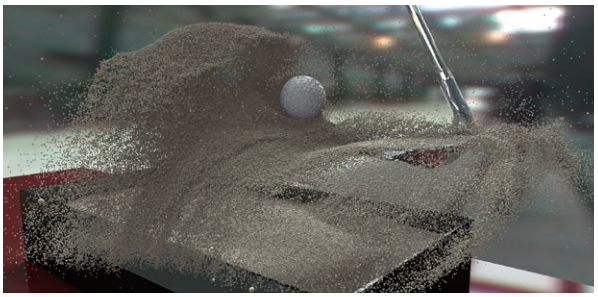
e-Science Journal
Global Scientific Information and Computing Center

Vol. 13

TSUBAME ESJ.



Multi-GPU Computation of Multi-phase Field Simulation of the Evolution of Metallic Polycrystalline Microstructure

A Large-scale Parallel Computation for Vibrational State Analysis Based on Quantum Monte Carlo method

Large-scale DEM Simulations for Granular Dynamics

http://www.gsic.titech.ac.jp/TSUBAME_ESJ

Multi-GPU Computation of Multi-phase Field Simulation of the Evolution of Metallic Polycrystalline Microstructure

Akinori Yamanaka* Masashi Okamoto Takashi Shimokawabe*** Takayuki Aoki*****

* Division of Advanced Mechanical Systems Engineering, Institute of Engineering, Tokyo University of Agriculture and Technology
** Department of Mechanical Systems Engineering, Graduate School of Engineering, Tokyo University of griculture and Technology
*** Global Scientific Information and Computing Center, Tokyo Institute of Technology

The multi-phase field method is recognized as one of the most promising simulation tools for predicting the evolution of polycrystalline microstructures in metals. We have recently developed a massively parallel computation technique that uses multiple graphics processing units (GPUs) for large-scale three-dimensional (3D) multi-phase field simulation. In this article, we introduce this multiple GPU computation technique, including an overlapping method that enables us to simultaneously perform computations concerning the GPU and data communication. We implemented our computation technique on the TSUBAME2.5 GPU supercomputer and evaluated its performance. Large-scale 3D simulations of polycrystalline grain growth performed on the TSUBAME2.5 exhibited high computing performance.

Introduction 1

In recent years, fuel-efficient vehicles such as hybrid cars have been actively researched and developed to reduce the environmental load of vehicular transportation. In order to further improve the fuel efficiency of vehicles, it is necessary not only to develop high performance engines and motors, but also to reduce the weight of vehicles by reducing the thickness of the sheet metal used in them. Meanwhile, sheet metal needs to be strengthened further to ensure the safety of vehicles' occupants in case of an accident. However, the strengthening of sheet metal often causes low formability: when sheet metal is strengthened, cracks and fractures occur in the press forming process. A large number of trial-and-error experiments have been carried out in the past to control the strength and formability of the sheet metal.

At present, because of intensifying international competition in the field of new materials, the infrastructure of computational materials engineering has grown increasingly important. Simulation-based material development is expected to shorten the development period and help efficiently produce new materials. In particular, the multi-phase field (MPF) method[1] has attracted considerable attention as one of the most promising simulation tools to predict microstructural evolution in metallic materials. In contrast to the phase field simulation of dendritic solidification using the TSUBAME2.0 supercomputer of the Tokyo Institute of Technology, which was awarded the ACM Gordon Bell Prize in 2011[2], an MPF simulation needs to solve multiple non-linear partial differential equations. Therefore, it requires more memory and a longer computational time than needed for conventional phase-field simulations.

In this study, we develop and test a multiple GPU computation technique for massively parallel computation of large-scale MPF simulations. This technique includes an overlapping method[3] that enables us to simultaneously carry out computations concerning the GPU and data communication. By implementing our computation technique to the TSUBAME2.5 supercomputer, we conducted large-scale three-dimensional (3D) MPF simulations of polycrystalline grain growth. We present the results of the performance evaluation of the large-scale MPF simulations thus obtained[4].

Multi-phase field method 2

The MPF method used in this study was proposed by Steinbach and Pezzola in 1999[1]. The MPF method simulates microstructural evolution in materials by assuming that the total free energy of the material monotonically decreases with time. When we consider a system of N crystal grains, the total free energy of the system is represented by the following equation:

$$G = \int_V \left[\sum_{i=1}^{N} \sum_{j=i+1}^{N} \left\{ W_{ij} \phi_i(\mathbf{r},t) \phi_j(\mathbf{r},t) \right\} + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left\{ -\frac{a_{ij}^2}{2} \left| \nabla \phi_i(\mathbf{r},t) \right|^2 \right\} \right] dV \quad (1)$$

where the first term of the right-hand side describes the potential energy and the second term corresponds to the gradient energy. The parameters W_{ij} and a_{ij} are functions of the interfacial energy and the interfacial thickness, and $\phi_i(\mathrm{r}, t)$ is a continuous order parameter called the phase field variable. $\phi_i(\mathrm{r}, t)$ describes the local volume fraction of the ith crystal grain at coordinate r and time t. Therefore, $\phi_i(\mathrm{r}, t)$ admits the value 1 for the ith grain and 0 otherwise. $\phi_i(\mathrm{r}, t)$ changes smoothly from 0 to 1 in the interfacial region.

The time evolution equation of the phase field variables (Allen–Cahn equation) can be derived by considering the monotonic reduction of the total free energy expressed by Equation (1):

$$\frac{\partial \phi_i}{\partial t} = -\frac{2}{n}\sum_{j=1,j\neq i}^{n} M_{ij}^{\phi}\sum_{k=1}^{n}\left\{\left(W_{ik}-W_{jk}\right)\phi_k + \frac{1}{2}\left(a_{ik}^2 - a_{jk}^2\right)\nabla^2\phi_k\right\} \quad (2)$$

where n is the number of the phase field variables greater than 0, and M_{ij}^{ϕ} is the mobility of the phase field variable.

In the MPF simulation, Equation (2) is solved by the second-order finite difference method for space and the first-order forward Euler method for time on a regular 3D computational grid. The program code was written in CUDA Fortran.

3 Multiple GPU computation of MPF method

3.1 Active parameter tracking method

Using the MPF method, we solve Equation (2) with respect not to N phase field variables but only n non-zero phase field variables. Therefore, we do not need to save the values of all N phase field variables. In this study, we employed the active parameter tracking (APT) method[5] to reduce memory consumption. The APT method is an essential algorithm for the efficient implementation of large-scale 3D MPF simulations. The details of the APT method can be found in reference[5]. In this article, we categorize the APT algorithm into APT1 and APT2. APT1 requires values of the phase field variables at each computational grid, whereas APT2 needs the values of variables at neighboring computational grids.

3.2 Domain decomposition

In order to perform the MPF simulation using multiple GPUs, we decomposed an entire computational domain into subdomains. A set of GPUs and a CPU is allocated for computation to each subdomain. Figure1 shows the 3D domain composition as an example of domain decomposition. When the entire computational domain divided by $NX \times NY \times NZ$ finite difference grids is decomposed into X, Y, and Z subdomains along X, Y, and z directions, the number of computational grids in each subdomain is $NX/X \times NY/Y \times NZ/Z$. The computation of the time evolution equation (Equation (2)) for each subdomain requires data on the surfaces of the neighboring subdomains. Therefore, we prepared boundary regions on the surfaces of each subdomain along x, y, and z directions. In the parallel computing, the data in the boundary region is transferred by using the Message Passing Interface (MPI) library. In the assessment of the performance of parallel computation, we attempted one-, two- and three-dimensional domain decompositions and selected the most-effective manner of these.

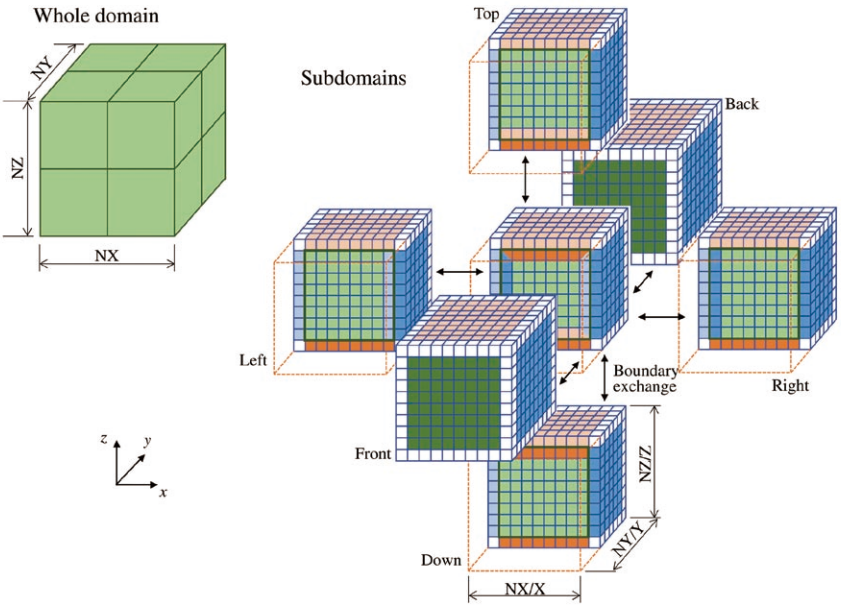


Fig. 1 3D domain decomposition.

3. 3 Assignment of threads and blocks

It is well-known that the assignment of CUDA threads and blocks is crucial to obtaining high performance from GPU computing. Figure 2 shows the assignment of CUDA threads and blocks to each subdomain employed in this study. We divided subdomains of size $nx \times ny \times nz$ into $X' \times Y'$ small domains along the x and y directions. Thus, the size of each small domain was $nx/X' \times ny/Y' \times nz$. A thread block of size $nx/X' \times ny/Y' \times 1$ handled each small domain by marching in the z direction[2]. The optimum number of CUDA threads and block were determined by trial and error because performance depends on the size of the computational domain and the number of GPUs used in the simulation.

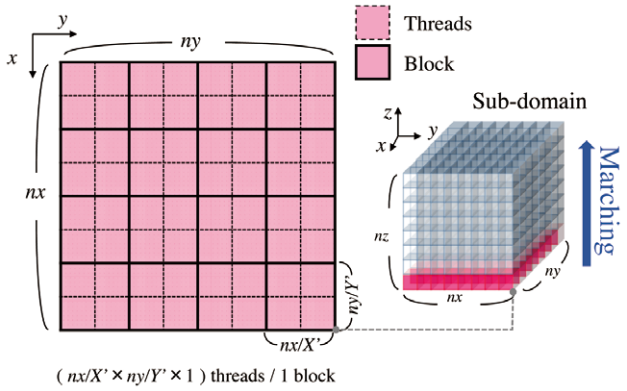


Fig. 2 Assignment of CUDA threads and blocks.

3. 4 Overlapping method

In order to perform parallel computation using multiple GPUs and CPUs, data communication is needed not only among CPUs, but also among the GPUs and between GPUs and CPUs. Therefore, the time required for data communication can degrade parallel efficiency. Thus, we propose an overlapping method that masks the time needed for data communication with the time required to perform computation on the GPUs[4].

Figure 3 shows the computational diagram used for the MPF simulation using the overlapping method. All operations (kernel execution and data communication) on the GPU are concurrently run in four CUDA streams (Stream 1~4). In Stream 1, we compute the time evolution equation for the internal region of the subdomain and handle the procedure for APT1. The computation of the time evolution equation for the boundary regions is carried out in Streams 2, 3, and 4 simultaneously. Following the completion of the computations in Streams 2, 3, and 4, data in the boundary regions is asynchronously transferred from the global memory on the GPU to the host (CPU) memory using the cudaMemCpyAsync function in the CUDA application programming interfaces (APIs).

The CPU transfers data in the boundary regions, calculated by the GPU in the previous time step, to the host memory. Following data communication between the GPU and the CPU in Streams 2, 3, and 4, the CPU initiates the procedures for APT1 and APT2 for the boundary regions and carries out data communication among CPUs with the MPI library. Data updates by APT1 and APT2 are transferred back from the CPU to the GPU while Stream 1 is executed on the GPU.

Once all streams on the GPU are synchronized with the computation and data communication on the CPU, the GPU handles the procedure for APT2 for the boundary regions and updates the calculated data. The calculated data on the CPU can be updated while all streams are executed on the GPU.

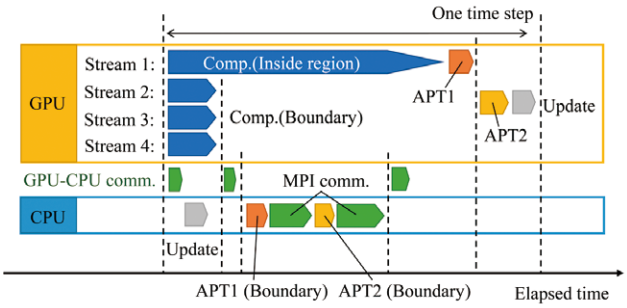


Fig. 3 The overlapping method.

4 Performance evaluation of multiple GPU computing on TSUBAME2.5

4. 1 Polycrystalline grain growth simulation

We evaluated the performance of our proposed multiple GPU computation by simulating large-scale 3D polycrystalline grain growth on the TSUBAME2.5 GPU supercomputer at the Global Scientific Information and Computing Center of the Tokyo Institute of Technology. All simulations were performed using single-precision floating-point calculation.

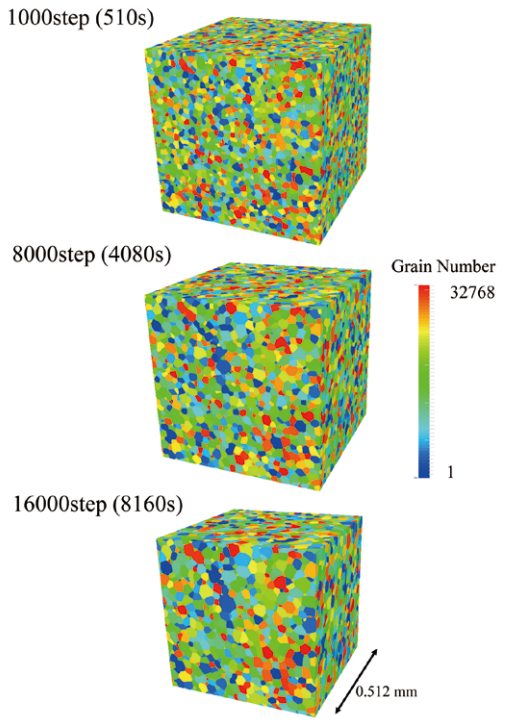


Fig. 4 Polycrystalline grain growth behavior simulated using the TSUBAME2.5 supercomputer. The simulation was performed using 256 GPUs on 10243 finite difference grids. Crystal grains were visualized by different colors.

As an example of the simulation results, we show snapshots of the polycrystalline grain growth behavior simulated using 256 GPUs on TSUBAME2.5 in Figure 4. In this simulation, the size of the computational domain was 0.512^3 mm^3 and the number of computational grids was 1024^3. The initial number of crystal grains was 32,768. The crystal grains were visualized by different colors. Polycrystalline grain growth can be observed in the heat treatment of metallic materials, e.g., annealing. The simulation result successfully reproduced the coarsening and the shrinking of crystal grains. The statistical evaluation of the polycrystalline microstructure, e.g., the distribution of crystal grain size and average grain size, is only possible when a large-scale MPF simulation taking into account a large number of crystal grains is performed.

In order to evaluate the performance of multiple GPU computing, we measured variation in the value of Floating-point Operations Per Second (FLOPS) by changing the number of GPUs used for the simulation. Figure 5 shows the results of the performance evaluation for weak scaling. Here, we evaluated the variation in the value of FLOPS with the number of GPUs for a fixed computational grid size per GPU. In this case, each GPU handled 256^3 computational grids and 512 crystal grains. The results show that the performance improved in proportion to the number of GPUs, and that the overlapping method improved performance. As a result, we attained 1.9 TFLOPS using 729 GPUs for 2304^3 computational grids and 373,248 crystal grains.

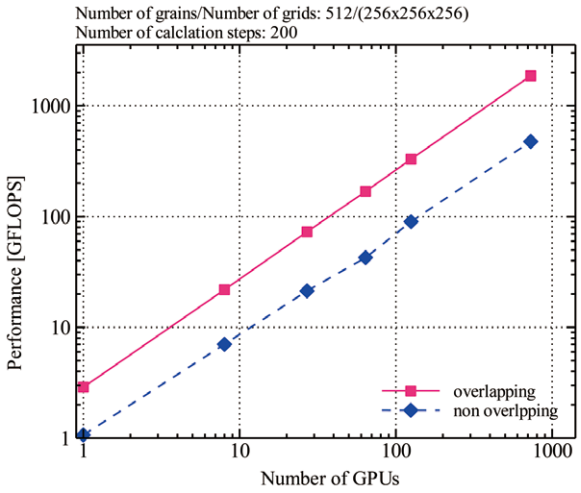


Fig. 5 Weak scaling of multiple GPU computing.

Furthermore, we evaluated the performance of our system for strong scaling. Here, the variation in the value of FLOPS with the number of GPUs for a fixed computational domain size was measured. The number of crystal grains was set to 512 per 256^3 computational grids. Figure 6 shows the results of the strong scaling for three different computational domain sizes: 256^3, 512^3, and 1024^3. Similar to the results for weak scaling shown in Figure 4, the performance improved with increasing

number of GPUs. However, in case a small computational domain size was used, the performance degraded when we used a large number of GPUs. This is because the size of subdomain handled by each GPU decreased by increasing the number of GPUs and the time for the data communication cannot be masked with that for computation. On the other hand, when we used large computational domain sizes and more than 100 GPUs, the performance clearly exhibited high scalability.

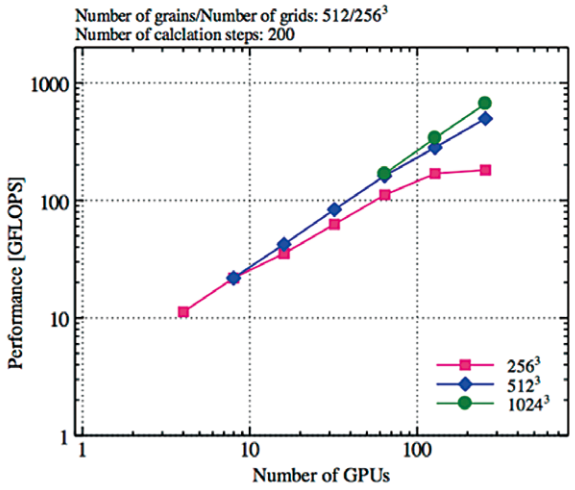


Fig. 6 Strong scaling of multiple GPU computing.

Summary 5

The multiple GPU computation technique has been developed for massively parallel computation of large-scale 3D MPF simulations. In this paper, we successfully applied this technique was successfully to the TSUBAME2.5 supercomputer. We showed that extremely large-scale MPF simulations of polycrystalline grain growth can be efficiently performed using our technique. The performance evaluation showed that high scalability was achieved by using the overlapping method.

It remains time-consuming to optimize the size and distribution of the polycrystalline microstructure that produces desirable mechanical properties in the material only by trial-and-error experiments. Therefore, we expect that efficient large-scale MPF simulations using the multiple GPU computing technique proposed in this article will contribute to the acceleration of materials development and the reduction of experimental cost.

Acknowledgements

This work was partially supported by the Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures in Japan. This work was also supported by a Grant-in-Aid for Scientific Research (KAKENHI 25630322) from the Japan Society for the Promotion of Science (JSPS).

References

[1] I. Steinbach, F. Pezzola: A Generalized field method for multiphase transformations using interface fields, Physica D, Vol. 45, pp. 385-393 (1999)

[2] T. Shimokawabe, T. Aoki, T. Takaki, A. Yamanaka, A. Nukada, T. Endo, N. Maruyama, S. Matsuoka: Peta-scale phase-field simulation for dendritic solidification on the TSUBAME2.0 supercomputer, Proceedings of the 2011 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, SC'11, IEEE Computer Society, Seattle, WA, USA, (2011)

[3] M. Okamoto, A. Yamanaka, T. Shimokawabe, T. Aoki: Multiple GPU Computing of Polycrystalline Grain Growth Simulation using Multi-phase-Field Method, Transaction of the Japan Society for Computational Engineering and Science, Vol. 2013, p. 20130018 (2013) (in Japanese)

[4] A. Yamanaka, M. Okamoto, T. Shimokawabe, T. Aoki: Large-scale 3D multi-phase field simulation of microstructure evolution using TSUBAME2.5 GPU supercomputer, Proceedings of 2nd International Congress on 3D Materials Science, The Minerals, Metals & Materials Society, pp. 59-64, (2014)

[5] S. G. Kim, D. I. Kim, W. T. Kim, Y. B. Park: Computer simulation of two-dimensional and three-dimensional ideal grain growth, Phys. Rev. E, Vol. 74, p. 061605 (2006)

A Large-scale Parallel Computation for Vibrational State Analysis Based on Quantum Monte Carlo method

Ryota Nakayama* Osamu Fujioka** Yukiumi Kita** Masanori Tachikawa**

* International College of Arts and Sciences, Yokohama City University
** Quantum Chemistry Division, Yokohama City University

We show theoretical outline of vibrational quantum Monte Carlo (vibQMC) method that we have recently developed, and its parallel implementation method toward a large-scale parallel computation on a super-computer system. The parallelization ratio of our vibQMC program code is 99.9981% , and the practical parallel efficiency using 5376 cores on TSUBAME 2.5 super-computer system is about 91% .
We also show theoretical results of vibrational state analysis of the monohydrated negative core ion, $H_3O_2^-$, which is a precursor ion in forming aerosols in the atmosphere.

1 Introduction

Quantum Monte Carlo (QMC) method is one of the most accurate *ab initio* methods to solve the Schrödinger equations of atoms, molecules, and solids [1]. In particular, diffusion Monte Carlo (DMC) method gives a very accurate variational energy of the systems. However, the scope of application of the DMC method is practically limited in the estimation of a variational energy due to some serious problem known as the population control bias, the difficulty to compute the expectation values of the non-commutative operators with Hamiltonian, *etc*.

Reptation Monte Carlo (RMC) method [2] is one of QMC methods using the imaginary-time propagation of time-dependent Schrödinger equation as well as the DMC method. In principle, RMC method has the equivalent theoretical accuracy to the DMC method, but allows the population control bias free algorithm and the exact estimation of non-commutative operators with Hamiltonian. The computational cost in RMC calculations is, however, extremely expensive compared to that in the DMC calculations. Thus, the scope of the application of the RMC method is limited in small molecular systems such as H_2O molecule [3].

In this report, we show the parallel implementation method of our RMC program developed toward RMC calculations of large-scale molecular systems. The RMC program has been developed as a part of vibrational quantum Monte Carlo (vibQMC) method that we have recently proposed. In the following sections, we describe the theoretical outline of our vibQMC method, the benchmark calculations of parallel efficiency of vibQMC program on TSUBAME 2.5 super-computer system, and theoretical results of vibrational state analysis of monohydrated negative core ion, $H_3O_2^-$, which is a precursor ion in forming aerosols in the atmosphere, as an application of a large-scale parallel RMC computation.

2 Method

In this study, we used two types of QMC technique, variational Monte Carlo (VMC) and RMC methods, for anharmonic vibrational state analyses of polyatomic molecules. We here briefly describe theoretical outline of the both methods.

2.1. Variational Monte Carlo (VMC) method

We consider the following expectation value E of Hamiltonian operator $\hat{H} = \hat{T} + \hat{V}$ ($\hat{T}$ and $\hat{V}$ are kinetic and potential energy operators, respectively) with a given trial wave function Ψ_T:

$$E = \frac{\int \Psi_T(\boldsymbol{Q}) \hat{H} \Psi_T(\boldsymbol{Q}) d\boldsymbol{Q}}{\int \Psi_T^2(\boldsymbol{Q}) d\boldsymbol{Q}} = \frac{\int \Psi_T^2(\boldsymbol{Q}) \varepsilon_L(\boldsymbol{Q}) d\boldsymbol{Q}}{\int \Psi_T^2(\boldsymbol{Q}) d\boldsymbol{Q}} \quad (1)$$

where $\boldsymbol{Q} \equiv \{q_1, q_2, ..., q_N\}$ is a generalized multi-dimension coordinate, and is referred to as a configuration or *walker* in QMC methods. The variable $\varepsilon_L \equiv \Psi_T^{-1} \hat{H} \Psi_T$ is a local energy. We assume a real trial wave function ($\Psi_T^* = \Psi_T$). In Eq. (1), the expectation value E can be evaluated as the average of ε_L over whole configuration space $\boldsymbol{Q}$ with the statistical weight Ψ_T^2. In VMC method, thus, we simply calculate the value of E by generating a set of $\{\boldsymbol{Q}_i\}(i=1 \sim M)$ with Metropolis's method [4], where M is the number of sampling points.

The VMC method enable us to analyze not only a total energy of a system, but also a expectation value of a given physical properties, although its theoretical accuracy strongly depends on the quality of trial wave function. In QMC calculations for many electron system such as atoms, molecules, and solids, Slater-Jastrow type trial wave function, which consists of a single (or multi) Slater determinant(s) obtained with molecular orbital

calculations and Jastrow factor involving many body effects, are generally employed. In the present study in which we focus on quantum many body problems of molecular vibrations, we employed the following Vibrational Self- Consistent Field (VSCF) type wave function [5]:

$$\Psi_{\mathrm{VSCF}} = \varphi_1(q_1)\times\varphi_2(q_2)\times\cdots\times\varphi_N(q_N) \tag{2}$$

where q_i and φ_i are a vibrational coordinate and modal function of ith normal vibrational mode, respectively. Each modal function is expanded by a set of eigenfunctions of harmonic oscillator. Variational parameters in the trial wave function such as centers, exponents, and expansion coefficients of basis functions are optimized with the linear optimization method [6] proposed by Umrigar *et al*. Optimized VSCF functions are used in the initial trial wave function in RMC calculations described in the next section.

2.2. Reptation Monte Carlo (RMC) method

Here, we consider a one-dimension case for the simplification of mathematical expression. In QMC method using an imaginary-time evolution of time-dependent Schrödinger equation, the exact wave function Ψ_0 are obtained by acting an imaginary-time propagator on a given initial trial wave function Ψ_{T} :

$$\left|\Psi_0\right\rangle = \lim_{\tau\to\infty} \mathrm{e}^{-\frac{\tau}{2}\hat{H}}\left|\Psi_{\mathrm{T}}\right\rangle \quad \cdots\times\varphi_N(q_N) \tag{3}$$

where τ is an imaginary-time. In RMC method, we consider the following pseudo-partition function Z_0 :

$$Z_0 \equiv \left\langle\Psi_0\middle|\Psi_0\right\rangle = \lim_{\tau\to\infty}\left\langle\Psi_{\mathrm{T}}\middle|\mathrm{e}^{-\tau\hat{H}}\middle|\Psi_{\mathrm{T}}\right\rangle \tag{4}$$

Decomposing the imaginary-time propagator $\mathrm{e}^{-\tau\hat{H}}$ into N short time propagators with an imaginary-time step $\Delta\tau = \tau / N$, we obtain the following partition function under the second order approximation (Suzuki-Torotter decomposition[7]):

$$\begin{aligned} Z_0 &= \int\cdots\int dq^{(0)}\cdots dq^{(N)} \\ &\quad P(q^{(0)},\cdots,q^{(N)})\times W(q^{(0)},\cdots,q^{(N)}) \end{aligned} \tag{5}$$

where

$$P \equiv \Psi_{\mathrm{T}}(q^{(0)})\left[\prod_{i=0}^{N-1} g(q^{(i)},q^{(i+1)};\Delta\tau)\right]\Psi_{\mathrm{T}}(q^{(N)}) \tag{6}$$

$$W \equiv \prod_{j=0}^{N-1} \mathrm{e}^{-\frac{\Delta\tau}{2}\left\{\varepsilon(q^{(j)})+\varepsilon(q^{(j+1)})\right\}} \tag{7}$$

The variable $q^{(i)}$ is the configuration at the imaginary-time $\Delta\tau\times i$, and a set of configurations $\{q^{(i)}\}$ $(i=0\sim N)$ is referred to as an imaginary-time path or *reptile*. In Eqs.(5)-(7), we employed the following second order decomposition:

$$\left\langle q\middle|\mathrm{e}^{-\tau\hat{H}}\middle|q'\right\rangle \cong \mathrm{e}^{-\frac{\Delta\tau}{2}\varepsilon_{\mathrm{L}}(q)}\left\langle q\middle|\mathrm{e}^{-\Delta\tau\mathcal{H}}\middle|q'\right\rangle \mathrm{e}^{-\frac{\Delta\tau}{2}\varepsilon_{\mathrm{L}}(q')} \tag{8}$$

where $\hat{H} = \mathcal{H} + \varepsilon_{\mathrm{L}}(q)$, $\mathcal{H} \equiv \hat{T} - \Psi_{\mathrm{T}}^{-1}(q)\hat{T}\Psi_{\mathrm{T}}(q)$, and $g(q_i,q_{i+1};\Delta\tau) \equiv \left\langle q_i\middle|\mathrm{e}^{-\Delta\tau\mathcal{H}}\middle|q_{i+1}\right\rangle$.

In Eq.(5), $P(q^{(0)},\cdots,q^{(N)})$ is the propagator to generate reptiles whose distributions are according to Ψ_{T}^2, and $W(q^{(0)},\cdots,q^{(N)})$ is the statistical weight of the generated reptile. In our RMC program, reptiles are generated with the Langevin equation defined as

$$q^{(i+1)} = q^{(i)} + 2Dv(q^{(i)})\Delta\tau + \chi(2D\Delta\tau) \tag{9}$$

where $D=1/2\mu$, μ is the reduced mass of a vibrational mode, $\chi(2D\Delta\tau)$ the Gaussian distribution with a variance $2D\Delta\tau$, and $v \equiv \Psi_{\mathrm{T}}^{-1}\nabla\Psi_{\mathrm{T}}$ drift velocity. In reject/accept procedures in RMC calculations, we employed Metropolis's method.

After a lot of Monte Carlo samplings with a long time imaginary-time τ, the distributions of $q^{(0)}$ and $q^{(N)}$ converge to the distribution $\Psi_{\mathrm{T}}\times\Psi_0$, and that of $q^{(N/2)}$ converge to the exact density distribution Ψ_0^2. To evaluate the expectation value of Hamiltonian operator, we used the following estimator (mixed estimator) with the distribution $\Psi_{\mathrm{T}}\times\Psi_0$:

$$\begin{aligned} \left\langle\hat{H}\right\rangle_{\mathrm{mix}} &\equiv \lim_{\tau\to\infty}\frac{\left\langle\Psi_{\mathrm{T}}\middle|\hat{H}\mathrm{e}^{-\tau\hat{H}}\middle|\Psi_{\mathrm{T}}\right\rangle}{\left\langle\Psi_{\mathrm{T}}\middle|\mathrm{e}^{-\tau\hat{H}}\middle|\Psi_{\mathrm{T}}\right\rangle} \\ &= \frac{\int\Psi_{\mathrm{T}}\Psi_0\varepsilon_{\mathrm{L}}(q)dq}{\int\Psi_{\mathrm{T}}\Psi_0\,dq} \end{aligned} \tag{10}$$

The exact density distribution Ψ_0^2 is used to calculate the expectation value of non-commutative operators with Hamiltonian such as potential energy operator, and to analyze the geometry of molecules.

Parallelization of QMC algorithm 3

As described in the previous sections, we generate multiple Markov chains with Metropolis's method for walkers and reptiles in VMC and RMC calculations, respectively. Thus, the most straightforward and useful approach toward a parallel computing of QMC calculations is simultaneous samplings in configuration space using statistical independent multiple walkers/ reptiles (see Fig.1). In our program, multiple Markov chain generations are simply performed with multiple processes. Figure 2 show the schematic illustration of our parallelized QMC program code. Data I/O from/to a external storage are performed in Master process. Markov chain generations for walkers/reptiles are performed on both Master and Slave processes. We implement such parallelized algorithm with MPI (Message Passing Interface), and use OpenMPI library version 1.4.2 on TSUBAME 2.5 super computer system.

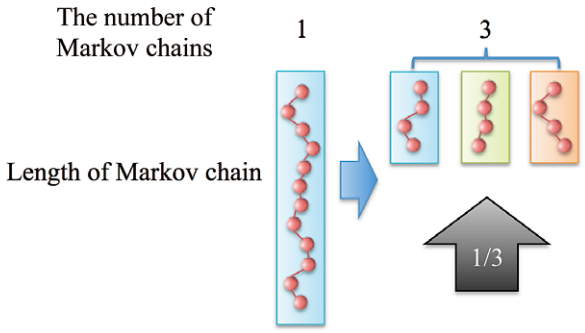


Fig. 1 Schematic diagram of decomposed Markov chain.

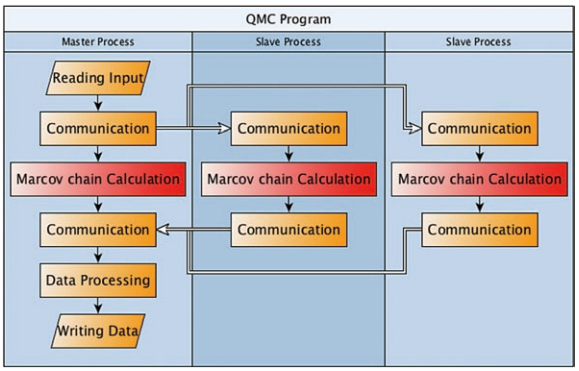


Fig. 2 Schematic illustration of parallelized QMC program code.

Results and discussion 4

4.1. Parallel efficiency

We performed a benchmark calculation of the parallel efficiency of our vibQMC program on TSUBAME 2.5 super-computer system in Tokyo Institute of Technology: the number of cores is up to 5376 cores (1 process/core × 12 cores/node × 448 nodes). The vibrational ground state of a monohydrated positive core ion, $H_5O_2^+$, are used in benchmark calculations. We used the analytical potential function proposed by Huang *et al.*[8] which well reproduces potential energy surface at CCSD(T)/aug-ccpVTZ level of *ab initio* calculations.

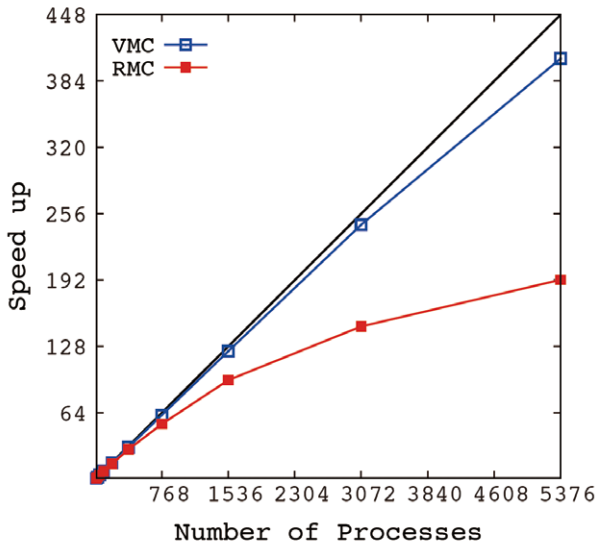


Fig. 3 Relative speed up to the computational time with 12 parallels (1 node) in VMC and RMC calculations on TSUBAME 2.5 super-computer system.

Figure 3 shows the relative speed up (=Time[12 process] /Time [N processes]) to the computational time with 12 parallels (1 node) in VMC and RMC calculations. The speed up with 5376 cores is 4868 (=405.70 × 12) in VMC calculations, and 2307 (=192.25 × 12) in RMC calculations: the parallel efficiencies in VMC and RMC calculations are about 91% and 43% , respectively. Assuming that the size of problems is constant in each calculation (Amdahl's law[9]), we can estimate the ratio of sequential execution process in 12 parallels calculation as $r_{\mathrm{VMC}}^{(12)}$ =0.023% and $r_{\mathrm{RMC}}^{(12)}$ =0.3% in VMC and RMC calculations, respectively.

These ratio denote that 99.9981% and 99.975% of total processing in a non-parallel (1 core) calculation are parallelized in VMC and RMC calculations, respectively.

4.2. Monohydorated negative ion core: $H_3O_2^-$

We performed anharmonic vibrational state analyses of a monohydrated negative core ion, $H_3O_2^-$, which is one of precursor ions in forming aerosols in the atmosphere, as an application of large-scale parallel computation of our QMC program. With a conventional *ab inito* calculation which does not include the nuclear quantum effect (NQE) or the effect of quantum molecular vibrations, the most stable equilibrium structure of $H_3O_2^-$ system becomes asymmetric O—H*···O or O···H*—O structures as shown in the Fig. 4 (a), because the central hydrogen atom (H*) in $H_3O_2^-$ has the double-well potential along the H* transfer coordinate. On the other hand, with theoretical calculations including the NQE[10,11], the effective potential energy curve along the H* transfer coordinate changes to the single-well due to the small potential barrier height (0.88 kJ/mol). Then, the H* is located at the center between two oxygen atoms as O···H*···O as shown in Fig. 4(b) which corresponds to the transition state (TS) in conventional *ab initio* calculations.

We analyzed the vibrational ground state and fundamental tone state of vibrational mode associated with the H* transfer between two oxygen atoms (here we call bridge-vibration) for $H_3O_2^-$ and its deuterium (D) and tritium (T) species. The VSCF type wave function was used as the trial wave function in VMC and RMC calculations. The analytical potential function proposed by Huang *et al.*[8] which well reproduces potential energy surface at CCSD(T)/aug-ccpVTZ level of *ab initio* calculation are used.

Table 1 shows the zero-point vibrational energy (ZPE) and the fundamental frequency of the bridge-vibration for all species. The ZPEs, which are the variational energy, obtained with RMC method are lower than that with VMC method for all species. The theoretical accuracy is, thus, improved with the RMC calculations compared to the VMC calculations. In addition, the RMC calculation well reproduce the experimental fundamental frequency (697cm^{-1}) of the bridge- vibration of H-species within the error of 12 cm^{-1}.

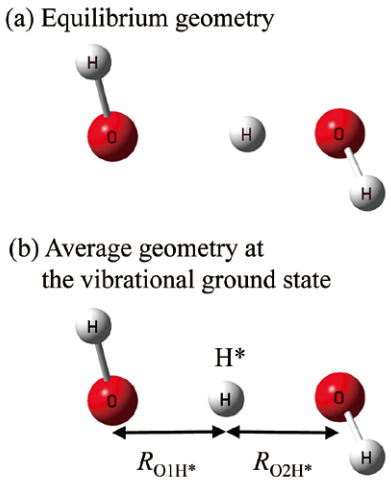


Fig. 4 Schematic illustrations of (a) equilibrium and (b) vibrational averaged geometries of $H_3O_2^-$.

Method	Species	ZPE	ω
VMC	H	6881.6(1)	763.9(1)
	D	5046.4(0)	488.4(1)
	T	4261.7(0)	383.7(1)
RMC	H	6702.2(2)	685.3(3)
	D	4923.8(1)	420.7(2)
	T	4160.1(1)	324.7(2)
Expl.[12]	H		697

Table 1. Zero-point vibrationalal energy (ZPE) and fundamental frequency (ω) of the bridge vibration mode of $H_3O_2^-$ system. Unit in cm^{-1}

In order to analyze structural properties of H*, D*, and T* at each vibrational state, we focus on the parameter $\delta_{OH*} = R_{O1H*} - R_{O2H*}$, where the two oxygen-hydrogen distances, R_{O1H*}, $_{O2H*}$, are defined in Fig. 4(b). The H* (or D*, T*) is located at the central position between two oxygen atoms at $\delta_{OH*} = 0$. Figure 5 shows one dimensional distribution of δ_{OH*} obtained with RMC calculations. In the vibrational ground states (Fig. 5(a)), the H-species has a single peak in the distribution at $\delta_{OH*} = 0$. Similar results have been reported with DMC calculations by McCoy *et al.*[10] and path-integral molecular dynamics (PIMD) calculation by Suzuki *et al.*[11] In their PIMD calculations, D- and T-species have almost the same distributions of δ_{OD*} and δ_{OT*} each other at 50 K. Our RMC calculation, however, shows that the

distribution of T-species is more localized than that of D-species at 0 K (vibrational ground state).

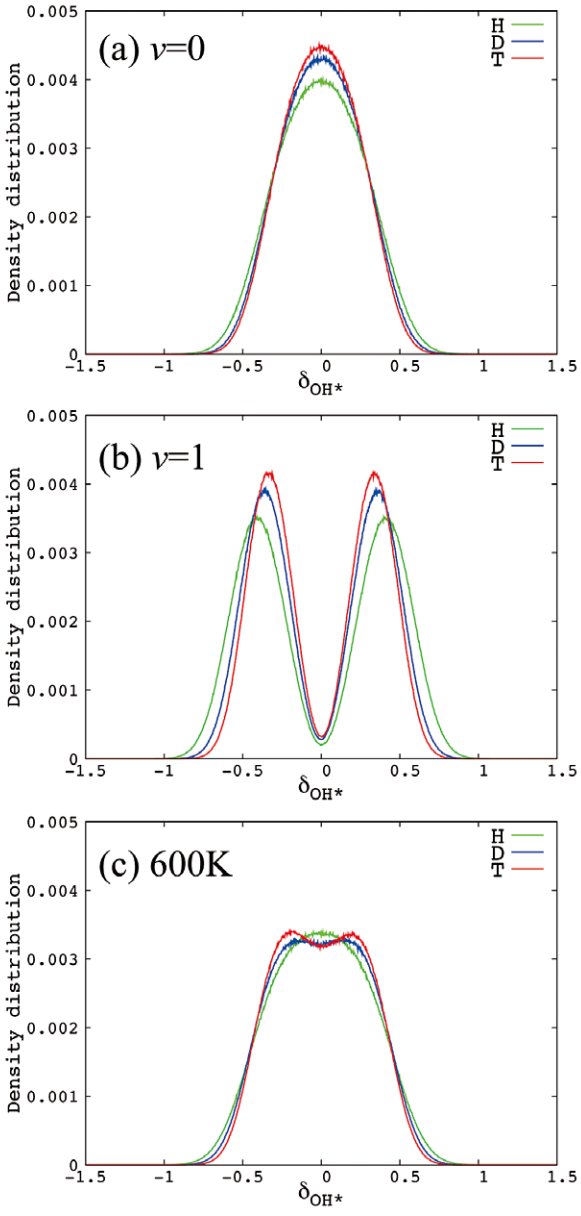


Fig. 5 One dimensional distribution of δ_{OH^*} in $H_3O_2^-$ and its D- and T-species. (a) the vibrational ground state (v=0), (b) the fundamental tone state of the bridge vibrational mode (v=1), (c) at 600K. Unit in Bohr.

The distribution of δ_{OH^*} at the fundamental tone state of the bridge-vibration is shown in Fig. 5(b). All species have double peaks in the distribution at around two equivalent equilibrium geometries. Each peak position of H-, D-, and T-species in one side of the distribution is shifted to a large $|\delta_{OH^*}|$ region as the nuclear mass becomes lighter due to the anharmonicity of potential energy curve along the H* (or D*, T*) transfer coordinate. It is interesting that the probability density at $\delta_{OH^*} = 0$ (TS with respect to H* transfer) increases as the nuclear mass becomes heavier at both the vibrational ground and fundamental tone states.

Figure 5(c) shows the distributions of δ_{OH^*} at 600K, where we assumed Boltzmann statistics with two vibrational states. The H-species has a single peak in distribution, but its distribution at 600 K is more delocalized than that at 0 K. On the other hand, the distributions of the D- and T-species slightly split as reported in the previous PIMD calculations at 600 K[11]. The two state model with RMC calculations well reproduces PIMD results at 600 K. The consistency between both methods indicates that the splits of the distribution of D- and T species at high temperature region are mainly due the vibrational excitation of the bridge-vibration.

Conclusion 5

In this report, we describe theoretical outline of vibrational quantum Monte Carlo (vibQMC) method, variational Monte Carlo (VMC) and reptation Monte Carlo (RMC) methods, and its parallel implementation method toward a large-scale parallel computation on a super-computer system. The parallelization ratios of our vibQMC program code are 99.9981 % and 99.975 % in VMC and RMC programs, respectively. The parallel efficiencies using 5376 cores on TSUBAME 2.5 super-computer system is about 91% and 43 % in VMC and RMC calculations, respectively. We also show theoretical results of vibrational state analysis of the monohydrated negative core ion, $H_3O_2^-$, which is a precursor ion in forming aerosols in the atmosphere as an application of a large-scale parallel computation of our vibQMC program. We used less time-consuming analytical potential functions in vibrational state analyses from a computational cost point of view. A further improvement of parallel efficiency is strongly expected by a combination of vibQMC and on-the-fly *ab initio* method in potential energy calculations. Such approach must be quite useful for analyzing a larger molecular system, because an accurate analytical potential function is no longer available for such systems.

Acknowledgements

The present calculations were carried out on the TSUBAME 2.5 supercomputer in the Tokyo Institute of Technology as a subject of the TSUBAME grand challenge (category B). Financial support was provided by Grant-in-Aid for Scientific Research and for the priority area by Ministry of Education, Culture, Sports, Science and Technology, Japan, for Y.K. and M.T.

References

[1] B.L. Hammond, W.A. Lester Jr. and P.J. Reynolds, "Monte Carlo Methods in Ab Initio Quantum Chemistry" (World Scientific, 1994).
[2] S. Baroni and S. Moroni, Phys. Rev. Lett., 82, 4745 (1999).
[3] D.G. Oblinsky, W.K. Yuen, S.M. Rothstein, J. Mol. Struct. (THEOCHEM) 961, 219 (2010).
[4] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, J. Chem. Phys., 21, 1087 (1953).
[5] J. M. Bowman, J. Chem. Phys. 68, 608 (1978).
[6] J.Toulouse, C.J. Umrigar, J. Chem. Phys. 126, 084102 (2007).
[7] M. Suzuki, Proc. Japan Acad., 69, 161 (1993).
[8] Huang et al. J. Am. Chem. Soc. 126, 5042 (2004) .
[9] Gene M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In AFIPS Conference Proceedings, pp. 483-485 (1967) .
[10] A. B. McCoy, X. Huang, S. Carter, and J. M. Bowman, J. Chem. Phys. 123, 064317 (2005)
[11] K.Suzuki, M. Shiga, and M. Tachikawa, J. Chem. Phys. 129, 144310 (2008).
[12] E.G. Diken, J.M. Headrick, J.R. Roscioli, J.C. Bopp, and M.A. Johnson, A.B. McCoy J. Phys. Chem. A, 109, 8 (2005)

Large-scale DEM Simulations for Granular Dynamics

Satori Tsuzuki Seiya Watanabe Takayuki Aoki
Global Scientific Information and Computing Center (GSIC) at Tokyo Institute of Technology

Granular materials such as sands and powders sometimes play important roles in science and engineering. In the numerical method DEM (discrete element method), the collisions between granular particles are described as classical spring force and friction force models. Particle simulations using the DEM have been commonly applied to study granular materials. Even in a spoon of sugar, there are more than 1 million grains. A large number of particles have to be used to simulate granular materials. Although coarse-graining models have been developed to reduce the particle number, it is quite meaningful to carry out DEM simulations with realistic-sized particles for quantitative granular analyses. Recent supercomputers perform well enough to carry out such granular simulations. Since granular material changes its particle distribution in time and space, it is necessary to introduce a dynamic load balance in spite of the large computation overhead. After developing several numerical methods for GPU implementation, we have succeeded in carrying out practical DEM applications using over 10 million particles. A golf bunker shot simulation demonstrates quite realistic results.

1 Introduction

Beside natural phenomena, granular materials often appear in machines as toners in laser printers, pharmaceutical tablets in formulation processes, suspension flows in chemical plants, and so on. Granular materials sometimes behave as a liquid and sometimes behave as a solid.

There is a lot of demand for studying granular materials by computer simulation. In the commonly used DEM (Discrete Element Method), spring and friction forces work only among particles that are in contact. Since the number of contact interactions with a particle is small, the cost of the memory access is greater than that of floating-point operation in interaction calculation. On the other hand, MD (molecular dynamics) or astrophysics N-Body problems are similar types of particle simulations; however, the cost of floating-point operation is quite dominant due to their long-range interactions.

There are few studies on large-scale granular simulations because the computational cost proportionally increases with increasing number of particles. When we want to simulate granular materials in a spoon of sugar, there are more than 1 million sugar grains, and the computational cost becomes quite large. Although coarse-graining models have been developed to reduce the particle number, it is very meaningful to carry out DEM simulations with realistic-sized particles for quantitative analyses. Recent supercomputers perform well enough to carry out granular simulations using the real particle number. Almost all supercomputers consist of multiple nodes, each node has a few processors with or without accelerators such as the GPU (graphics processing unit), and are interconnected by high-speed networks. Large-scale DEM simulations have to run efficiently on the memory-distributed system of supercomputers.

The computational cost of the DEM is proportional to the number of particles. We divide the particles into groups with the same number of particles. If we divide the particles by the numbers initially assigned to them, unacceptable data communication among nodes will occur. It is natural to apply spatial domain decomposition to the DEM simulation since the particles interact by being in contact with each other. Unfortunately, since granular material changes its particle distribution in time and space, we do not keep the same particle number in each subdomain with static domain decomposition. Therefore, it is necessary to introduce a dynamic load balance.

The GPU has the advantages of high performance for floating-point operation and wide memory bandwidth. We have to use the device memory on the GPU board, so the communication cost among the device memory becomes large. This makes large-scale DEM simulations more difficult on GPU supercomputers.

2 DEM computation on GPU

Recent GPUs have more than 2,000 processing cores (CUDA cores) on a single chip. Fine-grain parallelization and multi-thread programming are required to have high performance, and we have to take into consideration the hierarchical structure and the limited size of the memory. In the GPU implementation of our code, we use the CUDA programming framework given by NVIDIA.

In the DEM, the particle interaction is modeled as a spring and a dumping force proportional to the penetration depth and the relative velocity of the two particles in contact with each other in the normal direction. In the tangential direction, friction is also taken into account as shown in Fig. 1.

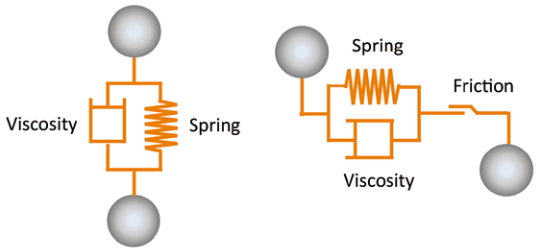


Fig. 1 DEM computational model.

The equation of translational motion is described in classical mechanics in Eq. (1), and the right-hand side is the contribution from all the particles in contact.

$$m\ddot{x}_i = \sum_{j \neq i}^{N} (-kx_{ij} - \gamma \dot{x}_{ij}) \qquad (1)$$

where x_i is the position of the i-th particle and x_{ij} is the penetration depth between the i-th and j-th particles. The notations k and γ are the spring constant and dumping coefficient, respectively. In rotational motion, similar equations for the angular velocities with torques are solved, and the particle positions and velocities are updated by time integration with the leap-frog scheme or the Runge-Kutta method.

All the dependent variables of particles are allocated to the device memory (so called "global memory" in CUDA programming). In the thread assignment for CUDA cores, one thread computes one particle motion by solving Eq. (1).

It is quite inefficient to make judgments on whether particles are in contact for all the particles. Neighbor-particle lists are commonly used to reduce the cost to find the particles in contact; however, the amount of memory needed to save the list in the cell often becomes a severe problem in large-scale simulations.

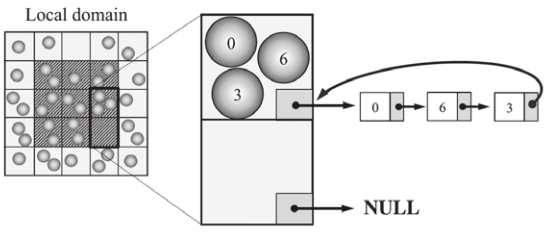


Fig. 2 Neighbor particle search using linked-list method on GPU.

The linked-list method is a candidate to reduce the memory use as shown in Fig. 2 [1][2]. Each particle has a memory pointer referring to the next particle in the same cell. Using chain access we can reduce the memory usage to 1/8.

Multi-GPU DEM simulation using Dynamic Load Balance 3

In large-scale DEM simulations requiring a lot of GPUs, the computational domain is decomposed into subdomains. A GPU is assigned to each subdomain and computes particles located in the subdomain. Since particle distributions change in time and space, static domain decomposition does not keep the same number of particles in each subdomain. The slice-grid method [3] is introduced to maintain equal numbers of particles to keep the memory usage equal and the computational load balance among GPUs. Figure 3 illustrates that the vertical boundaries of the subdomains move first to keep the vertical load balance of the horizontal subdomain group, and the horizontal boundaries of the subdomains move individually next.

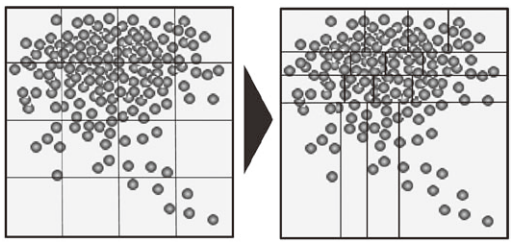


Fig. 3 Dynamic load balance based on two-dimensional slice-grid method.

To determine the moving distance of the previous subdomain boundary, we have to count the particles located near the boundary. We propose an efficient way to find near-boundary particles on "global memory" without copying the particle data to the host CPU memory. The subdomain is divided with a proper space Δl, as shown in Fig. 4, and we count the particle number within the Δl space by means of the Thrust library. The particles in the neighbor subdomain after moving the boundary are transferred there through the PCI-Express bus. When the neighbor subdomain is allocated to a different node, the data transfer includes the node-to-node communication by the MPI library.

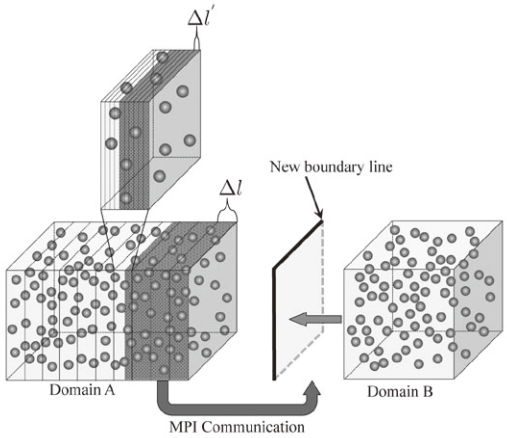


Fig. 4 Particle counting and boundary moving on GPU.

Frequent data transfer of particles among subdomains causes fragmentation of the GPU memory, which degrades the access performance and memory usage. In Fig. 5, a defragmentation should be executed with a proper frequency in spite of the overhead of data movement to the host memory.

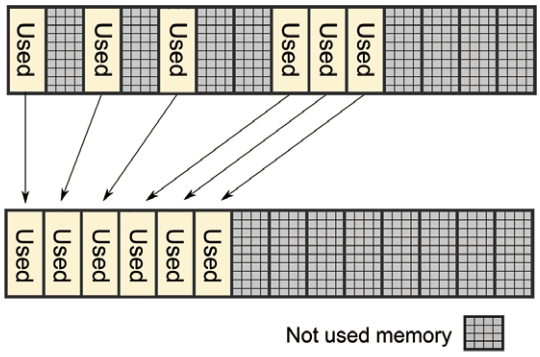


Fig. 5 Defragmentation of particle data on GPU device memory.

Strong and Weak scalabilities on TSUBAME 2.5

4

We examined the performance of our DEM simulation on the TSUBAME 2.5 supercomputer with GPUs (NVIDIA Tesla K20X) with the dynamic load balance based on the 2-dimensional slice-grid method. The scalabilities are studied for the agitation simulation shown in Fig. 6 in the range of 2 M (million) to 1.0 billion particles.

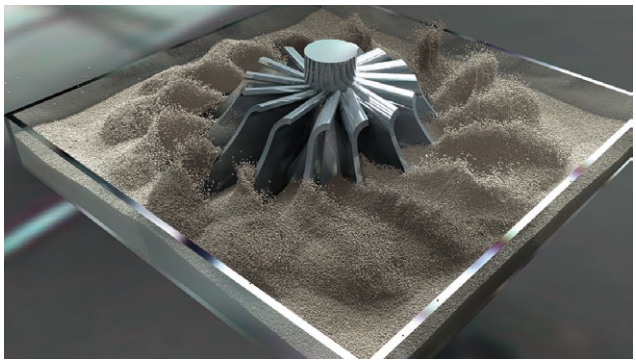


Fig. 6 Agitation simulation using 4,120,000 particles on 64 GPUs.

In Fig. 7, the vertical axis indicates the performance defined as the particle number divided by the elapsed time. The solid lines indicate strong scalability with 2M (2 million), 16M, and 129M particles, respectively. The square symbols show the results with 2M particles using 4 to 64 GPUs, the triangles denote 16M particles using 32 to 256 GPUs, and the cross marks are with 129M particles using 256 to 512 GPUs. According to Fig. 7, the performances keep improving in proportion to the number of GPUs with 8 to 16-fold and become sluggish with more than 16 GPUs when using 2M particles.

We study the weak scalability by comparing the performances for 2M, 16M, and 129M particles using 4 GPUs, 32 GPUs, and 256 GPUs, respectively. It was found that the weak scalability was degraded from the ideal dashed line with increasing GPU number. Some subdomains have shapes with high-aspect ratio and particles move across the boundary easily, so the amount of data communication increases and the total performance becomes worse. It is meaningful that we succeeded in a large-scale DEM simulation with 129M particles on 512 GPUs regardless of low parallel efficiency.

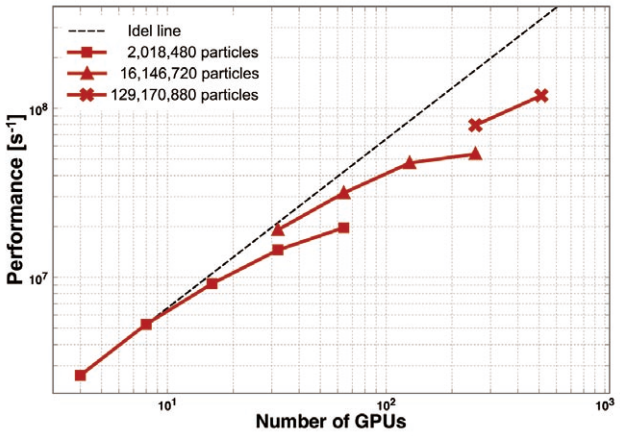


Fig. 7 Scalabilities of DEM simulation on TSUBAME 2.5.

Application to practical problems 5

To solve the problems when interacting with complex shapes of objects, the contact judgment is easily done by introducing the signed-distance function from the object surface. When we have the CAD data of the object, we have the distance from the object surface at the particle positions instead of having to calculate the minimum distance from all the polygons of the CAD data, as shown in Fig. 8[4].

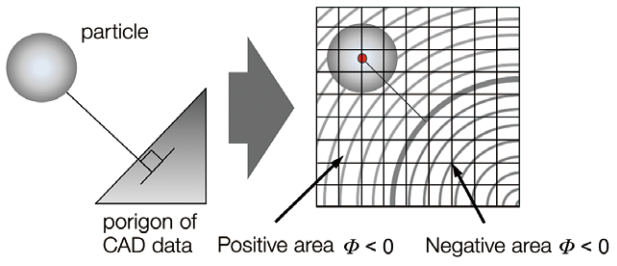


Fig. 8 Representation of object shape using Level Set method.

Figure 9 shows the simulation for a granular conveyor with 4,330,000 particles on 64 GPUs. The sand sliding down on the spiral structure is demonstrated with 4,160,000 particles on 32 GPUs, as shown in Fig. 10.

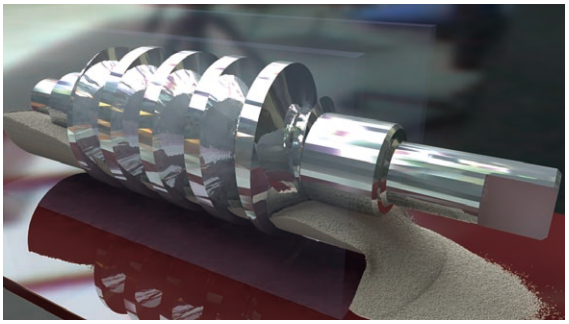


Fig.9 Conveyor simulation using 4,330,000 particles on 64 GPUs.

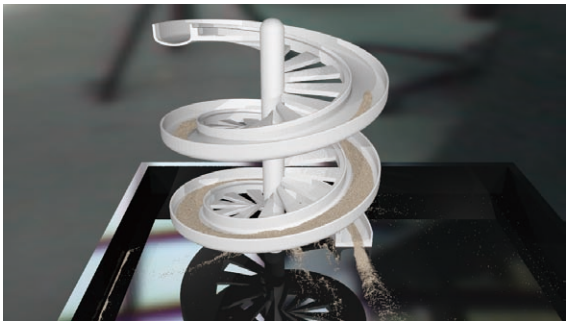


Fig.10 Sand simulation on spiral slider using 4,160,000 particles on 32 GPUs.

As a typical simulation of granular structure interaction, we apply our code to a simulation for a golf bunker shot that had been studied by 2-dimensional simulation in previous studies[5] due to the computational cost. We have successfully performed a 3-dimensional DEM simulation by using 16.7M particles with a realistic particle size on 64 GPUs of TSUBAME 2.5, which has never been done before. To have the initial condition called "eye-ball", we carried out a simulation of a golf ball falling on the sand in advance. We determine the swing path of the sand wedge by using a model based on a rotational or double pendulum. The swinging speed of the sand wedge is 5.0 m/s at the head edge. Figure 11 shows a snapshot of the simulation with 0.4-mm particles representing typical silica sand. The golf ball is pushed by the sand without the iron head of the sand wedge hitting it.

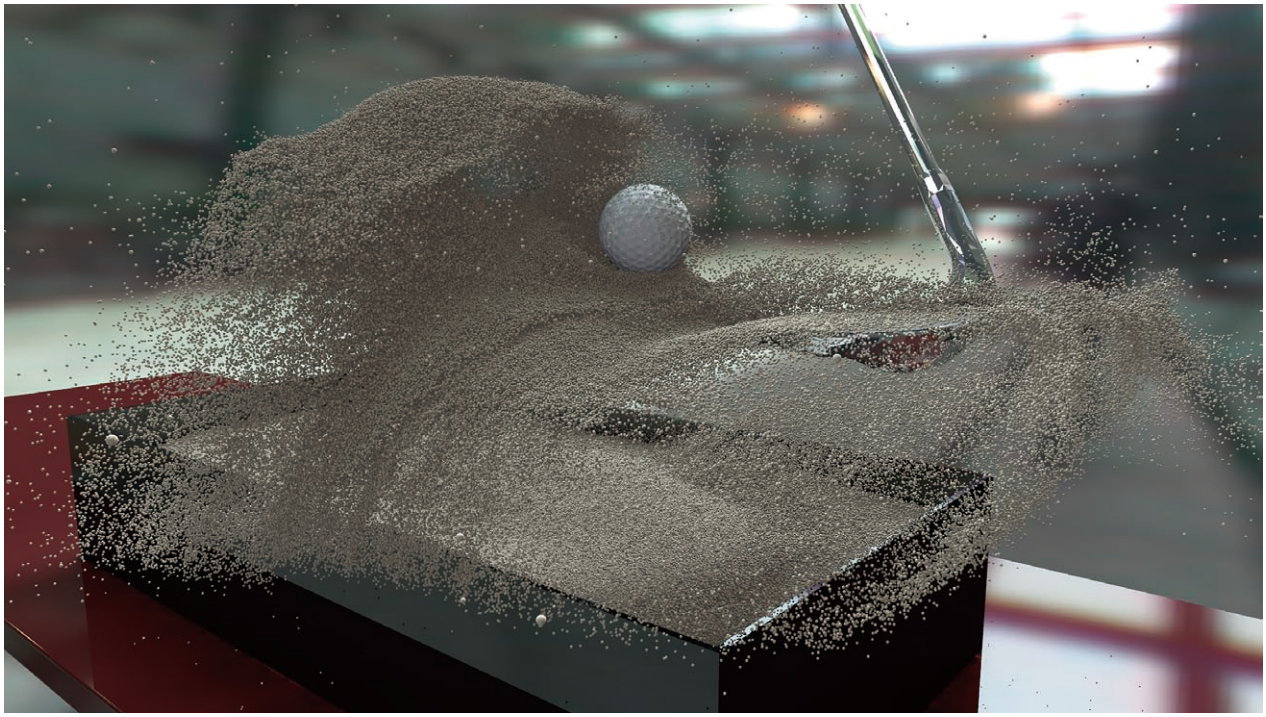


Fig.11 Golf bunker shot with 16.7M particles on 64 GPUs.

DEM simulations using Non-spherical particles 6

Real granular particles are not spherical. Using the model of non-spherical particles, we can conduct more realistic granular simulations [6]. It requires much higher computational cost and larger memory even if we use a simple model in which several spherical particles are rigidly connected. We carried out a large-scale simulation for a foot stamp using 405,000 tetrapod-shaped particles that consist of the same four spherical particles located at the vertex of a tetrahedron as shown in Fig. 12.

The numerical results are shown in Fig. 13 in comparison with using spherical particles. The footprint with the tetrahedral particles is clearly visible, and the shear friction is enhanced due to the inter-locking among tetrahedral particles.

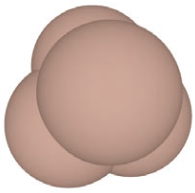


Fig.12 Tetrahedral non-spherical particle.

(a) Using spherical particles

(b) Using tetrahedral non-spherical particles

Fig.13 Foot stamp simulations.

Conclusion 7

By introducing a dynamic load balance, we successfully performed large-scale DEM simulations with a maximum of 129M particles on a GPU supercomputer, TSUBAME 2.5. The two-dimensional slice-grid method works well as a dynamic domain decomposition to keep equal memory consumption and computational load balance.

We applied the simulation code to several practical problems including a golf bunker shot, and the scalabilities were also examined. In addition, it is found that we can conduct more realistic granular simulations by using non-spherical particles.

Acknowledgements

This research was partly supported by KAKENHI, Grant-in-Aid for Scientific Research (S) 26220002 from the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan, partly by the Japan Science and Technology Agency (JST) Core Research of Evolutional Science and Technology (CREST) research program "Highly Productive, High Performance Application Frameworks for Post Peta-scale Computing", and partly by "Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures (JHPCN)" and "High Performance Computing Infrastructure (HPCI)" in Japan. The authors thank the Global Scientific Information and Computing Center, Tokyo Institute of Technology for use of the resources of the TSUBAME 2.5 supercomputer.

References

[1] G. S. Grest, B. D ¨ unweg, and K. Kremer, "Vectorized link cell Fortran code for molecular dynamics simulations for a large number of particles," Computer Physics Communications, vol. 55, pp. 269–285, Oct. 1989.

[2] Gomez-Gesteira, M., Crespo, A., Rogers, B., Dalrymple, R., Dominguez, J., and Barreiro, A.: fSPHysicsg development of a free-surface fluid solver Part 2: Efficiency and test cases, Computers and Geosciences, Vol. 48, No. 0, pp. 300-307 (2012).

[3] S. Tsuzuki and T. Aoki: Large-scale granular simulations using Dynamic load balance on a GPU supercomputer, in Poster at the 26th IEEE/ACM International Conference on High Performance Computing, Networking, Storage and Analysis (SC) 2014, New Orleans, US-LA, USA (2014).

[4] J. A. Bærentzen and H. Aanæs, "Computing discrete signed distance fields from triangle meshes," Informatics and Mathematical Modeling, Technical University of Denmark, DTU, Richard Petersons Plads, Building 321, DK-2800 Kgs. Lyngby, Tech. Rep., 2002.

[5] H. Horii, T. Koizumi, N. Tsujiuchi, M. Miki, and J. Hidaka: Computational Simulation Modeling of Bunker Shots, 5th International Conference on Engineering of Sport, September (2004)

[6] Ikuya Ono, Hiroshi Nakashima, Hiroshi Shimizu, Juro Miyasaka, and Katsuaki Ohdoi, Investigation of elemental shape for 3D DEM modeling of interaction between soil and a narrow cutting tool, Journal of Terramechanics, Volume 50, Issue 4, August 2013, Pages 265-276, ISSN0022-4898.

● TSUBAME e-Science Journal vol.13
Published 3/10/2015 by GSIC, Tokyo Institute of Technology ©
ISSN 2185-6028
Design & Layout: Kick and Punch
Editor: TSUBAME e-Science Journal - Editorial room
Takayuki AOKI, Toshio WATANABE,
Atsushi SASAKI, Eri Nakagawa
Address: 2-12-1-E2-6 O-okayama, Meguro-ku, Tokyo 152-8550
Tel: +81-3-5734-2085 Fax: +81-3-5734-3198
E-mail: tsubame_j@sim.gsic.titech.ac.jp
URL: http://www.gsic.titech.ac.jp/

TSUBAME

International Research Collaboration

The high performance of supercomputer TSUBAME has been extended to the international arena. We promote international research collaborations using TSUBAME between researchers of Tokyo Institute of Technology and overseas research institutions as well as research groups worldwide.

Recent research collaborations using TSUBAME

1. Simulation of Tsunamis Generated by Earthquakes using Parallel Computing Technique
2. Numerical Simulation of Energy Conversion with MHD Plasma-fluid Flow
3. GPU computing for Computational Fluid Dynamics

Application Guidance

Candidates to initiate research collaborations are expected to conclude MOU (Memorandum of Understanding) with the partner organizations/ departments. Committee reviews the “Agreement for Collaboration” for joint research to ensure that the proposed research meet academic qualifications and contributions to international society. Overseas users must observe rules and regulations on using TSUBAME. User fees are paid by Tokyo Tech’s researcher as part of research collaboration. The results of joint research are expected to be released for academic publication.

Inquiry

Please see the following website for more details.
http://www.gsic.titech.ac.jp/en/InternationalCollaboration

GSIC
Global Scientific Information and Computing Center