VII International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2017
M. Papadrakakis, E. Oñate and B. Schrefler (Eds)

LARGE-SCALE SIMULATIONS OF VISCOELASTIC DEFORMABLE MULTI-BODY SYSTEMS USING QUADRUPLE DISCRETE ELEMENT METHOD ON SUPERCOMPUTERS

S. TSUZUKI*, D. NISHIURA*, H. SAKAGUCHI*

* Japan Agency for Marine-Earth Science and Technology (JAMSTEC)
3173-25, Showa-machi, Kanazawa-ku, Yokohama-city, Kanagawa, 236-0001, Japan
E-mail : tsuzukis@jamstec.go.jp
Web page : http://str-prtc-gpu-comp.wixsite.com/pm-info-st-personal

Key words: Particle-based Methods, Multi-body Problems, Contact Problems, Viscoelastic Materials, Parallel Computing.

Abstract. Contact problems among viscoelastic materials in the multibody system is one of the challenging topics in science and many engineering applications. We have developed an effective simulation method of combining QDEM (Quadruple Discrete Element Method) for the deformation analysis of structures with the DEM for the collisions among structures. However, it is still difficult to reproduce surface topography of structures because particles only set on four nodes of tetrahedrons in our current method. In this paper, QDEMSM (QDEM with Surface Modeling) is newly developed. Point-polygon collisions and line-line collisions are effectively coupled with QDEM. Our improved method was validated by several simulation results; domino simulations using the 40 pieces of shogi (= Japanese chess) were successfully carried out. It was also found the friction forces acted on the surface critically effected on the propagation speeds of contact forces. In parallel computing, by applying the space-filling curve to decomposition of the computational domain, we make it possible to contain the same number of nodes in each decomposed domain. Our parallel simulation code achieves a good weak scalability on the TSUBAME2.5 supercomputer.

1 INTRODUCTION

It is crucial there to realize large-scale viscoelastic multi-body simulations in many fields (e.g. civil engineering, geoscience, and chemical engineering). Particle-based methods are promising approaches, however, it is still challenging because of the difficulties on several numerical problems; combining the deformation analysis of each structure with the contact analysis among contacting structures is not easy; surface topography of structures should be represented with a sufficiently accurate because friction forces acted on the surfaces are dominant in contact problems; three-dimensional simulations are essential in practical problems; parallel computing on memory-distributed systems is indispensable to realize realistic simulations. For all these reasons, there are few studies on large-scale 3-dimensional viscoelastic multi-body simulations in previous studies.

In this study, we present an effective method of large-scale 3-dimensional viscoelastic multi-body simulations using Quadruple Discrete Element Method (QDEM) [1] on supercomputers. QDEM, which is an explicit particle-based method for viscoelastic deformable materials originally designed to be compatible with the memory-distributed

system, enables easy implementation on supercomputers. The concept of QDEM is straightforward compared with finite-element approaches because QDEM requires only an inter-particle relationship among four particles which consist a quadruple discrete element (for details, see reference [1]). Despite its simplicity, it is known that QDEM has the same numerical accuracy as explicit FEM using primary element [2].

In our previous work, we had developed a simulation method of combining QDEM for deformation analysis of structures with the DEM for collisions among structures for large-scale viscoelastic multi-body problems, which brought a certain level of successful results [3]. However, in our current method, it is difficult to reproduce surface topography of structures because particles only set on four nodes of tetrahedrons. Introduction of an envelope curve by particle-overlapping technique is one approach, but is not compatible with QDEM algorithm. On the contrary, it is reasonable to introduce the collision among triangles (=polygons) of tetrahedrons. However, methodology of it has not been established.

For this purpose, an effective method of QDEMSM (QDEM with Surface Modeling) is newly developed in this paper. We also demonstrate several simulations for practical viscoelastic multi-body problems.

2 A BRIEF OVERVIEW OF QDEM

QDEM is based on a particle concept; a tetrahedron is defined as a representation of the interacting part of four particles (=quadruple discrete element). An inter-particle relationship among neighboring four particles is derived from two rheology models of isotropic elastic solid and viscos fluid. A schematic-view of QDEM and its mathematical formulas are shown at a glance in **Figure.1**. Note that particle motions are updated by the explicit Verlet time integration scheme [4] in this paper. For details, see reference [1, 2].

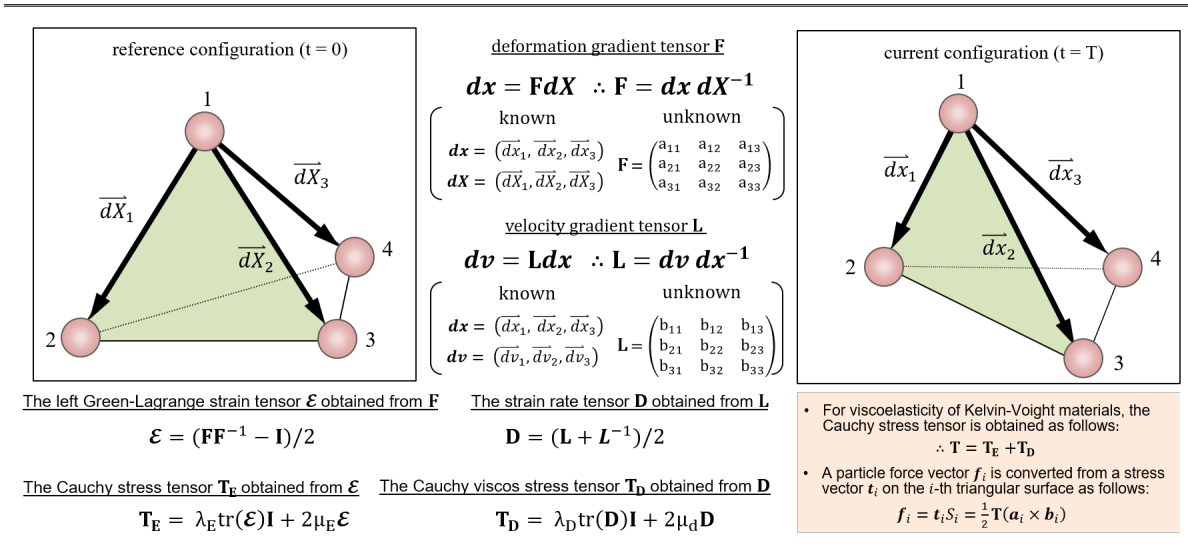


Figure 1 : A schematic-view of QDEM and mathematical formulas in QDEM algorithm.

3 SURFACE MODELING WITH QDEM

There exist the six kinds of possible collisions among the triangles of tetrahedrons. It is idealistic to compute all the six types of collisions. However, it is difficult because of both the high computational costs and lack of methodology. By taking account of real physical

phenomena, (c) and (d) are considered to be dominant in all the possible collisions. Therefore, in this paper, we experimentally introduce calculations of (c) and (d) as the first step toward realistic viscoelastic multi-body simulations.

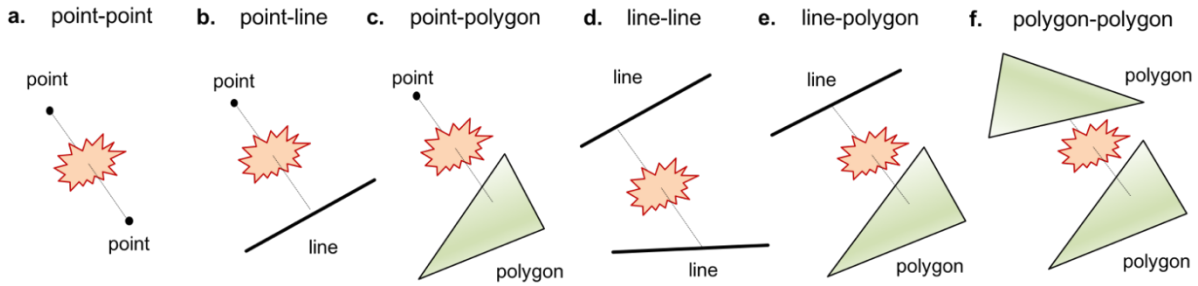


Figure 2 : Six kinds of possible collisions among the triangles(=polygons) of tetrahedrons.

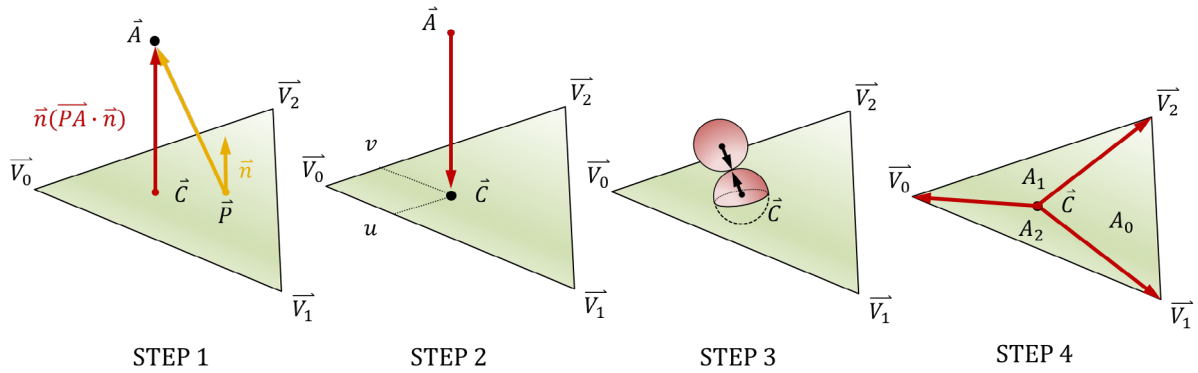


Figure 3 : A four-step procedure to compute point-polygon collisions.

3.1 Point-Polygon Collision

Figure.3 shows a four-step procedure for point-polygon collisions among the triangular surfaces of different tetrahedrons.

First, we calculate the closest point $\vec{C}$ on a triangle from a point $\vec{A}$ as follows:

$$\vec{C} = \vec{A} - \vec{n}(\overrightarrow{PA}\cdot\vec{n}) \tag{1}$$

Here, $\vec{P}$ is an arbitrary point on the triangle, $\vec{n}$ is the normal vector of the triangle, $\vec{A}$ is the center position of a particle, respectively. The geometric relationship is depicted in STEP1.

Second, we judge whether the point $\vec{C}$ is inside the triangle. The point $\vec{C}$ can be represented in two ways as follows:

$$\vec{C} = \overrightarrow{V_0} + u * \overrightarrow{V_0V_1} + v * \overrightarrow{V_0V_2} \tag{2}$$

$$\vec{C} = \vec{A} + t * \overrightarrow{AC} \tag{3}$$

Here, t, u, and v is scalar parameters as shown in STEP2. A 3×3 system of simultaneous equations is obtained by (2) and (3) as follows:

$$u * \overrightarrow{V_0V_1} + v * \overrightarrow{V_0V_2} - t * \overrightarrow{AC} = \vec{A} - \overrightarrow{V_0} \quad (4)$$

The solution of (4) is obtained by using Cramer's Rule. In case of (u, v) fulfills $u \geq 0,\ v \geq 0, u + v \leq 1$, the point $\vec{C}$ is on the triangle. In this case, we assume a virtual particle set on the point $\vec{C}$ and calculate a contact force between two particles by DEM as depicted in STEP3.

Finally, the contact force $\vec{F}$ is divided into each point of the triangle as shown in STEP4. Here, $A_0, A_1,$ and A_2 are the area of $\overrightarrow{CV_1V_2}$, the area of $\overrightarrow{CV_2V_0}$, and the area $\overrightarrow{CV_0V_1}$ respectively. For the simplicity, the divided forces $\overrightarrow{F_0}$, $\overrightarrow{F_1}$, and $\overrightarrow{F_2}$ are simply calculated as follows in this paper.

$$\overrightarrow{F_0} = \frac{A_0}{A}\vec{F}, \overrightarrow{F_1} = \frac{A_1}{A}\vec{F}, \overrightarrow{F_2} = \frac{A_2}{A}\vec{F} \quad (A = A_0 + A_1 + A_2) \quad (5)$$

3.2 Line-Line Collision

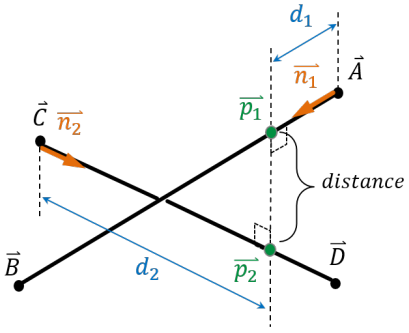
The closest points between two lines are obtained by the four-step procedure as shown in **Figure.4**. In case that $d_1 \geq 0\ and\ d_1 \leq \left|\overrightarrow{AB}\right|\ and\ d_2 \geq 0\ and\ d_2 \leq \left|\overrightarrow{CD}\right|$, two lines intersect each other. Thereafter, we compare $\left|\overrightarrow{p_1p_2}\right|$ with the interaction range of DEM calculations. In case the collision is detected, virtual particles are set on the closest points of each line, then the contact forces are calculated by using DEM. For the simplicity, the contact force for normal direction is only considered in case of line-line collisions in this paper.

1. We have four geometric relationships between two lines.

$$\begin{cases} \overrightarrow{n_1} \cdot (\overrightarrow{p_2} - \overrightarrow{p_1}) = 0 & \cdots\cdots (1) \\ \overrightarrow{n_2} \cdot (\overrightarrow{p_2} - \overrightarrow{p_1}) = 0 & \cdots\cdots (2) \end{cases} \qquad \begin{cases} \overrightarrow{p_1} = A + d_1 * \overrightarrow{n_1} & \cdots\cdots (3) \\ \overrightarrow{p_2} = C + d_2 * \overrightarrow{n_2} & \cdots\cdots (4) \end{cases}$$

2. (5) and (6) are obtained by substituting "(4)-(3)" to (1) and (2) respectively.

$$\begin{cases} \overrightarrow{n_1} \cdot \overrightarrow{AC} + d_2 * (\overrightarrow{n_1} \cdot \overrightarrow{n_2}) - d_1 = 0 & \cdots\cdots\cdots\cdots (5) \\ \overrightarrow{n_2} \cdot \overrightarrow{AC} - d_1 * (\overrightarrow{n_1} \cdot \overrightarrow{n_2}) + d_2 = 0 & \cdots\cdots\cdots\cdots (6) \end{cases}$$



3. d_1, d_2 are obtained by solving the simultaneous linear system of (5) and (6).

$$d_1 = \frac{(\overrightarrow{n_1} \cdot \overrightarrow{AC} - (\overrightarrow{n_1} \cdot \overrightarrow{n_2}) * (\overrightarrow{n_2} \cdot \overrightarrow{AC}))}{(1 - (\overrightarrow{n_1} \cdot \overrightarrow{n_2}) * (\overrightarrow{n_1} \cdot \overrightarrow{n_2}))} \cdots\cdots\cdots\cdots (7) \qquad d_1 = \frac{((\overrightarrow{n_1} \cdot \overrightarrow{n_2}) * (\overrightarrow{n_1} \cdot \overrightarrow{AC}) - \overrightarrow{n_2} \cdot \overrightarrow{AC})}{(1 - (\overrightarrow{n_1} \cdot \overrightarrow{n_2}) * (\overrightarrow{n_1} \cdot \overrightarrow{n_2}))} \cdots\cdots\cdots\cdots (8)$$

4. $\overrightarrow{p_1}$, $\overrightarrow{p_2}$ are obtained by (3), (4), (7), and (8).

Figure 4 : A four-step procedure to compute the closest points between two lines.

4 IMPLEMENTATION

An effective implementation for the computation of QDEM with Surface Modeling on parallel environments is presented by taking account of the locality of QDEM algorithm; QDEM requires only an inter-particle relationship among neighboring four particles (quadruple discrete element) even though structures are represented by groups of particles.

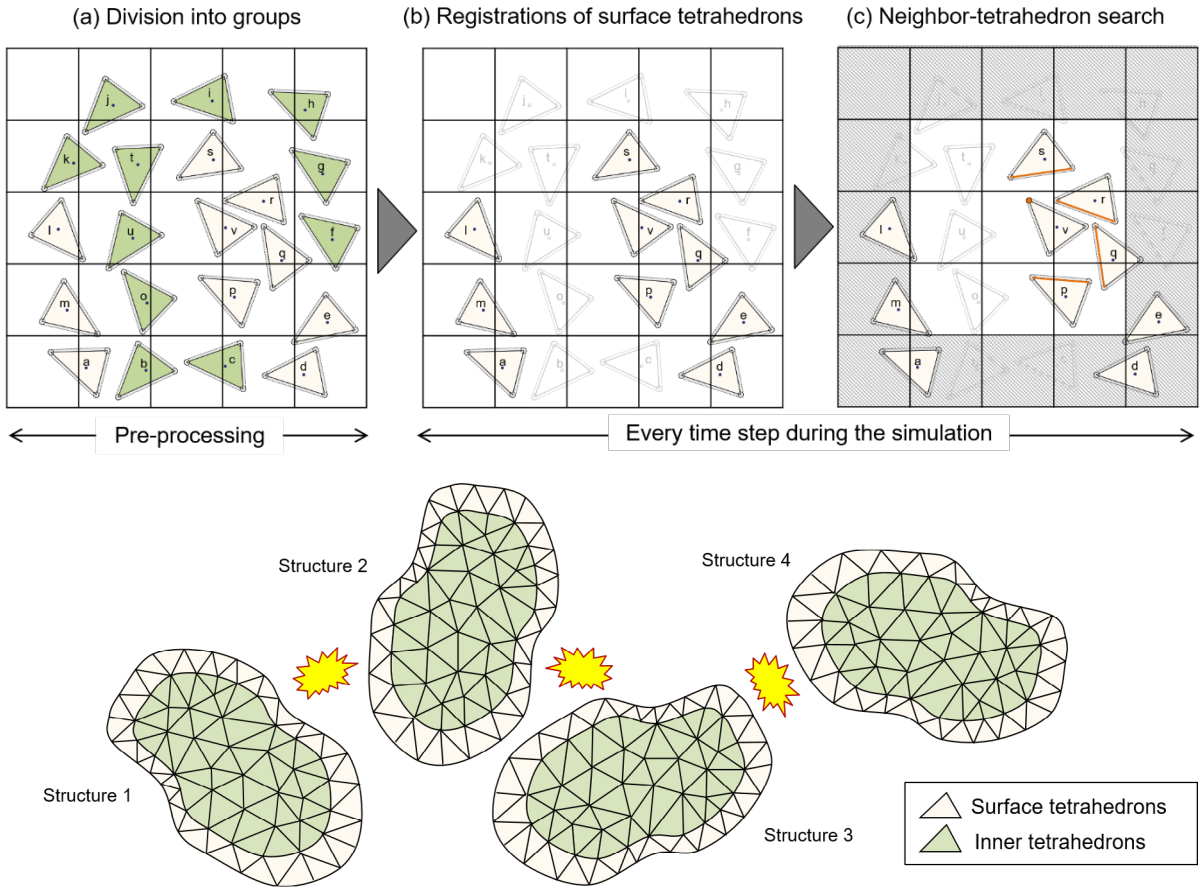


Figure 5 : An schematic view of effective neighbor tetrahedron search (upper) and an illustration of collisions among structures (lower).

4.1 Multithread Computing of QDEM

As well as accelerators, such as GPU or MIC, even scalar-type computers have multiple processing cores on a single chip. Therefore, fine-grain parallelization and multithread programming are required to obtain high performance. In multithread computing of QDEM, the dependent variables of a node (e.g. position, velocity) are defined for each particle. On the other hand, the dependent variables of a quadruple discrete element (e.g. the stress, the four-node connectivity) are defined for each tetrahedron. In the thread assignment for multi-cores, one thread computes one particle in case of point-polygon collision calculations, and likewise one thread computes one tetrahedron in case of line-line collision calculations.

4.2 Neighbor-Tetrahedron List

The (a) in **Figure.5** depicts an imaginary case that all the tetrahedrons (all the triangles in 2D case) are scattered in a computational domain. As with general particle-based simulations, it is inefficient to compute collisions among all the tetrahedrons. Neighbor-tetrahedron lists are used to reduce the computational cost from $O(\mathrm{N}^2)$ to $O(\mathrm{N})$ to find the tetrahedrons in the neighboring cells. A linked-list technique [5] is coupled with neighbor-tetrahedron lists; each

tetrahedron has a memory pointer referring to the next tetrahedron in the same cell. By using chain access of the linked-list, it becomes possible to drastically reduce the memory usage.

In realistic situations, some tetrahedrons are connected to form structures as depicted in the lower column in Figure.5. Even in such cases, the deformation analysis by QDEM is carried out separately in each tetrahedron. Therefore, it is possible to reduce the amount of collision calculations by registering only tetrahedrons which consist the surface of structures. In pre-processing, we divide tetrahedrons into two groups: surface tetrahedrons and inner tetrahedrons. In every time step during the simulation, we register only surface tetrahedrons to the cells as in (b) in Figure.5. Thereafter, we compute point-polygon collisions as shown in (c), followed by line-line collisions.

A constraint is imposed on L_{cell}, the length of a cell of the neighbor-tetrahedron lists, making it larger than the maximum interaction-range of collisions among tetrahedrons; namely:

$$L_{cell} \geq max_{i=0,1,2\ldots N_{tet}}\{D^i_{max}\} + D_{DEM}, \tag{6}$$

Here, N_{tet} is the number of tetrahedrons in the whole computational domain, D_{DEM} is a delimeter of a particle set on the surface. D^i_{max} is the delimeter of a sphere which is large enough to be able to contain i-th tetrahedron.

There exist several ways to determine D^i_{max}; applying the circurmspheres of tetrahedrons is one approach, but it is often too large. In case that tetrahedrons are registered to the cells by their center positions, D^i_{max} can be obtained by:

$$D^i_{max} = 2.0 * \max(d^i_1, d^i_2, d^i_3, d^i_4) \tag{7}$$

Here, d^i_j is a distance between j^{th} vertex and center of i^{th} tetrahedron (j=1,2,3,4,).

4.3 Parallel Computing on the Memory-distributed System

An entire computational domain is decomposed into several sub-domains and the processing unit is assigned to each compute tetrahedrons in the sub-domain. Because of a layer that has the same size as L_{cell} must be copied to the neighboring sub-domains, we pack both the nodes and the tetrahedrons in halo area to a buffer and exchange them by MPI communications [6].

In this paper, we apply the space-filling curve to decomposition of the computational domain because the locality of space-filling curves is compatible with the QDEM algorithm. The computational load of each decomposed domain is dynamically balanced by domain re-decomposition techniques. For details of dynamic domain decomposition using space-filling curves, see reference [7]. Note that our parallel simulation code achieves a weak scalability with 5120 cores on the TSUBAME2.5 supercomputer [8].

4.4 Numerical Experiments

Figure.6 shows a breakdown of the time required for a static analysis including 27 cubes with 33,507 particles and 165,888 tetrahedrons. Each cube has 1,241 nodes (386 for surface nodes) and 6,144 tetrahedrons (2,028 for surface tetrahedrons). The benchmark test was carried out using two sockets of the Intel CPU Xeon X5670 (Uestmere-EP) 2.93 GHz 6-core.

For the sake of simplicity, point-polygon collisions were only introduced. We measured three computational cases: (A) all the tetrahedrons, (B) only surface tetrahedrons, (C) skipping the calculation of non-surface triangles on surface tetrahedrons. The result clearly shows that our method makes it possible to drastically reduce the computational costs.

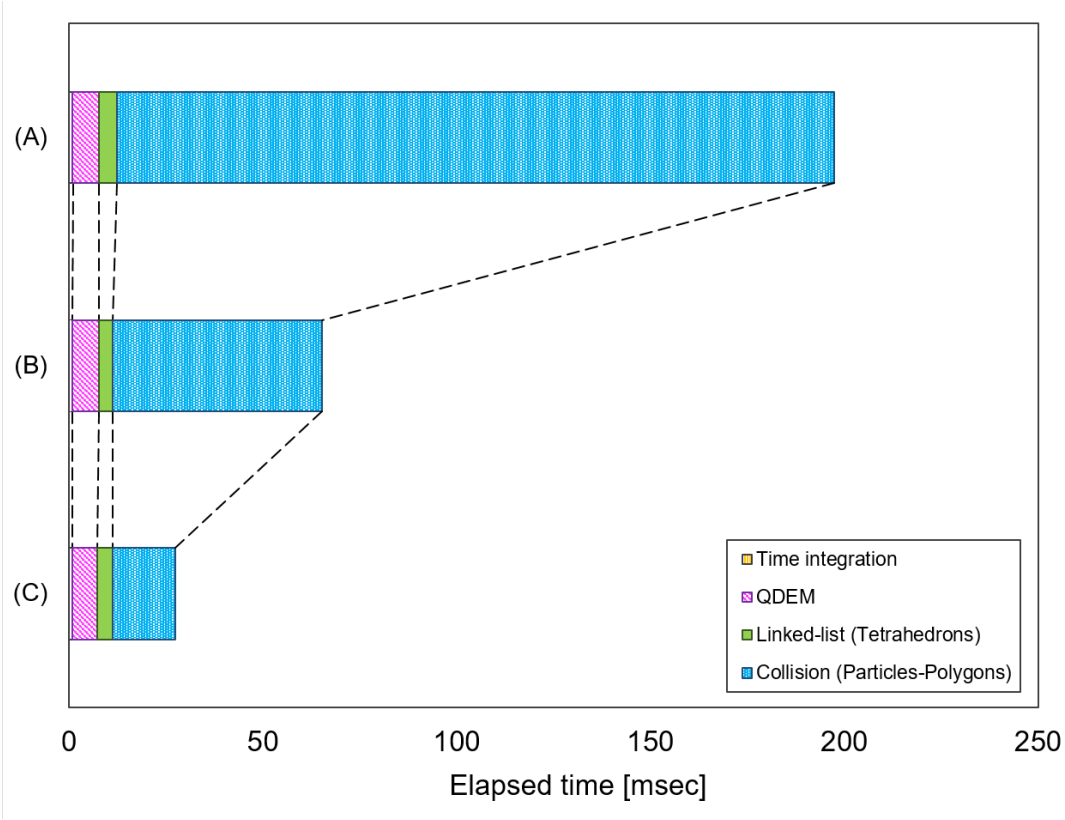


Figure 6 : Breakdown of the time required for a static analysis of 165,888 tetrahedrons.

5 APPLICATION TO A PRACTICAL PROBLEM

We demonstrate domino simulations using shogi pieces (= the pieces of Japanese chess). Unlike the ordinary domino pieces, the shapes of shogi pieces are not rectangular. A single player has 8 kinds of pieces (one king, one rook, one bishop, two gold generals, two silver generals, two knights, two lances, and nine pawns); we have 20 pieces in total for each player. The shapes of these 20 pieces are same, but the scale is different each other. Input physical parameters of our shogi simulations are listed in **Table.1**.

Figure.7 shows snapshots of a shogi simulation of two rooks in case of different kinetic friction coefficients: $\mu = 0.1$ (right), $\mu = 0.5$ (center), and the overlap view of both cases (left). It is cleared that difference of friction forces delayed the propagation speeds of contact forces. **Figure.8** shows snapshots of a shogi simulation of one king and two pawns. It was found that our method could reproduce the effect of eccentric collisions in real problem. **Figure.9** shows snapshots of a shogi simulation of all the forty shogi pieces. A snapshot of stress analysis is shown in the lower-left in Figure.9. **Figure.10.** shows a snapshot of shogi simulations of all the forty pieces in different kinetic friction coefficients: $\mu = 0.1$ (red), $\mu = 0.5$ (yellow). As shown in the two-shogi simulation in Figure.7, it was found that the difference of friction forces delayed the propagation speeds of contact forces.

Table 1 : A numerical condition for domino simulations using shogi pieces.

Initial parameters	Values
Radius of particle [m]	0.04
Density [kg/m^3]	700
Poisson's ratio	0.4
Young's modulus	5.0e+8
Viscosity coefficient	2.5e+5
Kinetic friction coefficient (piece)	0.1
Kinetic friction coefficient (wall)	0.5
Gravity [m/s^2]	9.80665
Number of points / (piece)	797
Number of tetrahedrons / (piece)	2,566
Time step size [s]	6.4e-6

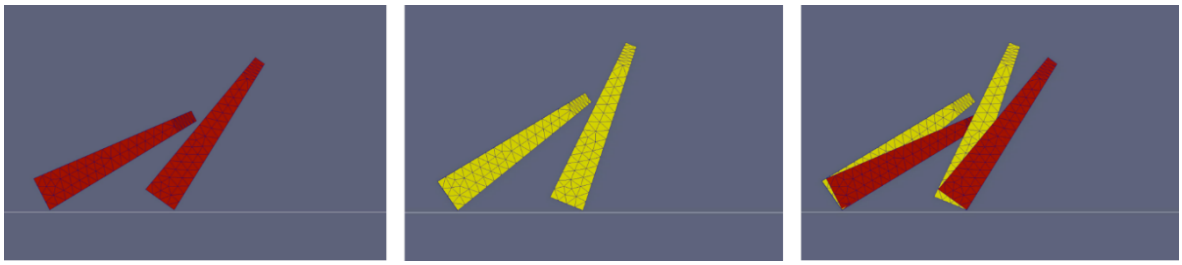


Figure 7 : A domino simulation of two rooks in case of different kinetic friction coefficient.

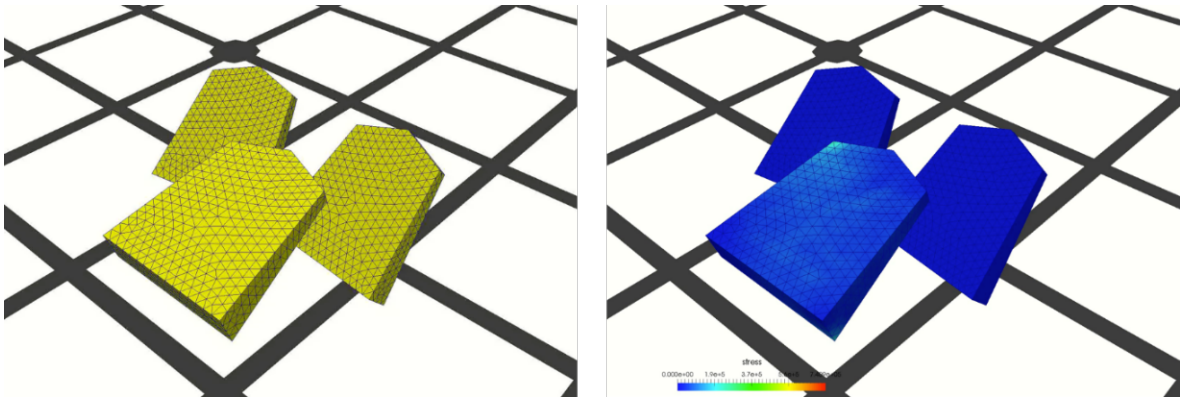


Figure 8 : A domino simulation of one king and two pawns.

6 CONCLUSIONS

An implementation method for effective computations of QDEMSM (QDEM with Surface Modeling) on parallel computing environments is presented.

Our proposal method was validated by several simulation results; domino simulations using the 40 pieces of shogi (= Japanese chess) were successfully carried out. It was also found the friction forces acted on the surface critically effected on the propagation speeds of contact forces.

In parallel computing, by applying the space-filling curve to decomposition of the computational domain, we make it possible to contain the same number of nodes in each decomposed domain. Our parallel simulation code achieves a good weak scalability on the TSUBAME2.5 supercomputer.

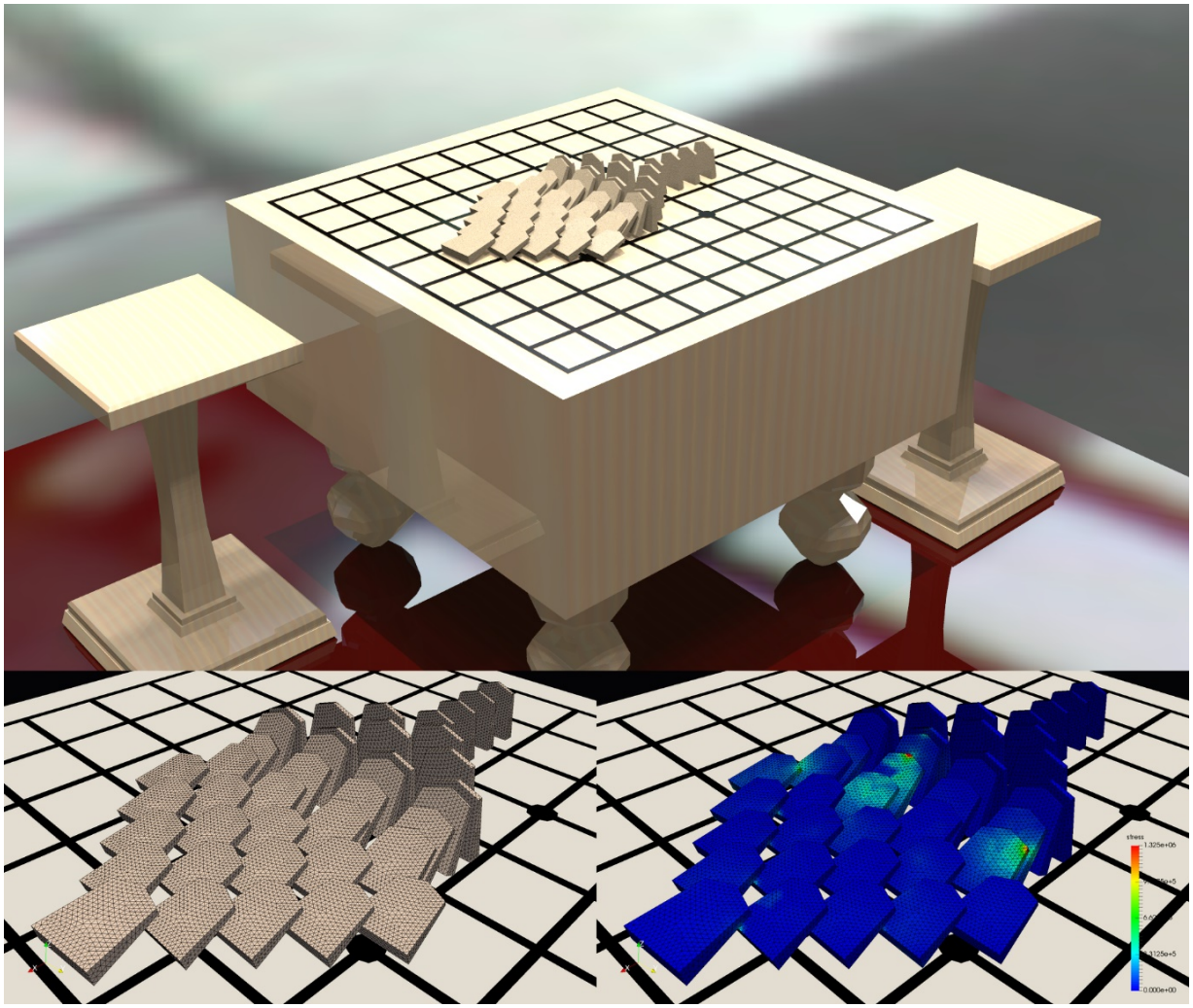


Figure 9 : A domino simulation of all the forty shogi pieces.

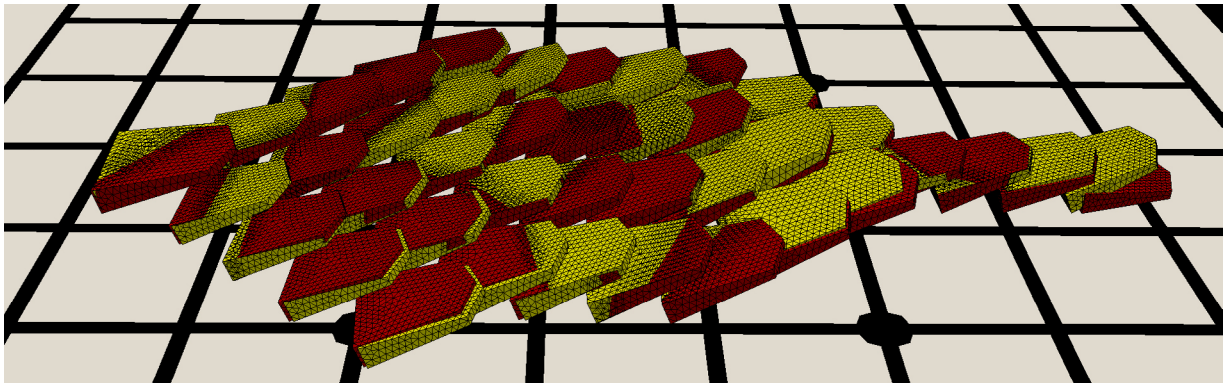


Figure 10 : A snapshot of shogi simulations of all the forty pieces in case of different kinetic friction coefficients: μ = 0.1 (red), μ = 0.5 (yellow).

Acknowledgements

This research was supported by ”Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures (JHPCN)” and ”Grant-in-Aid for JSPS Fellows” in Japan.

REFERENCES

[1] Sakaguchi, H., New computational scheme of discrete element approach for the solid earth multi-materials simulation using three-dimensional four particle interaction. *FRONTIER RESEARCH EARTH EVOLUTION, Vol. II,* (2003).

[2] Nishiura, D., Sakaguchi, H., and Aikawa, A., QDEM: A simple explicit method for deformation analysis of linear viscoelastic materials. *Journal of Computational Physics* (submitting).

[3] Nishiura, D., Sakaguchi, H., Sakai, H., and Aikawa, A., Discrete element modeling of ballasted railway track. *The 3rd International Workshops on Advances in Computational Mechanics* (2015).

[4] Verlet, L., Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules, *Phys. Rev. 159: 98* (1967)

[5] 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 (1989).

[6] MPI Forum, Message Passing Interface (MPI) Forum Home Page, http://www.mpi-forum.org (2009)

[7] S. Tsuzuki, T.Aoki, “Effective Dynamic Load Balance using Space-Filling Curves for Large-scale SPH Simulations on GPU-rich Supercomputers”, *Proceedings of the 6th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems. ACM, New York, NY, USA* (2016)

[8] TSUBAME 2.5 hardware software specifications. [Online]. Available: http://www.gsic.titech.ac.jp/sites/default/files/spec25e1.pdf