

# OpenMP Parallelization of Multicolour Particle Contact Detection Method with Computational Block Parallelism for Sorting

Takahiro Katagiri<sup>†</sup>, Hiroshi Takeda<sup>††</sup> and Yasuyoshi Horibata<sup>\*</sup>

<sup>†</sup> Information Technology Centre  
The University of Tokyo  
2-11-16 Yayoi, Bunkyo-ku, Tokyo 113-8658, Japan  
e-mail: katagiri@cc.u-tokyo.ac.jp, web page: <http://www.u-tokyo.ac.jp/>

<sup>††</sup> R-flow Corporation, Ltd  
Takanashi Bld., 1-10-45 Takasago, Soka, Saitama 34-0015, Japan  
e-mail: takeda@rflow.co.jp, web page: <http://www.rflow.co.jp/>

<sup>\*</sup> Faculty of Science and Engineering  
Hosei University  
3-7-2 Kajino-cho, Koganei, Tokyo 184-8584, Japan  
e-mail: horibata@hosei.ac.jp, web page: <http://www.hosei.ac.jp/>

We evaluate OpenMP parallel implementation of simulation for freely dropping particles within a cubic cavity in the gravitational field on a multicore processor.

In the Discrete Element Method (DEM), particle contact detection is the most important process, and consumes the largest fraction of the execution time. All particles have a chance to make contact with other particles. The calculation domain is divided into cells. The cell is a cube, the edge length of which is as long as the particle diameter. A set of cells is called contact detection grid. Particle contact detection consists of three steps. First, for each particle, we specify the cell in which it is located. Second, we sort the particles so that particles in a cell become neighbours. Third, we calculate the distance between two particles. In this paper, we propose computational block parallelism for the sorting. We divide the calculation domain into a number of computational blocks. Each computational block contains a large number of cells. Sorting is done in each computational block in a loop implemented via do construct. The loop is shared among threads.

After the particle contact detection, we calculate contact force, and then update velocity and position for each particle. We have proposed Multicolour Particle Contact Detection Method (MPCD) for calculating the contact force, and have shown the method is superior to other two conventional methods such as critical and redundant methods [1]; a critical method is an implementation with a mutual exclusion, whereas a redundant method is an implementation with a double of computations. In MPCD method, we colour all cells with eight colours. Contact force calculation is parallelized for the cells with the same colour; in the parallelization, multiple threads do not attempt to update the same shared data simultaneously.

Preliminary performance evaluation was done on the Fujitsu supercomputer system (FX10) installed at the University of Tokyo. Each node has 1 CPU with 16 cores. Initially, 18000000 particles were randomly set in a cavity. The number of computational blocks is 16. On 16 threads, the overall execution time was reduced by half compared to the conventional parallelism for the sorting.

Performance evaluation is done on a multicore processor. It has two CPUs: Intel Xeon E5-268W (3.10GHz) with 8 cores. The main memory is 128GB. Performance is evaluated for serial version, and for 1 to 16 threads versions,

In the conference, performance results will be presented for hybrid MPI-OpenMP parallel implementation, too.

## REFERENCES

- [1] T. Katagiri, H. Takeda, J. Kato, S. Kawamura, Y. Horibata, "Towards multicolor particle contact detection method for hybrid MPI-OpenMP execution in DEM", VECPAR 2014 - 11th International Meeting High Performance Computing for Computational Science, June 30 - July 3, 2014, University of Oregon and Hilton Conference Center, Eugene, Oregon, USA, (2014)