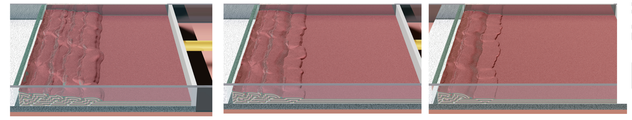
The first real-scale DEM simulation of a sandbox experiment using 2.4 billion particles

Mikito Furuichi *1, Daisuke Nishiura *1, Mitsuteru Asai *2 and Takane Hori *3

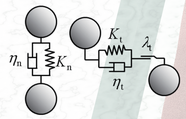
*1 Department of Mathematical Science and Advanced Technology, Japan Agency for Marine-Earth Science and Technology (JAMSTEC), Japan
*2 Department of Civil Engineering, Kyushu University, Japan
*3 R&D Center for Earthquake and Tsunam, Japan Agency for Marine-Earth Science and Technology (JAMSTEC), Japan



Introduction

- We are interested in real-scale numerical sandbox experiment with a Discrete Element Method (DEM) as a scaled analog model of thrust formations in the accretionary prisms.
- To perform realistic DEM simulation with over 2B (billion) particles, we have developed new DEM codes designed for HPC systems
- An efficient parallel algorithms and implementation techniques are proposed to overcome the load imbalance and communication cost problems.

DEM and challenges in parallel computing



$$\ddot{\boldsymbol{x}}_i = \sum_j (\boldsymbol{F}_{c_n} + \boldsymbol{F}_{c_t})_{ij} + \boldsymbol{g}\,, \quad \dot{\boldsymbol{\omega}}_i = \sum_j \boldsymbol{M}_{ij}\,/I_i$$

$$\boldsymbol{Fc}_{\mathrm{n}} = \eta_{\mathrm{n}}\boldsymbol{v}_{\mathrm{n}} + K_{\mathrm{n}}\boldsymbol{x}_{\mathrm{n}}$$

$$\boldsymbol{Fc}_{\mathrm{t}} = \min\left[\eta_{\mathrm{t}}\boldsymbol{v}_{\mathrm{t}} + K_{\mathrm{t}}\,\boldsymbol{x}_{\mathrm{t}}, \lambda_{\mathrm{t}}\left|\boldsymbol{Fc}_{\mathrm{n}}\right|\boldsymbol{v}_{\mathrm{t}}\middle/\left|\boldsymbol{v}_{\mathrm{t}}\right|\right]$$

- **Realistic sand-box simulation needs parallel computing.**
Over billions of particles are required for our target problem with realistic granular size (~100μm) with small time-step size.

- **Workload imbalance problems.**
A spatial domain decomposition is a natural approach to parallelize the DEM using MPI. However, workload imbalance problem arises because particles distribution largely changes in the course of the simulation.

- **Memory and arithmetic cost for tangential force.**
The particle interactions with tangential forces should be updated in incremental form. Thus, different from Smoothed Particle Hydrodynamics (SPH) and Molecular Dynamics (MD), contacting pairwise interaction should be stored in the memory array.

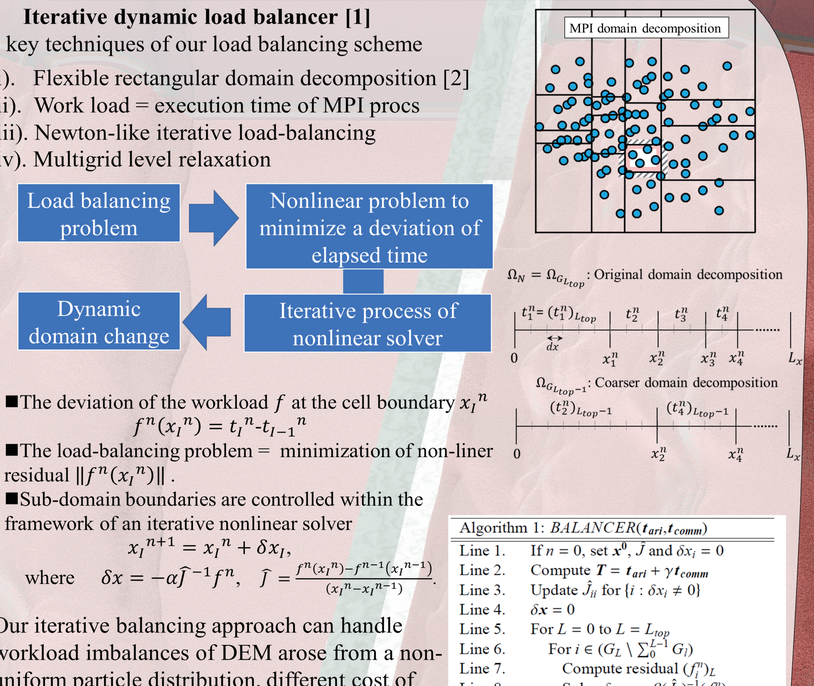
- **Efficient shared-memory access with action-reaction law.**
The use of action-reaction law can save the memory size and halve the arithmetic costs, but the memory access conflict should be avoided.

Key techniques

- **Iterative dynamic load balancer [1]**

4 key techniques of our load balancing scheme

i). Flexible rectangular domain decomposition [2]
ii). Work load = execution time of MPI procs
iii). Newton-like iterative load-balancing
iv). Multigrid level relaxation



■The deviation of the workload f at the cell boundary $x_I{}^n$
$$f^n(x_I{}^n) = t_I{}^n\text{-}t_{I-1}{}^n$$
■The load-balancing problem = minimization of non-liner residual $\|f^n(x_I{}^n)\|$.
■Sub-domain boundaries are controlled within the framework of an iterative nonlinear solver
$$x_I{}^{n+1} = x_I{}^n + \delta x_I,$$
where $\delta x = -\alpha\hat{J}^{-1}f^n$, $\hat{J} = \frac{f^n(x_I{}^n) - f^{n-1}(x_I{}^{n-1})}{(x_I{}^n - x_I{}^{n-1})}$.

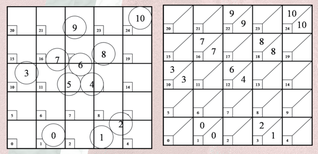
Our iterative balancing approach can handle workload imbalances of DEM arose from a non-uniform particle distribution, different cost of individual particles, and heterogeneous computer architecture which were difficult with previously proposed methods.

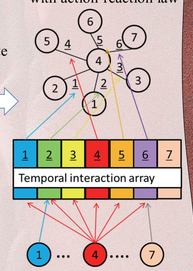
Algorithm 1: $BALANCER(\boldsymbol{t}_{ari}, \boldsymbol{t}_{comm})$

Line 1. If $n = 0$, set $\boldsymbol{x}^0$, $\hat{J}$ and $\delta x_i = 0$
Line 2. Compute $\boldsymbol{T} = \boldsymbol{t}_{ari} + \gamma \boldsymbol{t}_{comm}$
Line 3. Update $\hat{J}_{ij}$ for $\{i : \delta x_i \neq 0\}$
Line 4. $\delta \boldsymbol{x} = 0$
Line 5. For $L = 0$ to $L = L_{top}$
Line 6. For $i \in (G_L \setminus \sum_0^{L-1} G_l)$
Line 7. Compute residual $(f_i^n)_L$
Line 8. Solve $\delta x_i = -\beta(\hat{J}_{ii})^{-1}(f_i^n)_L$
Line 9. Update $\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \delta \boldsymbol{x}$
Line 10. If $\delta \boldsymbol{x} = 0$, control γ with line search
Line 11. Return

- **Shared-memory optimization with action-reaction law using sorting technique.**

The sorting technique is applied to create neighboring particle pair list based on the cell labels [3].





Efficient performance with action-reaction law is obtained by using the temporal memory array of particle interactions [4].

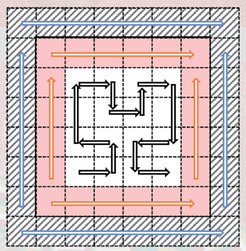
The load balancer and sorting techniques are applied only when the cumulative displacement of a particle exceeds a certain distance, not at every time step.

- **Consistency of particle ordering**

To ensure that consistent relations between action and reaction law, particles in the halo and corresponding cell in adjacent MPI subdomain are ordered by the initial particle IDs (= partially stable sort).

- **Overlapping communication with space filling curve (SPC)**

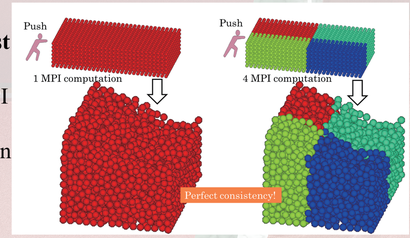
Key idea is to optimize the memory access, particle data are separately ordered in the halo, outermost and inner cells. In addition, we reorder the cell label with the Hilbert curve for a cache-aware data access.



Halo (striped), outer most (pink) and inner cells (white) in MPI subdomain. The arrows indicate the order of memory allocation

Results

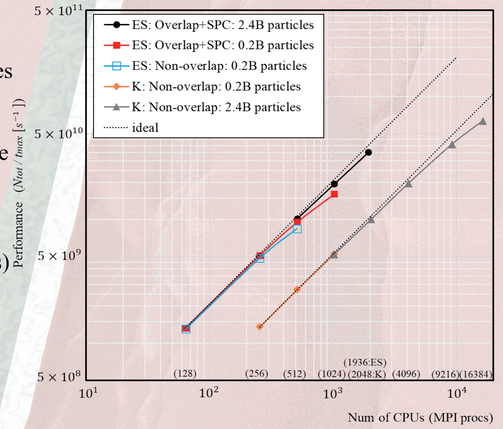
- **Reproducibility test**
The two solutions with different number of MPI processes showed the exactly same solution. In this test, the share-memory parallelization was switched off.



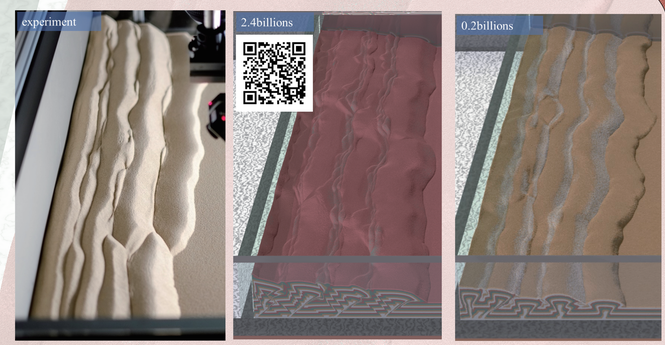
- **Impact of ordering with SPC**
The performance improvement on ADB (cache) hit ratio was 1.26% by using space filling curve.

	overlap	overlap + SPC
Elapsed time [msec]	123.1	120.9
ADB hit ratio [%]	74.24	75.505

- **Scaling test**
Scaling test with 0.2B particles shows the performance improvement using the overlapping technique. For the problem with 2.4B particles, strong scaling behaviors were confirmed upto 1,936 CPUs/MPI procs (7,744 cores) for the Earth Simulator (ES: 8,192 cores) and 16,384 CPUs/MPI procs (131,072 cores) for the K-computer.



- **Sand-box experiment**
The world's first real scaled sandbox simulation was performed with 2.4B particles by using ES and K computer (right and background figures). The domain size is 1 m x 1 m x 0.02 m. The load imbalance (t_max- t_min)/t_ave observed around at the time step in the right figure was controlled under 7.4%.



Averaged particle size = 100μm | Averaged particle size = 100μm | Averaged particle size = 200μm

Summary

The world's first real-scale sandbox simulation was performed with our DEM code developed with iterative load-balancer, efficient implementation of action-reaction law and overlapping communication with space filling curve. Our next challenge is fluid-particle coupling method because our load balancer naturally handles the different computational cost between DEM and fluid (e.g. SPH) particles.

ACKNOWLEDGMENTS

This study was supported by the Post-K project of (MEXT) using the K computer at the RIKEN (Project ID:hp160221), Earth Simulator project of JAMSTEC (Project name: Development of Advanced Simulation Methods for Solid Earth Simulations) and a Grant-in-Aid (Project ID:JP15K17754) from the Japan Society for the Promotion of Science (JSPS).

REFERENCES

[1]M. Furuichi and D. Nishiura, 2017, Phys. Commun. DOI: 10.1016/j.cpc.2017.05.015
[2] J. Makino, 2004. A Fast Parallel Treecode with GRAPE. PASJ. 56, 521–531
[3] D. Nishiura, H. Sakaguchi, 2011, J. Comput. Phys. Doi:10.1016/j.jcp.2010.11.040
[4] D. Nishiura, M. Furuichi and H. Sakaguchi, 2015. Phys. Commun. DOI:10.1016/j.cpc.2015.04.006
[5] S. Tsuzuki and T. Aoki. 2016. Proceedings of the 7th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems ScalA. 16, 1–8.