tag{10}$$

where $z_{gj} \equiv z_g - z_j$, and $\Lambda(\alpha,z)$ is defined as

$$\Lambda(\alpha,z) \equiv -2\left[\frac{1}{\alpha\sqrt{\pi}}\exp\left[-(\alpha z)^2\right] + z\,\mathrm{erf}(\alpha z)\right], \tag{11}$$

where erf is the error function.

The Coulomb force, $\mathbf{F}$, is also a summation of the contributions from the real-space sum, $\mathbf{F}^r$, and from the reciprocal-space sums without and with the $\mathbf{k}=\mathbf{0}$ term, $\mathbf{F}^{k}_{\mathbf{k}\neq\mathbf{0}}$ and $\mathbf{F}^{k}_{\mathbf{k}=\mathbf{0}}$, respectively. Each contribution is easily derived by differentiating the corresponding potential energy term.

2.2 Implementation of 2D-PME method into the MD calculations

Embedding the 2D-PME method into MD calculations with 2D periodic boundary conditions is straightforward on a single processor, because the functions are simply inserted into the calculation of the Coulomb interactions. To yield computationally efficient MD calculations with the 2D-PME method, similar to the other Ewald methods, the load balance between calculating the real and the reciprocal-space sums should be optimized for desired accuracy. The method for determining the optimal set of parameters to implement the 2D-PME method is described in detail in Section III in Ref. [6(a)]. Unlike the 3D-PME method, where 3D-FFTs are directly used, in the 2D-PME method 2D-FFTs and 1D Fourier integrals (1D-FI) are used. Because of the exponential factor in Eq. (7), however, 2D-FFTs and 1D-FIs can be approximately evaluated by using 3D-FFTs combined with additional arithmetic operations[8] (see Appendix in Ref.[6(a)] for pseudo code in C) .

Figure 2 shows a flow chart of the parallel MD calculations with the 2D-PME method. In the parallel MD calculations, compared with the calculations on a single processor, additional operations that redistribute the coordinates of particles over all nodes (shaded part in Fig. 2) are required to implement the parallel 2D-PME method. This procedure is similar to that required for the parallel 3D-PME method. The double-box items shown in Fig. 2 are scaled as $1/P$, where P is the number of processors. This implies that the reduction of computational time is proportional to P. The 2D-FFT and 1D-FI (shown as single-box items in Fig. 2) can be evaluated by using parallel 3D-FFTs similar to the calculations on a single processor. For the data structure needed to calculate parallel FFTs, the entire system should be decomposed

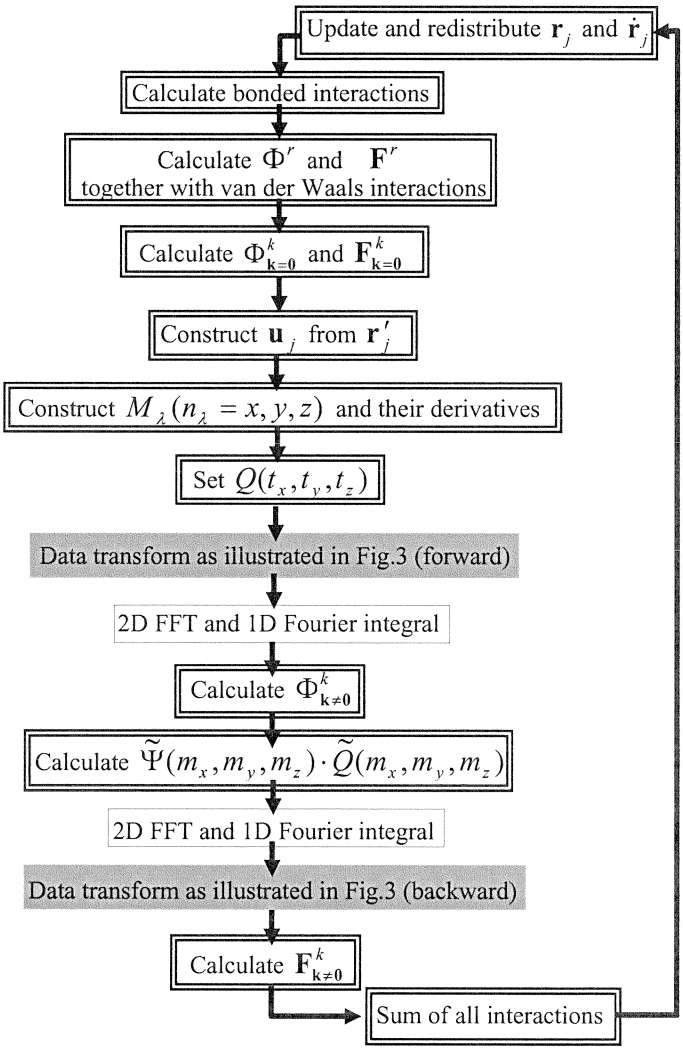

Figure 2. Flow chart of parallel MD calculations with the 2D-PME method.

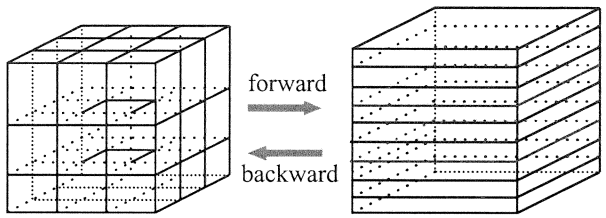

Figure 3. Data transformation between the space decomposition conventionally used in parallel MD calculations, and the slab decomposition used for the parallel FFT. To reduce the complexity of the figure, in the right box the number of slabs shown is reduced to one third the number actually used.

into slabs, as shown in the right part of Fig. 3. However, in the space-decomposition scheme typically used in MD calculations, the entire simulation is decomposed equally among the x, y, and z directions (shown in the left part in Fig. 3) and each processor is responsible for the calculations of the data in the decomposed cell. Because of the non-scalability of the transformations, the forward and backward data transformations shown in Fig. 3 are the bottleneck of the parallel MD calculations with the 2D-PME method.

3 PERFORMANCE EVALUATION

We used FFTW[9] to calculate the FFTs. The system used to evaluate the performance of the 2D-PME method is described elsewhere.[6] Compared with the set of parameters used previously for the 2D-PME method (see Refs. [6]), the optimal set of parameters required for the single-processor version of the new 2D-PME method changed, because the new algorithm requires less computational effort than the previous algorithm for calculating the reciprocal-space sums, and because the optimized load balance between the real- and reciprocal-space sums resulted in the increase of α .

Table 1 shows the performance of the MD calculations with the 2D-PME method on a single processor. For the same accuracy, the calculation with the new 2D-PME method requires about 40% less computational time than that with the previous 2D-PME method.

Table 1: CPU time[a] for a single-step MD calculation for a given accuracy, by using the methods described in Ref. [5] and the 2D-PME method [6].

	Water system
Number of particles	2928
L_x=L_y (nm)	3.215
L_z (nm)	3.215
Method in Ref. [5(a-c)] (s)	1099.18
Our previous method (s)	3.11
Our new method (s)	2.12

[a] CPU time on Compaq Alpha Station XP1000 (Alpha21264 667MHz).

For parallel MD calculations, we used the set of parameters optimized for calculations on a single processor. The computational performance of the parallel MD calculations strongly depends on the efficiency of the data transformations shown in Fig. 3 and on the computational efficiency of the parallel FFT. The performance depends not only on the algorithms used in the calculations, but also on the speed of the network used to link the multiple nodes in the parallel calculation. We evaluated the speedup factor of parallel MD calculations made on an IBM RS/6000 SP with 128 nodes (CPU: 200-MHz Power 3, 2-way SMP/node). As shown in Fig. 4, the new 2D-PME method is useful for parallelization of MD calculations, although higher speed networks are required to further increase the computational speed of parallel calculations.

The new method implemented in the MD calculations can extend the frontier of accurate atomistic simulations for surfaces and membranes, which should permit realistic modeling of nano-scale systems.

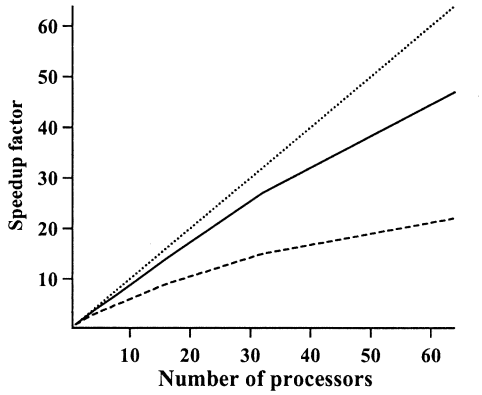

Figure 4: Speedup factor of the parallel MD calculations for the quasi-2D systems by using the 2D-PME method. The solid line is for calculations with an SP switch (300MB/sec BI-Direction) with user space protocol, and the dashed line is for calculations with internet protocol. The dotted line is the ideal speedup factor, assuming an infinitely fast network connecting the nodes.

REFERENCES

[1] M. P. Allen and D. J. Tildesley, "Computer Simulation of Liquids", Oxford University Press, 1989.

[2] D. M. Heyes, "The Liquid State", Wiley, 1998.

[3] P. Ewald, Ann. Phys. (Leipzig) 64, 253, 1921.

[4] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, J.Chem.Phys. 103, 8577, 1995.

[5] (a)D. E. Parry, Surf. Sci. 49, 433, 1975; (b)D. M. Heyes, M. Barber, and J. H. R. Clarke, J. Chem. Soc. Faraday Trans. II 73, 1485, 1977; (c)S. W. de Leeuw and J. W. Perram, Mol. Phys. 37, 1313, 1979; (d)F. E. Harris, Int. J. Quantum Chem. 68, 385, 1998; (e)J. Hautman and M. L. Klein, Mol. Phys. 75, 379, 1992.

[6] (a)M. Kawata, M. Mikami, and U. Nagashima, J. Chem. Phys. 116, 3430, 2002; (b)M. Kawata and M. Mikami, Chem. Phys. Lett. 340, 157, 2001; (c)M. Kawata and U. Nagashima, Chem. Phys. Lett. 340, 165, 2001; (d)M. Kawata, M. Mikami, and U. Nagashima, J. Chem. Phys. 115, 4457, 2001; (e)M. Kawata, M. Mikami, and U. Nagashima, J. Chem. Phys. 117, 3526, 2002.

[7] I. J. Schoenberg, *Cardinal Spline Interpolation* (Society for Industrial and Applied Mathematics, Philadelphia, 1973).

[8] W. H. Press, S. A. Teukolske, W. T. Vetterling, and B. P. Flannery, "Numerical Recipes in C: The Art of Scientific Computing", Cambridge University Press, 1993.

[9] M. Frigo and S. G. Johnson, ICASSP conference proceedings, 3, 1381, 1998.