3D NUMERICAL WAVE FLUME WITH INTERACTIVE PRE- AND POST- PROCESSORS

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

[1]Dept. of Civil and Earth Resources Eng., Kyoto Univ., gotoh@particle.kuciv.kyoto-u.ac.jp
[2]Dept. of Civil and Earth Resources Eng., Kyoto Univ., ikari@particle.kuciv.kyoto-u.ac.jp
[3]School of Global Engineering, Kyoto Univ., naoki.yoshifuji@kw4.ecs.kyoto-u.ac.jp

Geometry of coastal structures is characterized by a three-dimensional and complicated configurations like a wave-dissipating block or a slit type seawall. In an analysis treating this kind of geometries, skill is required to set the boundary condition (pre-processing). In order to establish the numerical wave flume as a practical tool of structure design, easy pre-processing is indispensable. In this point of view, an interactive pre-processor which helps visualization of detailed configurations of structures is essential. Therefore, in this study, the development of a pre-processor for three-dimensional numerical wave flume based on a particle method is carried out. In addition, as for the image processing of calculated result, it is crucial to visualize behavior of fluid (post-processing). Hence, in this study, an effective way of pre- and post-processes for a three-dimensional numerical wave flume based on the particle method is proposed.

Key Words **:** *Particle method, Pre-processing, Post-processing, 3D numerical wave flume*

1. INTRODUCTION

A 3D numerical wave flume is useful for an investigation of detailed wave motion, however, there are some problems in an execution of simulation. One is a heavy computational load. In a 3D analysis, a computation inevitably becomes complex, because much number of calculating points are required for a discretization of differential operation than that in a 2D calculation. In these days, however, owing to development of parallel computing and application of GPGPU, this problem is being improved.

In a 3D analysis, a time-consuming setting of the boundary condition (pre-processing) is also one of the problems. Geometry of coastal structures is characterized by three-dimensional and complicated configurations like a wave-dissipating block or a slit type seawall. In an analysis treating this kind of geometries, skill is required. In addition, as for the image processing of calculated result, especially in a solver of Navier-Stokes equation, it is crucial to visualize behavior of fluid (post-processing).

In this study, the MPS method (Koshizuka and Oka., 1996), which is categorized as a particle method, is treated as a computational engine for a 3D numerical wave flume. The particle method shows excellent performance for an analysis of a violent flow. In the particle method, reduction of computational time has been examined by installing the parallel computing (e.g. Ikari and Gotoh, 2008) and GPGPU (e.g. Gotoh et al., 2010), however, to be established as a practical tool, the problem of pre-processing must be resolved. Therefore, in this study, an effective way of pre- and post-processes for a three-dimensional numerical wave flume based on the particle method is proposed.

2. MPS METHOD

The equation of motion is as follows:

$$\rho\frac{D\boldsymbol{u}}{Dt} = -\nabla p + \mu\nabla^2\boldsymbol{u} + \rho\boldsymbol{g} \qquad (1)$$

where $\boldsymbol{u}$ is the velocity vector, p is the pressure, ρ is the density, $\boldsymbol{g}$ is the gravitational acceleration vector, μ is the viscosity.

In the MPS method, the pressure term and the viscous term are discretized by a particle

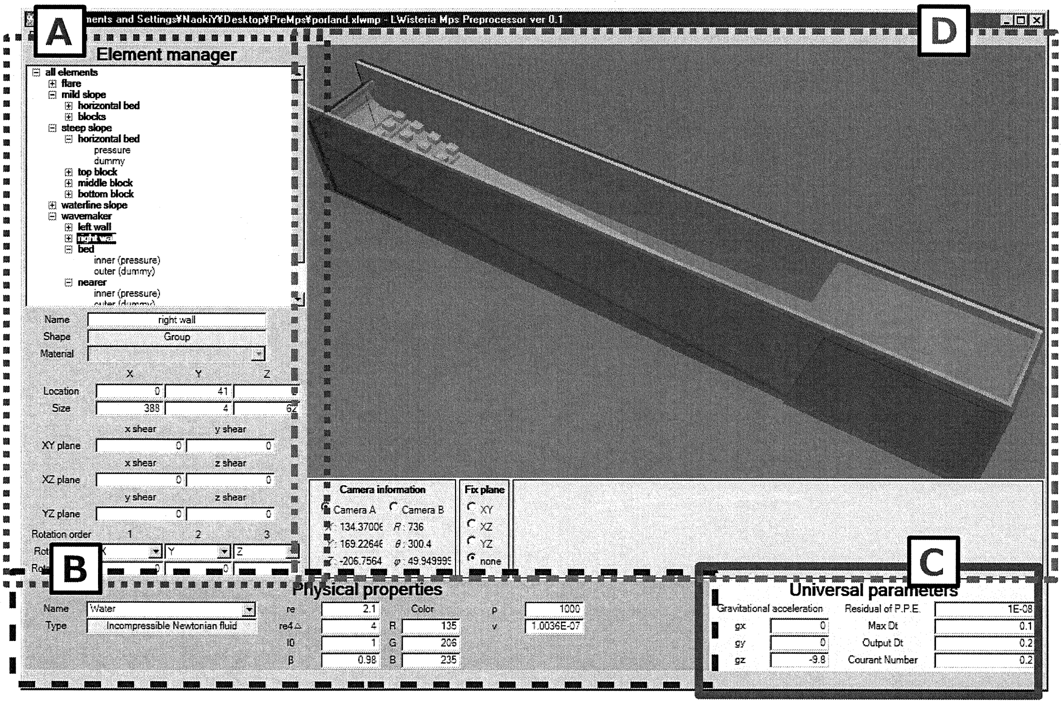

Fig.1 Main screen of pre-processor

interaction model between neighboring particles of the target particle i, as follows:

$$\langle \nabla p \rangle_i = \frac{D_0}{n_0}\sum_{j\neq i}\left\{\frac{p_j-p_i}{|\boldsymbol{r}_{ij}|^2}(\boldsymbol{r}_{ij})\, w(|\boldsymbol{r}_{ij}|)\right\} \tag{2}$$

$$\langle \nabla^2 \boldsymbol{u} \rangle_i = \frac{2D_0}{n_0\lambda}\sum_{j\neq i}(\boldsymbol{u}_j-\boldsymbol{u}_i)\, w(|\boldsymbol{r}_{ij}|) \tag{3}$$

in which D_0 is number of dimensions, n_0 is the constant particle number density, $\boldsymbol{r}_{ij}$ is the relative position vector between two particles i and j, λ is the model constant and $w(r)$ is the weight function based on the distance r defined in influencial sphere for particle interaction.

2. PRE-PROCESSOR

(1) Outline of pre-processor

The pre-processor for MPS calculation, which was developed by Gotoh et al. (2004), has an interactive user interface for helping an operator in arranging particles, however, it can handle only a 2D calculation.

The pre-processor for 3D calculation developed in this study has also interactive user interface. The pre-process of the particle method simulation can be divided in two stages. One is the generation of the model of wave flume, and the other is arrangement of particles. In **Fig. 1**, the main screen of the present pre-processor is shown. In this pre-processor, shape of model, physical properties and universal parameters can be set as mentioned in detail later. In the area D in **Fig. 1**, the model under generation is displayed, in which a view point can be moved by a mouse operation.

(2) Parameter setting

a) Element manager (area A)

In the area A in **Fig. 1**, elements of the model of wave flume is managed. The model is generated by combining basic polygons such as a quadrangular, triangular prism, cylinder and sphere. These polygons can be scaled up and down, sheared and rotated by inputing geometrical parameters. The elements of the model under generation are displayed in a tree structure as shown in the area A in **Fig. 1**. Elements can be treated and deformed as a group. Because group of elements can be saved and loaded, groups generated previously can be reused. This function was not included in the previous pre-processor by Gotoh et al.(2004), however, it is very useful because a group can be

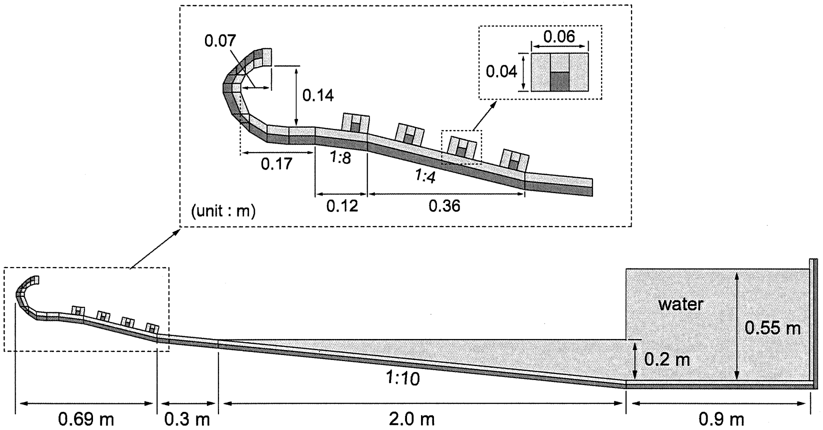

Fig.2 Calculated domain

treated as a template.

b) Physical properties (area B)

In the area B in **Fig. 1**, physical properties of an element can be defined. Basically, density, viscosity and particle diameter can be set, however, it is possible to add other parameters (e.g. friction coefficient or Lame constants for elastic body). The color of an element displayed in area D in **Fig. 1** can also be changed by designating its RGB value.

c) Universal parameters (area C)

In the area C in **Fig. 1**, universal parameters (e.g. the gravitational acceleration, time increment and convergence criterion of ICCG method for Poisson pressure equations) can be defined.

3. POST-PROCESSING

The post-processing in this study is based on the procedure proposed by Harada et al. (2007). In this post-processing, a construction of a surface of object is divided into two stages.

At first, a concentration profile around each particle is defined, and implicit surfaces are constructed by summing up concentration values. The concentration value $\phi(\boldsymbol{x})$ at the position $\boldsymbol{x}$ is as follows:

$$\phi(\boldsymbol{x}) = \sum_j w_{pp}(\boldsymbol{x} - \boldsymbol{r}_j) \tag{4}$$

where $\boldsymbol{r}_j$ is the position of particle j and w_{pp} is the weight function. The weight funcion in this study is as follows:

$$w_{pp}(\boldsymbol{x}) = \begin{cases} \dfrac{r_e}{|\boldsymbol{x}|} - 1 & for \quad 0 \le |\boldsymbol{x}| < r_e \\ 0 & for \quad r_e \le |\boldsymbol{x}| \end{cases} \tag{5}$$

where r_e is the radius of influence sphere. This weight function is the same as the one used generally in the MPS method. The intersection points between computational grid and implicit surfaces defined at the appropriate threshold value of concentration are picked up. And surface mesh whose vertexes are the intersection points are formed by using the Marching Cube method. These vertexes are named as $V^0=\{\boldsymbol{v}_0^{\,0}, \boldsymbol{v}_1^{\,0}, \boldsymbol{v}_2^{\,0}, \cdots, \boldsymbol{v}_n^{\,0}\}$.

In the second stage, another point cloud $T^k=\{\boldsymbol{t}_0^{\,k}, \boldsymbol{t}_1^{\,k}, \boldsymbol{t}_2^{\,k}, \cdots, \boldsymbol{t}_n^{\,k}\}$ is defined by using V^k as follows:

$$\boldsymbol{t}_i^k = \frac{\sum_j w_{pp}(\boldsymbol{v}_i^t - \boldsymbol{r}_j)\boldsymbol{r}_j}{\sum_j w_{pp}(\boldsymbol{v}_i^t - \boldsymbol{r}_j)} \tag{6}$$

Then, positions of point cloud V^k are revised as follows:

$$\boldsymbol{v}_i^{k+1} = (1-c)\boldsymbol{v}_i^k + c\boldsymbol{t}_i^k \tag{7}$$

in which c is a tuning parameter for sharpness of surface. By iteration of Eqs (6) and (7), the shape of the surface can be close to arrangement of particles.

4. WAVE RUN-UP SIMULATION

(1) Calculated condition

In **Fig. 2**, the calculated domain is shown. The flaring shaped seawall is arranged at the left end in the calculated domain. A wave generated by dam-break at the right end in the calculated domain attacks this seawall. Fixed rectangular blocks are arranged at equal intervals in front of the seawall. In this pre-processor, arrangements of blocks like

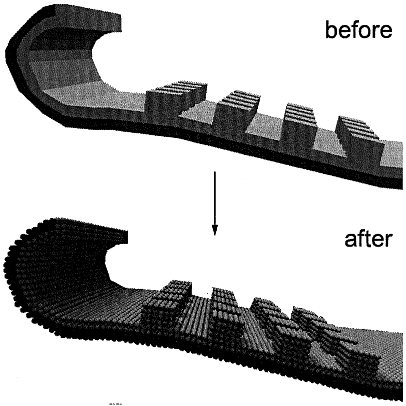

Fig.3 Conversion into particles

this is also easy by using a copy command of pre-processor.

Fig. 3 shows flaring shaped seawall and blocks converted into particles from the model by the pre-processor. The model is converted into particles arranged on square-lattice points basically, however, positions of particles are adjusted by deforming the model to keep the particle number density constant automatically.

(2) Calculated result and post-processing

Fig.4 shows snapshots of a calculated result by a simple particle plot. This post-processing way is suitable to show exact position of calculating points. In **Fig. 5**, mesh generation and texture mapping are done in fluid and fixed wall separately. Fluid is expressed with higher reality than **Fig. 4**, and it can be a help to imagine behavior of fluid, however, the edge of fixed wall is not clear. This drawback can be improved by rendering the model generated by the pre-processor (see **Fig. 6**). As shown in these three figures, it is possible to improve the quality of the ray-tracing CG effectively by combined operation of pre- and post-processing systems.

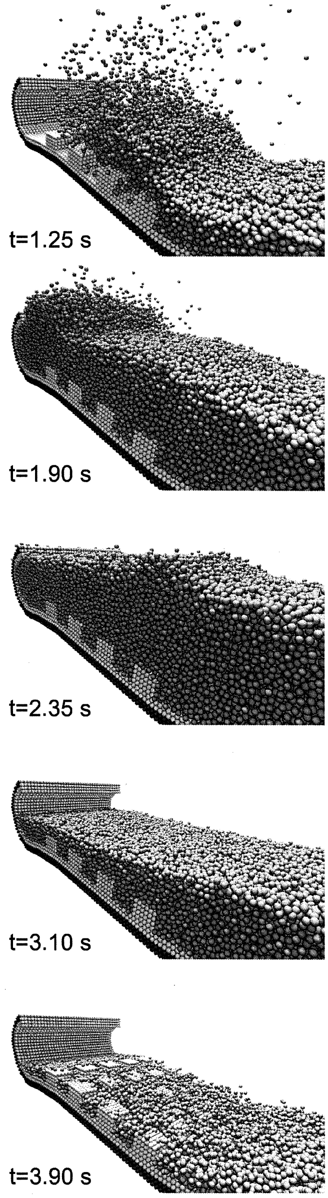

Fig.4 Post-processing by simple particle plot

5. CONCLUDING REMARKS

In this study, development of an interactive pre-processor for 3D numerical wave flume based on the MPS method was carried out. Generation of a complicated 3D structure can be handled by using the presented user-friendly pre-processor.

As for the post-processing, three types of post-processing way were compared. By mapping texture on implicit surfaces generated based on the position of particle, the expression of fluid can be improved, however, the edges of an object are not clear. In such a case, it is effective to render the model generated in pre-processing in post-processing again.

REFERENCES

Gotoh, H., Sumi, T. and Sakai, T. (2004): Preprocessor for human interface of numerical wave flume, *Proc. of the 6th ICHE*, electric paper.

Gotoh, H., Hori, C., Ikari, H. and Khayyer, A. (2010): Semi-implicit algorithm of particle method accelerated by

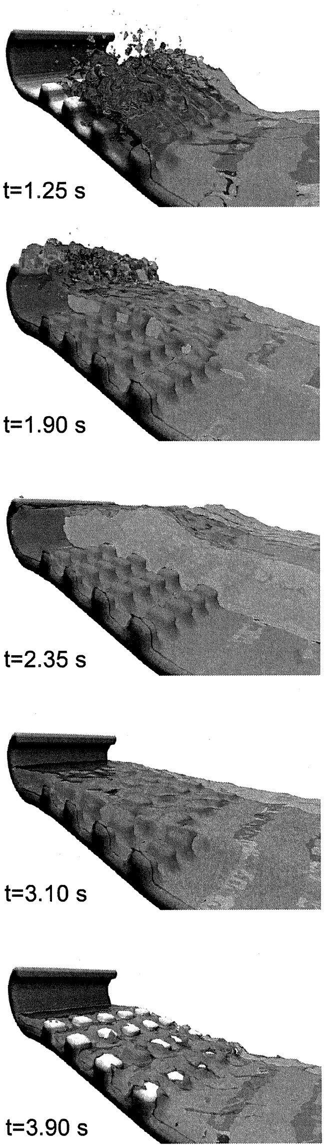

Fig.5 Post-processing by Marching Cube method

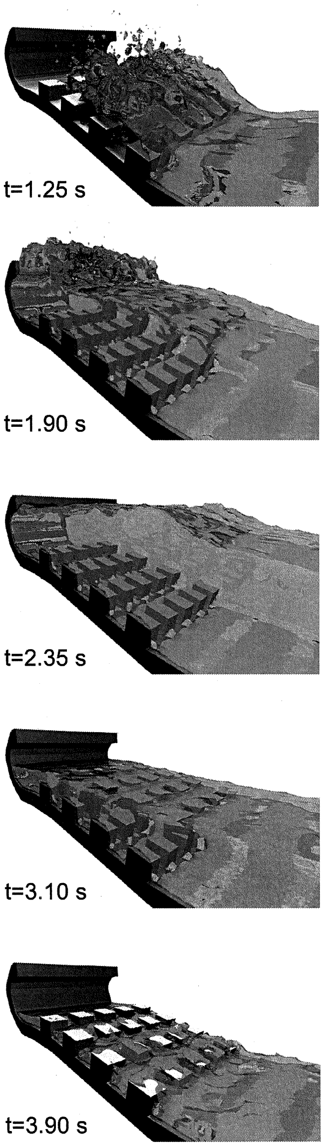

Fig.6 Post-processing by Marching Cube method with high resolution fixed boundary

GPU, *Journal of Hydraulic, Coastal and Environmental Engineering*, (in printing; in Japanese).

Harada, T., Koshizuka, S. and Kawaguchi, Y. (2007): An improvement of surface construction from particles, *IPSJ Journal*, Vol. 48, pp. 4059-4063 (in Japanese).

Ikari, H. and Gotoh, H. (2008): Parallelization of MPS method for 3-D wave analysis, *Proc. of the 8th ICHE2008*, electric paper.

Koshizuka, S. and Oka, Y. (1996): Moving-particle semi-implicit method for fragmentation of incompressible fluid, *Nucl. Sci. Eng.*, Vol. 123, pp. 421-434.

(Received March 19, 2010)