Proceedings of the Twenty-eighth (2018) International Ocean and Polar Engineering Conference **www.isope.org**
Sapporo, Japan, June 10-15, 2018

ISBN 978-1-880653-87-6; ISSN 1098-6189

Simulation of Dam-break Wave Impacting onto the Square Column by GPU Accelerated MPS Method

*Xin Tian, Decheng Wan**
Collaborative Innovation Center for Advanced Ship and Deep-Sea Exploration, State Key Laboratory of Ocean Engineering,
School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai, China
*Corresponding author

ABSTRACT

In the current study, a simulation of the interaction between the three-dimensional dam-break wave and the vertical square column is carried out by using the MPSGPU-SJTU solver. The simulation conditions are arranged according to the experiments performed by Yeh and Petroff (2006). The results of GPU solver are compared to other researches. The evolution procedure of three-dimensional dam-break wave, including the climb, fragmentation and rollover of free surface is presented in this paper. In the process of dam-break wave and vertical square column interaction, the net force exerted on the column is monitored and in good agreement with existing experimental data. A remarkable speedup is obtained by comparing the calculation time of the GPU solver with that of the CPU version. The effect of bottom water layer is investigated. The result shows a significant difference between flow phenomenon with and without water layer.

KEY WORDS: Moving particle semi-implicit (MPS); GPU acceleration; MPSGPU-SJTU solver; dam-break wave with square column

INTRODUCTION

The impact of waves on structures is an important problem in ship and ocean engineering, including nonlinear wave surface evolution, wave climbing and slapping on structures, and severe deformation or even fragmentation of free surface under the effect of structures. In recent years, the mesh-free method MPS has gained popularity for modeling free surface flows, and it has become an alternative to traditional mesh-based methods for modeling waves. Owing to the Lagrangian nature of the mesh-free method, there is no need to deal with the free surface when it is applied to simulate nonlinear free surface flows, especially when the surface tension is not important. This property makes it particularly attractive to modeling water waves, e.g., dam-break (Zhang et al., 2011), sloshing (Yang et al., 2015), water entry (Chen et al., 2017).

The earlier MPS method was limited to the two-dimensional flow problem. This is because of the large amount of calculation of MPS method, the calculation of three-dimensional problem requires a large number of particles. In order to improve the efficiency of MPS method, researchers have two main ideas: one is the method of local encryption of particles, using fewer particles to obtain better simulation results, such as multi-resolution particle method (Tang et al., 2016), overlapping particle method (Shibata et al., 2012). Another kind of parallel algorithm is divided into two kinds from the hardware environment: one is the parallel method based on CPU environment (Ikari and Gotoh, 2008, Iribe et al., 2010), the other is the parallel method based on GPU. Zhu et al. (2011) developed different versions of MPS code based on different GPU memories. Hori et al. (2011) used CUDA (Compute Unified Device Architecture) language to develop a GPU-accelerated MPS code and only acquired about 3-7 acceleration ratio by simulating two-dimensional (2-D) dam break. Li et al. (2015) applied GPU acceleration technique to two parts of MPS, neighbor particle list and pressure Poisson equation. By simulating 3-D dam break and sloshing, the speedup of these two parts is about 1.5 and 10, respectively. Gou et al. (2016) used GPU accelerated MPS to simulate the isothermal multi-phase fuel-coolant interaction.

In this work, the GPU acceleration technique is applied to simulate 3-D free surface flows based on modified MPS. The brief introduction of modified MPS and GPU implementation in this paper is presented. Then the GPU solver is used to simulate 3-D dam-break wave problems. The numerical results of GPU code such as fluid field, and net force acting on the column are compared to the results of CPU solver, experiment and other methods. In addition, the comparison of computation time between GPU solver and CPU solver is conducted.

NUMERICAL METHODS

MPS method is a Lagrange method, its basic idea is to discrete the continuous flow field region into a series of particles with properties such as mass, momentum and energy, through the integration of kernel function to realize the interaction between particles. The entire flow problem can be simulated by calculating the force on the particles and tracking the movement of the particles. In the MPS method, the attribute value of the target particle i in the flow field is equal to the

weighted average of the surrounding particle attribute values, as follows:

$$\langle f_i \rangle = \frac{\sum_{j \neq i} f_j W\left(\left|r_i - r_j\right|\right)}{\sum_{j \neq i} W\left(\left|r_i - r_j\right|\right)} \tag{1}$$

Where $\langle \ \rangle$ represents particle integration, r_i and r_j are the coordinates of target particle i, neighbor particle j, f_i and f_j are the attribute values at target particle i and neighbor particle j, respectively. And $W(|r_j - r_i|)$ is the weight function.

Based on the above interpolation ideas of MPS method, we can get the mathematical models for discrete control equations. These models are generally referred to as particle interaction models, including gradient model, Laplacian model and divergence model. This section describes the characteristics of the various models, the boundary treatment problem in MPS method calculation, and the pressure Poisson equation of MPS method.

Governing Equations

In the MPS method, governing equations contain the mass and momentum conservation equations. They can read as:

$$\frac{1}{\rho}\frac{D\rho}{Dt} = -\nabla \cdot \boldsymbol{V} = 0 \tag{2}$$

$$\frac{D\boldsymbol{V}}{Dt} = -\frac{1}{\rho}\nabla P + \nu \nabla \boldsymbol{V} + \boldsymbol{g} \tag{3}$$

Where ρ denotes the density, P is the pressure, $\boldsymbol{V}$ is the velocity, g is the gravity acceleration and ν is the kinematics viscosity.

Particle Interaction Models

In meshfree particle method, governing equations are transformed to particle interaction equations. The interaction between particles is described through a kernel function. In this paper, we adopt the following kernel function (Zhang and Wan, 2011a):

$$W(r) = \begin{cases} \dfrac{r_e}{0.85r + 0.15r_e} - 1 & 0 \le r < r_e \\ 0 & r_e \le r \end{cases} \tag{4}$$

Where $r = |r_i - r_j|$ denotes the distance between two particles, r_ε is the supported radius of the influence area of each particle. The gradient model and the divergence model is $r_e = 2.1 l_0$, while $r_e = 4.01 l_0$ is used for the Laplacian model, where l_0 is the initial distance between two adjacent particles.

To calculate the weighted average in MPS method, particle number density is defined as (Koshizuka et al., 1998):

$$< n >_i = \sum_{j \neq i} W(|\boldsymbol{r}_j - \boldsymbol{r}_i|) \tag{5}$$

This value is assumed to be proportional to the density, so the particle number density can be applied instead of density in particle discretization.

Gradient Model

In this paper, the gradient operator can be discretized into a local weighted average of radial function as follows:

$$< \nabla P >_i = \frac{D}{n^0} \sum_{j \neq i} \frac{P_j + P_i}{|\boldsymbol{r}_j - \boldsymbol{r}_i|^2} (\boldsymbol{r}_j - \boldsymbol{r}_i) \cdot W(|\boldsymbol{r}_j - \boldsymbol{r}_i|) \tag{6}$$

Where D is the number of space dimension, $\boldsymbol{r}$ represents coordinate vector of fluid particle, $W(r)$ is the kernel function and n^0 denotes the initial particle number density for incompressible flow. Eq.5 can not only improve the stability of the calculations but also maintain the momentum conservation.

Laplacian Model

Laplacian operator is derived by Koshizuka et al. (1998) from the physical concept of diffusion as:

$$< \nabla^2 \phi >_i = \frac{2D}{n^0 \lambda} \sum_{j \neq i} (\phi_j - \phi_i) \cdot W(|\boldsymbol{r}_j - \boldsymbol{r}_i|) \tag{7}$$

The parameter λ is introduced as:

$$\lambda = \frac{\sum_{j \neq i} W(|\boldsymbol{r}_j - \boldsymbol{r}_i|) \cdot |\boldsymbol{r}_j - \boldsymbol{r}_i|^2}{\sum_{j \neq i} W(|\boldsymbol{r}_j - \boldsymbol{r}_i|)} \tag{8}$$

In Eq.6, the parameter λ is introduced to keep the increase of variance equal to that of the analytical solution.

Model of Incompressibility

Here we adopt a mixed source term for PPE proposed by Tanaka and Masunaga (2010), which combines the velocity divergence and the particle number density. The PPE equation is modified to the following form by Lee et al. (2011):

$$< \nabla^2 P^{n+1} >_i = (1 - \gamma) \frac{\rho}{\Delta t} \nabla \cdot V_i^* - \gamma \frac{\rho}{\Delta t^2} \frac{< n^* >_i - n^0}{n^0} \tag{9}$$

where γ is a parameter to account for the relative contributions of the two terms with the range of $0 \le \gamma \le 1$.

Free Surface Boundary Condition

To improve the accuracy of surface particle detection, Zhang and Wan employed a new detection method in which a vector function is defined as follow (Zhang and Wan, 2011b):

$$< \boldsymbol{F} >_i = \frac{D}{n^0} \sum_{j \neq i} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} (\boldsymbol{r}_i - \boldsymbol{r}_j) W(\boldsymbol{r}_{ij}) \tag{10}$$

The vector function $\boldsymbol{F}$ represents the asymmetric arrangements of neighbor particles and has a large value at the free surface.

Thus, particles satisfying:

$$<|\boldsymbol{F}|>_i \ > \alpha \tag{11}$$

are considered as surface particle, where α is a parameter with a value of $0.9|F|^0$ in this paper, $|F|^0$ is the initial value of $|F|$ for surface particle.

Wall Boundary Condition

In this paper, there are two different boundary particles at the wall. A layer of wall particles is arranged on the wall surface to participate in the solution of PPE together with the fluid particles. Two layers of ghost particles are arranged outside the wall, and the pressure of such boundary particles is obtained by extrapolation.

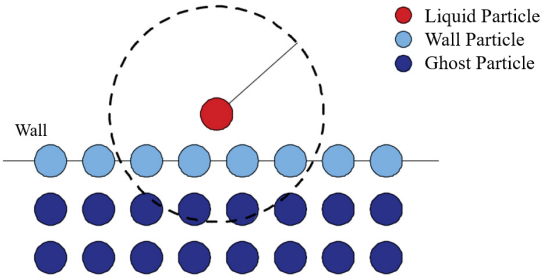

Fig. 1 Schematic of boundary particles

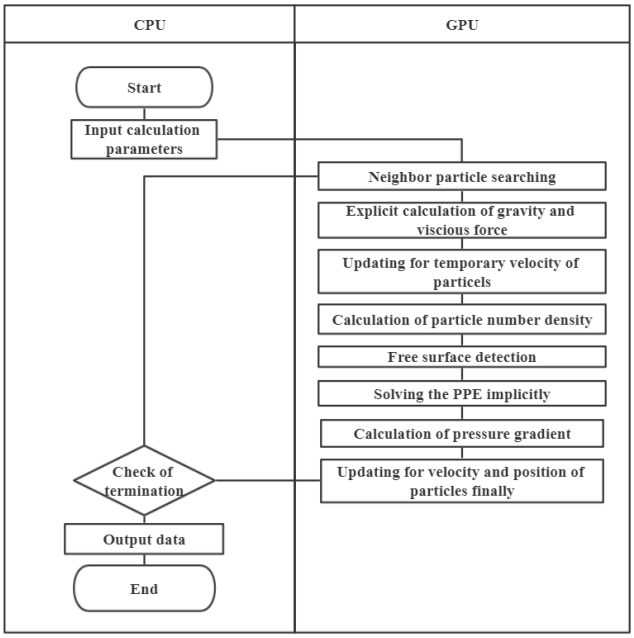

Fig. 2 The flow chart of GPU implementation

GPU Acceleration

In this paper, we accelerate the solution process by GPU parallel acceleration technique. Parallel programs are written based on CUDA platform. Cusp, a library for sparse linear algebra and graph computations based on Thrust, is used in solving PPE equation. Calculation process is shown in Fig. 2.

GEOMETRY AND SIMULATION DETAILS

The model referenced by the simulation was performed by Yeh and Petroff (2006). A rectangular tank is 0.61 m wide, 1.6 m long and 0.75 m high with a 0.12 m square column located 0.9 m from one end of the tank. A volume of water (0.61 m wide, 0.4 m long and 0.3 m high) is initially contained behind a gate before being released. During the experiment a thin layer of water (10 mm deep) was also present in the tank downstream of the gate.

Our numerical model is shown in Fig. 4, water density ρ=1000 kg/m^3, kinematic viscosity coefficient $\nu = 10^{-6}\ m^2/s$, gravitational acceleration $g = 9.81\ m/s^2$. In all simulations the column is modelled as an infinitely stiff rigid structure. The particle spacing was 0.006 m. The whole flow process lasted 3 s, and the time step is 10^{-4} s. In the simulations with water layer, the total number of particles used was763950, of which the number of fluid particles was 345960. In the simulation without water layer, the corresponding numbers are 744690, 326700, respectively.

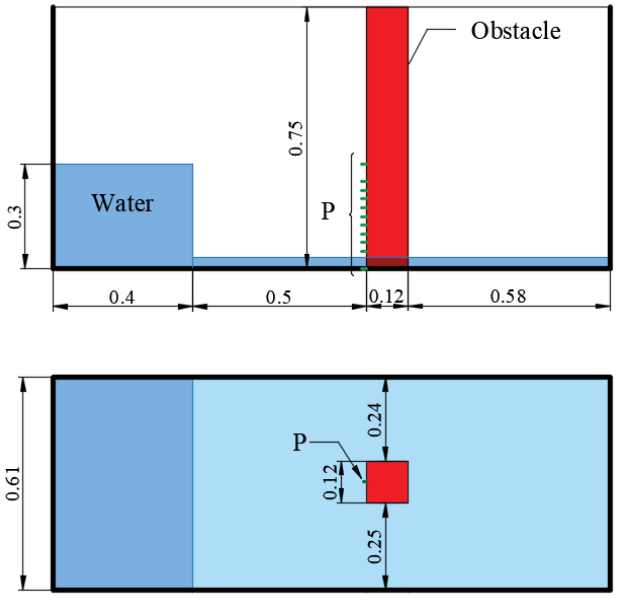

Fig. 3 Experimental setup: side view and top view

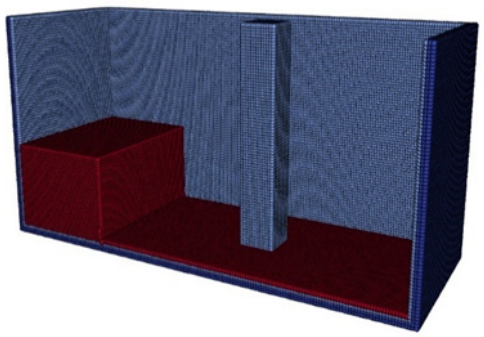

Fig. 4 Numerical model

Simulations are performed on parallel high performance computing (HPC) with multi cores of Intel(R) Xeon(R) E5-2680 v2, 2.80 GHz. The GPU card is NVIDIA Tesla K40M, which has 2880 CUDA cores with 12GB graphics memory. Table 1 shows the parameters of computing devices. All data are saved by double precision floating point in both CPU and GPU solvers. Ten-core parallelism is used in CPU computing.

Table 1. Computational environment of CPU and GPU

	CPU	GPU
Device	Intel(R) Xeon(R) E5-2680v2, 2.80Ghz	Telsa K40M
Memory	DDR3 1600,16 GB	12 GB
Max Cores	10	2880
Programming Language	C++	CUDA C/C++
Compiler	gcc, MVAPICH	CUDA 7.0, Cusp v0.5.1

RESULTS AND DISCUSSIONS

Validation of GPU Solver

In this section, we use both GPU solver and CPU solver to calculate examples with the same parameters. The calculated results of GPU including fluid flied and net x-force acting on the column are compared to those of CPU solver, experiment and SPH simulation (Cummins et al., 2012). Some snapshots of numerical flow fields are shown in Fig. 5, and it can be seen that the flow patterns of GPU simulation in this study are highly consistent with the fluid flied of CPU code and SPH. In Fig. 6, the net x-force acting on column of CPU and GPU codes basically coincide with the experimental data. The peak value is closer to the experimental value than SPH calculation.

After releasing, the water flows forward under the action of gravity to form dam-break waves. During the forward process of the wave, the static water layer of the bilge is pushed, and the phenomenon of roll-over occurs at the leading edge. When the wave touches the square column at about 0.35 s, the resultant force of the square column increases sharply, reaching the peak value of about 43 N, and then decreases sharply to the half. As the water flows around the square column to the right-side wall, the resultant force becomes smaller steadily. After the water slams the right side wall of the tank, the reflected wave impacts the square column at about 1.2 s, and the resultant force of the square column is negative. After that, the flow slows down gradually, and the horizontal resultant force acting on the square column stabilizes at about 0 N.

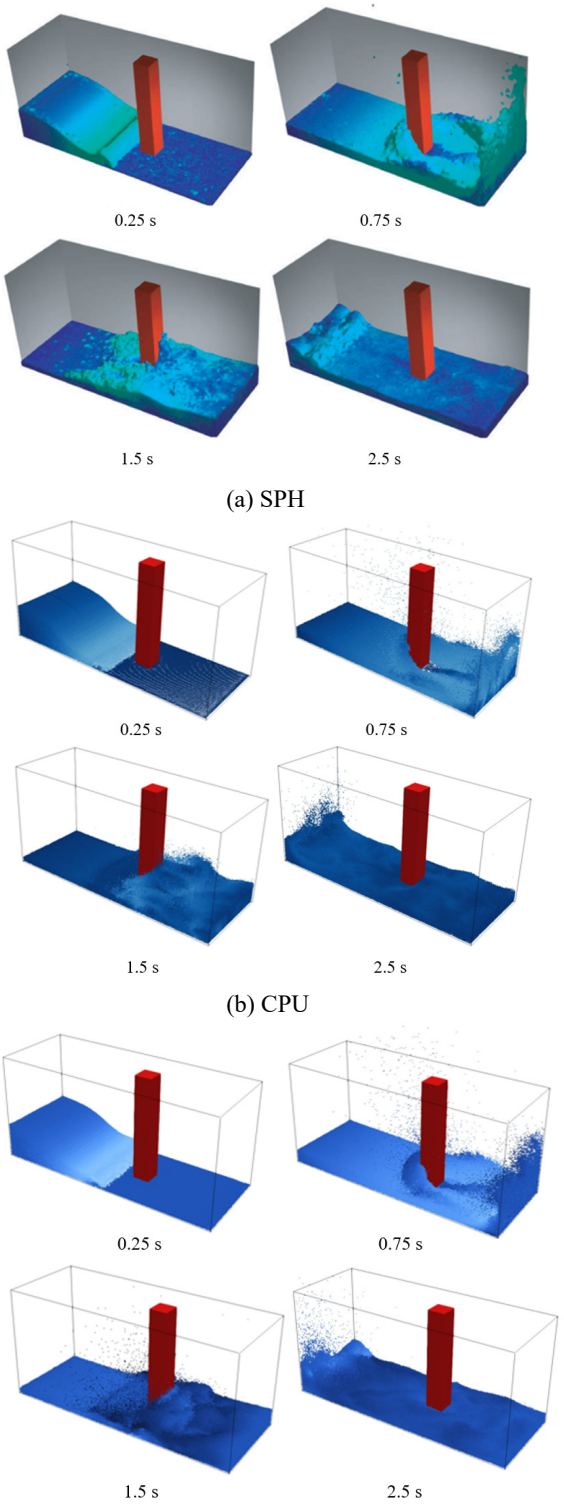

Fig. 5 The flow fields of SPH, CPU and GPU simulations

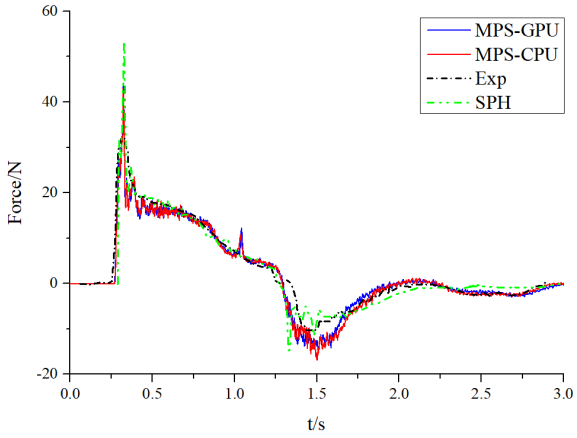

Fig. 6 The time history of net x-force acting on column

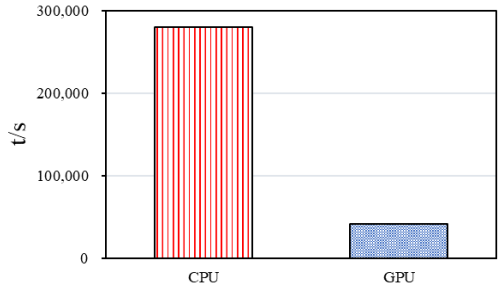

Fig. 7 The computation times of GPU and CPU

Fig. 7 shows a comparison of the calculation time between CPU and GPU codes. The calculation total time of CPU version is 280141 s while that of GPU version is 41223 s. By applying GPU acceleration, the computational efficiency is improved by about 6 times, and the acceleration effect is remarkable.

Effect of Water Layer

Considering that the bottom water layer is thin and the number of particles that can be set is small, the simulation accuracy will be affected to a certain extent. Therefore, the effect of the water layer on the simulation is considered. We arranged a series of pressure measuring points along the height direction on the left side of the square column. The pressure measurement point arrangement is shown in Fig. 3, marked as P points, with different heights of 0 m, 0.05 m, 0.075 m, 0.1 m, 0.125 m, 0.15 m, 0.175 m, 0.2 m, 0.225 m, 0.25 m, 0.3 m, 0.35 m, respectively.

Fig. 8 shows the comparison of flow fields among the cases with and without water layer. It can be seen that at 0.2 s, the front edge of dam-break wave has not touched the square column. In the case with water layer, the leading edge of dam-break wave pushes the static bottom water layer to form an obvious roll-over phenomenon, which is not found in the case without water layer. At 0.5 s, the water front with water layer just touches the right-side wall surface. However, the water front has impacted the wall surface and appeared obvious climbing and splashing phenomenon in the case without water layer. The flow velocity with water layers is significantly slower due to the obstruction of the bottom water layer. The climbing heights of the column in the two cases are also different. The height of the case without water layer is significantly higher than that of the case with water layer.

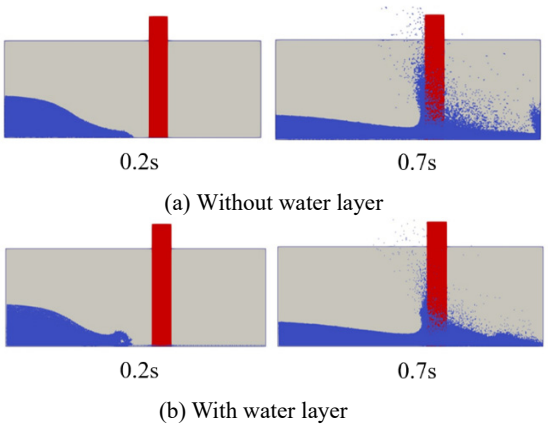

Fig. 8 Comparison between flow with and without water layer

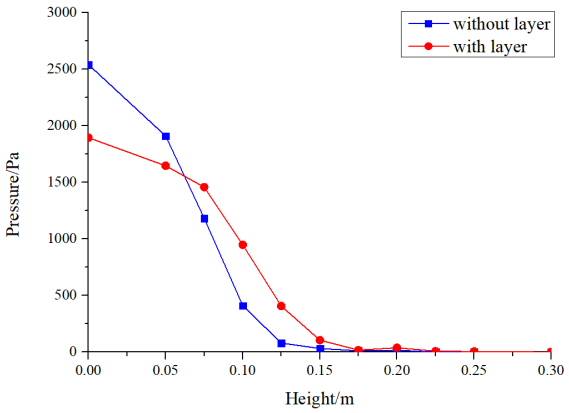

Fig. 9 Pressure distribution on left side of the square column

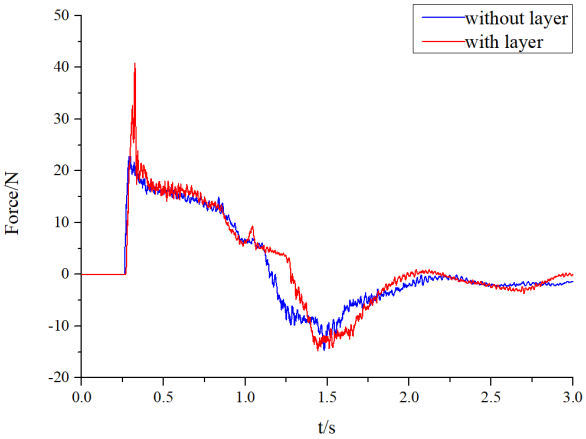

Fig. 10 Column net x-force time history

The pressure distribution on the column at 0.48 s is shown in Fig. 9, when the dam-break waves just touches the square column in both sets of tests. It can be seen that in the group with water layer, the pressure is concentrated at a higher position. This is due to the roll up waves caused by the water layer hit there. From the x-direction resultant force of the square column in Fig. 10, the peak value of the resultant force in

the group with water layer is higher and the development process is slower, which confirms our previous observation. The water layer has an obvious effect on the computational results.

CONCLUSIONS

In this paper, the MPSGPU-SJTU solver is used to simulate the interaction problem between dam-break wave and square column. Firstly, the phenomena of wave rolling and liquid splashing in the flow process can be well simulated. In addition, the results of the resultant force calculated by the solver are in good agreement with the experimental, SPH and CPU results. And the solver achieves a significant acceleration effect relative to the CPU version. Then, the influence of the water layer on the calculation results is studied comparatively. It is found that the water layer causes a slowdown on the flow phenomenon and increases the peak value of the impact pressure.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (51490675, 11432009, 51579145), Chang Jiang Scholars Program (T2014099), Shanghai Excellent Academic Leaders Program (17XD1402300), Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning (2013022), Innovative Special Project of Numerical Tank of Ministry of Industry and Information Technology of China (2016-23/09) and Lloyd's Register Foundation for doctoral student, to which the authors are most grateful.

REFERENCES

Chen, X, Rao, CP and Wan, DC (2017). "Numerical simulation of water entry for two-dimensional wedge by MPS," *Chinese Journal of Computational Mechanics*, 34, 356–362.

Cummins, SJ, Silvester, TB and Cleary, PW (2012). "Three-dimensional wave impact on a rigid structure using smoothed particle hydrodynamics," *International Journal for Numerical Methods in Fluids*, 68, 1471-1496.

Gou, W, Zhang, S and Zheng, Y (2016)."Simulation of isothermal multiphase fuel-coolant interaction using MPS method with GPU acceleration," *Kerntechnik*, 81, 330–336.

Hori, C, Gotoh, H, Ikari, H and Khayyer, A (2011). "GPU-acceleration for moving particle semi-implicit method," *Computers & Fluids*, 51, 174–183.

Ikari, H and Gotoh, H (2008). "Parallelization of MPS method for 3D wave analysis," *Advances in Hydro-science and Engineering, 8th International Conference on Hydro-science and Engineering (ICHE)*, Nagoya, Japan.

Iribe, T, Fujisawa, T and Koshizuka, S (2010). "Reduction of communication in parallel computing of particle method for flow simulation of seaside areas," *Coastal Engineering Journal*, 52, 287–304.

Lee, BH, Park, JC, Kim, MH and Hwang, HC (2011). "Step-by-step improvement of MPS method in simulating violent free-surface motions and impact-loads," *Computer methods in applied mechanics and engineering*, 200(9), 1113-1125.

Li, HZ, Zhang, YL and Wan, DC (2015). "GPU Based Acceleration of MPS for 3D Free Surface Flows," *Proceedings of the 9th International Workshop on Ship and Marine Hydrodynamics*, Glasgow, UK.

Shibata, K, Koshizuka, S, Sakai, M and Tanizawa, K (2012). "Lagrangian simulations of ship-wave interactions in rough seas," *Ocean Engineering*, 42, 13–25.

Tanaka, M and Masunaga, T (2010). "Stabilization and smoothing of pressure in MPS method by Quasi-Compressibility," *Journal of Computational Physics*, 229, 4279-4290.

Tang, ZY, Zhang, YX and Wan, DC (2014). "Overlapping MPS Method for 2D Free Surface Flow," *ISOPE International Society of Offshore and Polar Engineers*, Busan, 3, 411-419.

Tang, ZY, Zhang, YL and Wan, DC (2016). "Multi-resolution MPS method for free surface flows," *International Journal of Computational Methods*, 13(04).

Yang, YQ, Tang, ZY, Zhang, YL and Wan, DC (2015). "Investigation of Excitation Period Effects on 2D Liquid Sloshing by MPS Method," *Proceedings of the Twenty-fifth (2015) International Ocean and Polar Engineering Conference*, Hawaii, USA, 937–944.

Zhang, C, Zhang, YX and Wan, DC (2011). "Comparative study of SPH and MPS methods for numerical simulations of dam breaking problems," *Chinese Journal of Hydrodynamics*, 26, 736–746.

Zhang, YX and Wan, DC (2011a). "Apply MPS method to simulate motion of floating body interacting with solitary wave". *IWSH International Workshop on Hydrodynamics*, Shanghai, 275-279.

Zhang, YX and Wan, DC (2011b). "Application of improved MPS method in sloshing problem," *Proc. of the 23rd Chinese Symposium on Hydrodynamics*, Xi'an.

Zhu, XS, Cheng, L, Lu, L and Teng, B (2011). "Implementation of the moving particle semi-implicit method on GPU," *Science China*, 54, 523–532.