NUMERICAL ANALYSIS ON DEFORMATION OF WAVE DISSIPATING BLOCKS BY GPU-ACCELERATED DEM COMPUTATION

Naoki Yoshifuji[1], Hiroyuki Ikari[2] and Hitoshi Gotoh[3]

[1] Dept. of Civil and Earth Resources Eng., Kyoto Univ., yoshifuji.naoki.78n@st.kyoto-u.ac.jp
[2] Dept. of Civil and Earth Resources Eng., Kyoto Univ., ikari@particle.kuciv.kyoto-u.ac.jp
[3] Dept. of Civil and Earth Resources Eng., Kyoto Univ., gotoh@particle.kuciv.kyoto-u.ac.jp

In a breakwater covered with wave dissipating blocks, a leg of a wave dissipating block and wall of caisson can be broken due to high waves. To simulate this kind of phenomenon, a numerical model is required to treat deformation and failure of a block. In order to simulate them in detail, a block must be composed of sufficient number of elements, and a time increment should be set smaller to satisfy CFL condition which is limited by allowable tensile displacement. These requirements lead to a high computational load. Therefore, in this study, GPGPU with OpenCL is introduced to accelerate DEM computation. Deformation and failure process of wave dissipating blocks under high waves is simulated.

Key Words **:** *Wave dissipating block, Distinct Element Method, OpenCL, GPGPU*

1. INTRODUCTION

A breakwater covered with wave dissipating blocks is one of the most popular breakwaters. In these days, some analyses on a wave dissipating block have been executed by DEM, however, blocks in previous studies were not always suitable for a detailed investigations because they were composed of a few elements or treated as a rigid body.

Passively moving solid model (Koshizuka et al., 1998) can be applied to connect DEM particles to treat as a rigid body. Gotoh et al. (2005) simulated compaction process of wave dissipating blocks due to high waves by this model. However, in fact, because failure of a leg of block or a caisson wall can occur, it is desirable to treat deformation and failure of a block.

A block should be composed of sufficient number of particles for accurate simulation. A time increment should be set smaller based on CFL condition, because a block made of concrete has very small allowable tensile strain. This fact lead to a high computational load. Heterogeneous computation, in which not only CPU but also a processor other than CPU (e.g. GPU) are used, is expected to accelerate DEM computation and to enable to analyze wave dissipating blocks which can be deformed and fail.

In this study, GPGPU with OpenCL is introduced to accelerate DEM, and deformation and failure process of wave dissipating blocks in a breakwater is simulated.

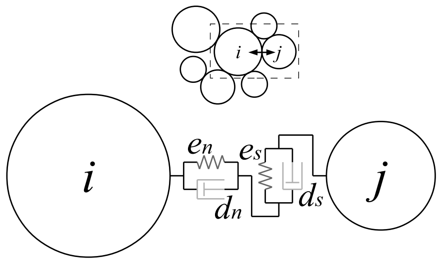


Fig. 1 Kelvin-Voigt model

2. COMPUTATIONAL METHOD

(1) Distinct Element Method

Distinct Element Method (DEM) is a numerical model proposed by Cundall and Strack. (1979) based on an overlap between two elements. A force is evaluated by Kelvin-Voigt model, which is composed of elastic spring and viscous dash-pot as shown **Fig.1**. A spring constant and viscous modulus are based on Hertzian elastic collision theory and the critical condition of 1-dimensional damping theory as follows:

$$k_n = \frac{2}{3}\frac{E}{1-\nu^2}\sqrt{\delta_n \frac{d_i d_j}{2\left(d_i + d_j\right)}} \qquad (1)$$

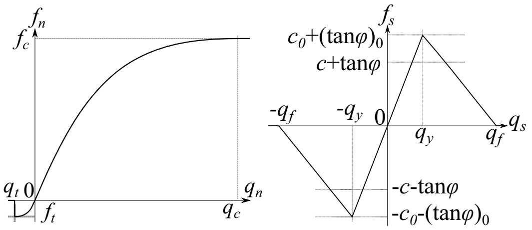


Fig. 2 Force and displacement

$$k_s = \frac{1}{2(1+\nu)} k_n \tag{2}$$

$$c_n = 2\sqrt{M k_n} \tag{3}$$

$$c_s = \frac{1}{\sqrt{2(1+\nu)}} c_n \tag{4}$$

where k_n is the spring constant for normal force, k_s is the spring constant for shear force, c_n is the viscous modulus for normal force, c_s is the viscous modulus for shear force, E is the Young's elastic modulus, ν is Poisson's ratio and M is the mass of a particle. Young's modulus and Poisson's ratio are not equal to real material constants but tuning parameters of DEM.

(2) Deformation of concrete

The model proposed by Katsuki and Haraki(2009) is applied to simulate deformation and failure of concrete in this study. **Fig.2** shows relation between force and displacement in this model.

Normal spring force is calculated by Popovics's formula as follows:

$$f_n = f_{n\max} \frac{q_n}{q_{n\max}} \frac{N}{N-1+\left|\dfrac{q_n}{q_{n\max}}\right|^N} \tag{5}$$

$$f_{n\max} = \begin{cases} f_c & (q_n \geq 0) \\ f_t & (q_n < 0) \end{cases} \tag{6}$$

$$q_{n\max} = \begin{cases} q_c & (q_n \geq 0) \\ q_t & (q_n < 0) \end{cases} \tag{7}$$

where f_n and q_n are normal force and displacement of spring respectively, $f_{n\max}$ and $q_{n\max}$ are normal strength and maximum displacement of spring respectively, f_c and f_t are compressive and tensile force corresponding to strength of concrete respectively, q_c and q_t are compressive and tensile displacement corresponding to maximum strain of concrete respectively and N is a parameter for non-linearity.

Shear spring force is calculated with plastic slide by Mohr-Coulomb failure criterion as follows:

$$f_s = \min(k_s q_s, C + f_n \tan\phi) \tag{8}$$

$$C = \alpha C_0 \tag{9}$$

$$\tan\phi = \alpha(\tan\phi)_0 \tag{10}$$

$$\alpha = \begin{cases} 1 & (q_s \leq q_{sy}) \\ 1 - \dfrac{(1-\alpha_y)\, q_{sp}}{q_{sf} - q_{sy}} & (q_{sy} < q_s < q_{sf}) \\ \alpha_y & (q_s \geq q_{sf}) \end{cases} \tag{11}$$

$$q_{sp} = q_s - q_{sy} \tag{12}$$

$$q_{sf} = r\gamma_{sf} \tag{13}$$

where f_s and q_s are shear force and displacement of spring respectively, C is cohesion, $\tan\phi$ is coefficient of friction (ϕ is friction angle), α is reducing coefficient, C_0 is initial cohesion, $(\tan\phi)_0$ is initial coefficient of friction, q_{sy} is shear displacement when shear force reaches the Mohr-Coulomb failure criterion, q_{sf} is shear displacement corresponding to maximum shear strain of concrete, q_{sp} is shear plastic displacement, r is radius of particle element, α_f is the minimum value for reducing coefficient and γ_{sy} is deformation angle after softening. This model describes that a critical value of concrete is reduced by plastic strain.

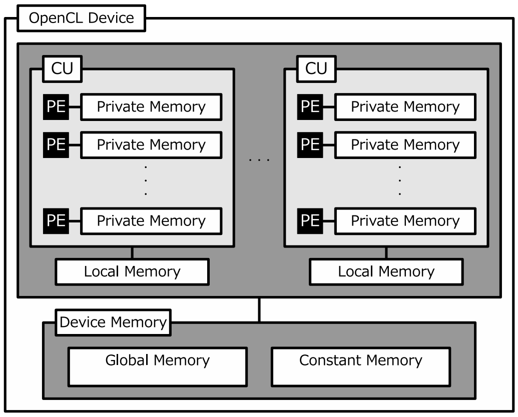


Fig. 3 OpenCL Device

3. GPGPU WITH OPENCL

(1) OpenCL

OpenCL (Open Compute Language), which is maintained by Khronos Group, is an open and standard framework for parallel computing using any processors including CPU, GPU, DSP and Cell. And it does not depend on platform. OpenCL ver.1.1, which is the latest version including C++ bindings, is used in this study.

Table 1 Computing environment

OS	Microsoft Windows XP Professional SP3
CPU	Intel Xenon E5420
RAM	3.00[GB]
GPU	NVIDIA GeForce GTX 295
GPU driver	ver. 6.14.0012.9610

Each processor (e.g. CPU or GPU) is treated as one of the OpenCL Devices. The OpenCL Platform is composed of OpenCL Devices and a host processor operating them. **Fig.3** shows a composition of OpenCL Device. Arithmetic processing is executed parallelly by Processing Element (PE) in Computation Unit (CU). There are 4 kinds of memories in OpenCL. Each PE has Private Memory, which can be accessed only by the owner PE. Local Memory is shared with all the PEs belonging to same CU. Global Memory is shared with any PEs and CUs belonging to same Device. Constant parameters are stored in Constant Memory.

Programs for OpenCL are categorized into two groups; Host Program and Kernel Program. Host Program is executed on a host hardware (usually a single core of CPU) and Kernel Program on OpenCL Devices (usually CPU cores other than host or a hardware other than CPU). Firstly, Host Program creates an executing environment called Context based on information provided from Platform. Context is composed of following objects:

- Command-Que, which operates devices
- Memory object, which handles data on Device
- Kernel object, which means one process on Kernel Program
- Program object, which is composed of more than one Kernel object

Kernel Program is written in OpenCL C language, which is based on C99. Host Program operates Kernel Program with OpenCL APIs.

(2) Acceleration rate of computing

Computing environment in this study is shown in **Table 1**. Free-fall to floor by the same 80t block as one used in Chapter 5 is simulated to measure computing time. Acceleration rate (ratio of computing time of CPU to GPU) in 4 cases (number of blocks $N = 1, 8, 34, 64$) are shown in **Fig.4**. Blocks arranged straight are fallen down to floor at the same time. Single block is composed by around 700 number of particles.

Each calculation process shows different acceleration rate. The more number of particles becomes, the more efficiently acceleration rate is found. ”ForceParticles” (which means calculation

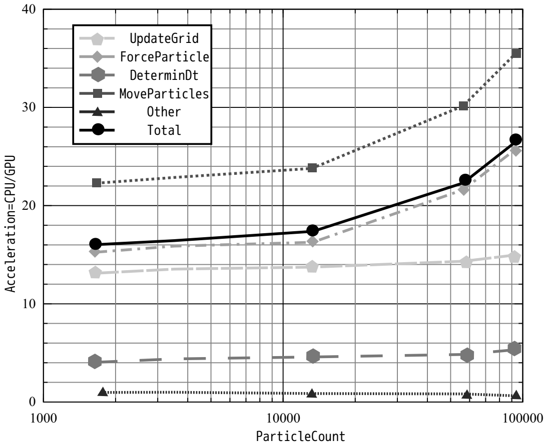
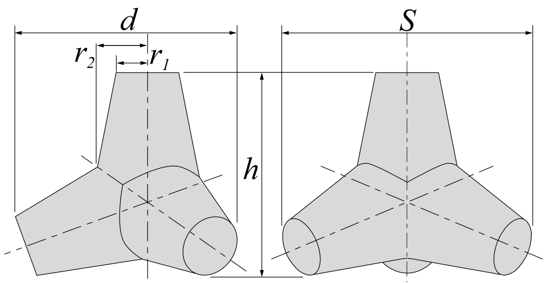


Fig. 4 Acceleration rate



Type	d	S	h	$2r_1$	$2r_2$
0.5t	967	1075	900	270	540
5t	2074	2306	1930	580	1160
80t	5375	5975	5000	1500	3000

Fig. 5 Overview of wave dissipating blocks

of inter-particle forces) and ”MoveParticles” (which means update of positions and velocities of particles) increase remarkably in comparison to ”UpdateGrid” (which means to store particle index to computational cell on the particle location) and ”DetermineDt” (which means determination of time increment Δt to satisfy CFL condition). This is because the former processes depend on the number of particles more strongly than the latter processes. The maximum accelerations is about 35 times on ”UpdateGrid” in case of about 1 million particles. However, accelerations of total computing time is to about 25 times in 1 million order particles case because ”ForceParticle” process occupies most of total computation time. Therefore, it is most important to accelerate this process.

Double-precision floating point calculation, computation speed of which is expected slower than single-precision floating point, is applied in this study to simulate deformation of concrete, allowable

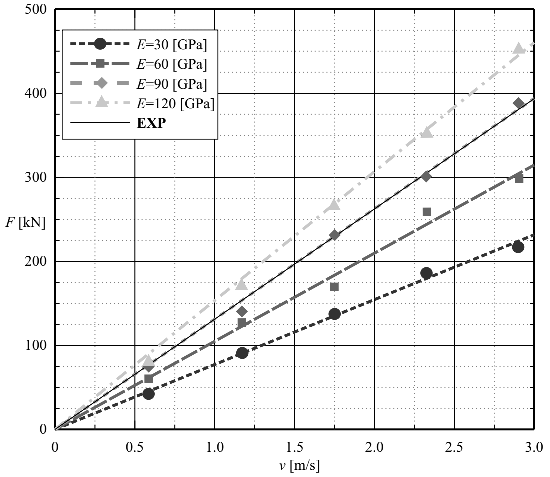


Fig. 6 Collisional velocity v and collisional force F

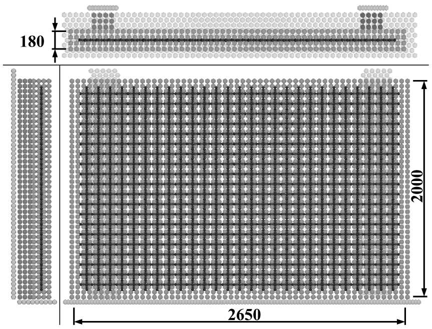


Fig. 7 Concrete plate$[\mathrm{mm}]$

displacement of which is very small. This is the reason why acceleration rate is not so higher in comparison to result by Harada et al. (2007) although 3 generation newer GPU than their one is used in this study.

4. COLLISION OF SINGLE BLOCK

(1) Collisional force to rigid wall

To determine Young's modulus in Hertzian elastic collision theory, a wave dissipating block (0.5t type shown in **Fig.5**) colliding with a rigid wall is simulated. Collisional velocity and force are shown in **Fig.6**. Calculated results in case of $E = 90\,[\mathrm{GPa}]$ show good agreements with experimental result (approximate lines by least-squares method) by Arikawa et al. (2005). Therefore, this value is adapted in this study.

(2) Failure of concrete plate

To examine applicability of the method proposed by Katsuki and Haraki (2009), a 5t type block colliding with a concrete plate is simulated. The concrete plate is shows in **Fig.7**. This is almost same as experimental one by Arikawa et al. (2005), however, reinforcing steel bars are located on only the center of plate and treated as a rigid body to accelerate computation speed.

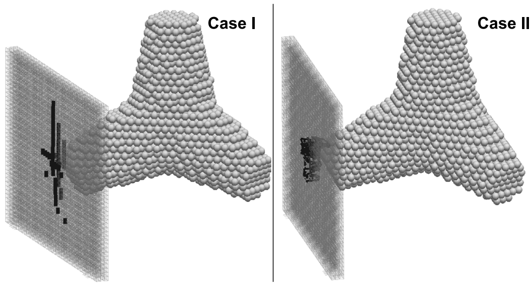


Fig. 8 Horizontal collision and failure (black line)
Case I: $v = 1.2\,[\mathrm{m/s}]$, Case II: $v = 3.0\,[\mathrm{m/s}]$

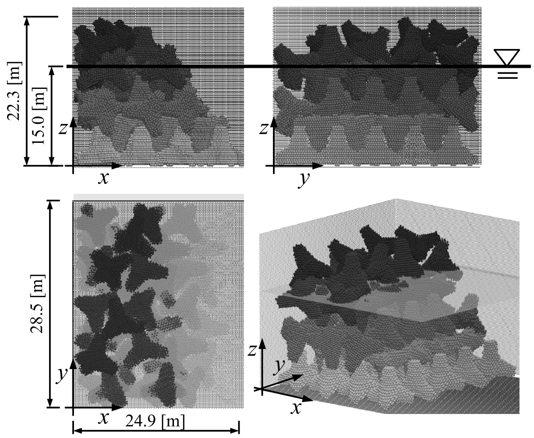


Fig. 9 A breakwater after packing computation

Fig.8 shows results when a block collides with a concrete plate horizontally. In lower speed case, vertical cracks are found on the back face of the concrete plate. This means the occurrence of bending failure. On the other hand, in higher speed case, there are many failures intensively around the collision point. In this case, punching shear failure occurs. These results are in good agreements with experimental ones by Arikawa et al. (2005).

5. ANALYSIS ON BREAKWATER UNDER HIGH WAVES

Fig.9 shows an overview of the breakwater covered with 80t type wave dissipating blocks after packing computation. Wave force is given to blocks as an external force changing periodically ($1.0\,[\mathrm{s}]$ loading and $4.0\,[\mathrm{s}]$ unloading, wave height H is

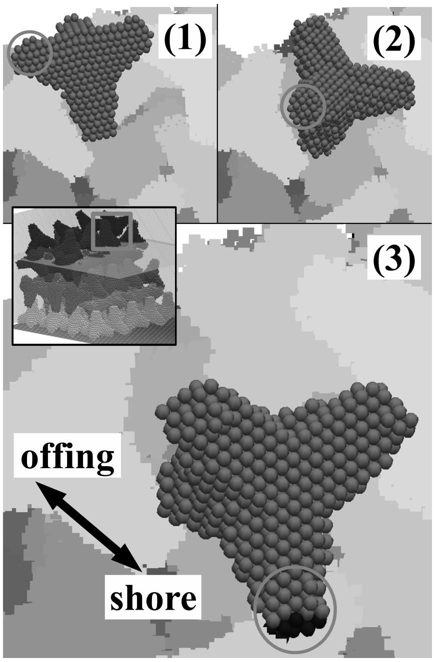


Fig. 10 Failure of a tip of block
(black-colored particle means occurrence of failure)

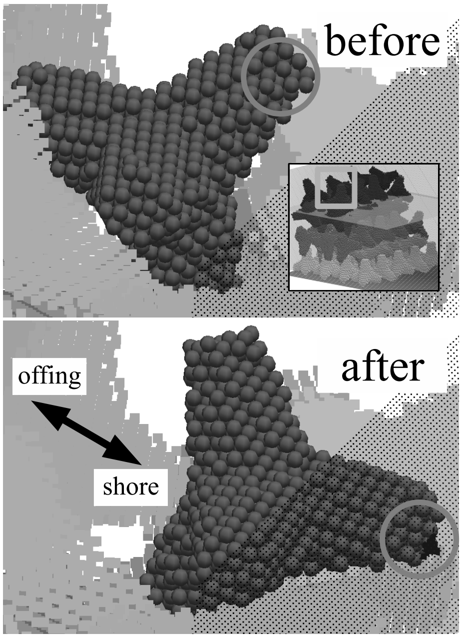


Fig. 11 Rocking motion and failure of caisson wall
(wall is shown as the dotted plane, block-colored one means failure)

$10.0\,[\mathrm{m}]$ and water depth in front of breakwater h is $15.0\,[\mathrm{m}]$). Wave force profile in vertical direction is based on the Goda's formula. These conditions are the same as Gotoh et al. (2005). Width and height of the breakwater are also almost the same, however, the location of each block is different because of dependency on initial condition before packing. Therefore, 5 cases, in which initial position of each block is changed randomly in a scale as half length of leg of block, are executed.

Fig.10 shows a block on the top layer drops down and collides with another block. **Fig.11** shows a block rocks and collides with caisson wall.

6. CONCLUDING REMARKS

In this study, DEM was accelerated by GPGPU for deformation and failure of concrete. The maximum acceleration rate reached 25 times. Results of collision of a block showed the same failure type as experimental one. In a simulation on a breakwater covered with wave dissipating blocks under high waves, failures which caused by dropping down and rocking of block were found.

In the present model, it is possible to execute stable calculation continuously even if a block splits into more than two bodies, however, any splits did not occur in this study. This would be because complicated flow in void of blocks was not treated.

In a future, effectiveness and advantage of the present model would be shown by coupling with a fluid dynamics model.

REFERENCES

Cundall, P.A. and Strack, O.D.L (1979): A discrete numerical model for granular assemblies, *Geothechnique*, Vol.29, No.1, pp.47-65.

Koshizuka, S., Nobe, A. and Oka, Y. (1998): Numerical analysis of breaking waves using the moving particle semi-implicit method *International Journal for Numerical Methods in Fluids*, Vol.26.

Gotoh, H., Harada, E., Takayama, T., Mizutani, M., Fudou, M. and Iwamoto, T. (2005): Compaction mechanism of wave dissipating blocks due to high waves, *Annual Journal of Coastal Engineering JSCE*, Vol.52, pp.781-785. (in Japanese)

Katsuki, S. and Haraki, D., (2009): Simulation of fragment scattering of concrete specimen subjected to high speed loading by Distinct Element Method, *AIP Conference Proceedings*, Vol.1145, Issue.1, pp.751-754.

Harada, T., Tanaka, M., Koshizuka, S. and Kawaguchi, Y. (2007): Acceleration of rigid body simulation using graphics hardware, *Proceedings of the 2007 symposium in Interactive 3D Graphics and Games*.

Arikawa, T, Ikebe, M., Osaki, N., Kuroda, T., Orita, T. and Simosako, K. (2005): Study on punching shear failures of caisson wall caused by the impulsive force of dissipating blocks *Report of the Port and Airport Research Institute*, Vol.44, No.1, pp.43-83. (in Japanese)

(Received June 15, 2012)