The Simulation of Three-Dimensional Flow by Using GPU-based MPS Method

Xiang Chen
State Key Laboratory of Ocean Engineering, School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Collaborative Innovation Center for Advanced Ship and Deep-Sea Exploration
Shanghai 200240, China

Decheng Wan*
State Key Laboratory of Ocean Engineering, School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Collaborative Innovation Center for Advanced Ship and Deep-Sea Exploration
Shanghai 200240, China
*Corresponding author: dcwan@sjtu.edu.cn

***Abstract*—A modified moving particle semi-implicit (MPS) method based on GPU acceleration technique is applied to simulate three-dimensional (3-D) free surface flow by using our in-house solver MPSGPU-SJTU in this work. In order to validate MPSGPU-SJTU solver, 3-D dam break and sloshing, two typical violent flows with large deformation and nonlinear fragmentation of free surface are simulated. For dam break case, the results of fluid flied, water front, wave height and impact pressure by GPU simulation are compared to CPU calculation, experiment, smooth particle hydrodynamics (SPH) and boundary element method (BEM). The comparison of fluid field and impact pressure among GPU, CPU and experiment is made in sloshing case. The accuracy of GPU solver is verified by these comparisons. Moreover, the computation time of every part in each calculation step is compared between GPU and CPU solvers. The results show that computational efficiency is improved dramatically by employing GPU acceleration technique.**

I. INTRODUTION

Moving particle semi-implicit method is one Lagrangian meshless method for incompressible fluid field, which was introduced by Koshizuka and Oka in 1996 [1]. Similar to other meshless methods like SPH, many randomly distributed particles are used to represent the fluid domain in MPS. These particles contain the information of mass, momentum, pressure and so on. The pattern of solving Navier-Stokes equation is semi-implicit, which is distinctive feature of MPS. The stable pressure field of fluid can be obtained from solving pressure Poisson equation. Because of the particles representation, MPS can easily track free surfaces and moving boundaries and remove many numerical difficulties due to the nonlinear surface. In the resent years, more and more researchers have used MPS to simulate the problems of violent flow with large deformation or nonlinear fragmentation of free surface, such as dam break (Zhang et al. (2011) [2]), sloshing (Yang et al. (2015) [3]), water entry (Chen et al. (2017) [4]), fluid-structure interaction (Zhang et al. (2016) [5]) and so on.

In order to obtain more accuracy and stable results, many researchers devoted themselves to improving the calculation accuracy and suppressing the pressure oscillation. Many numerical models of MPS are modified such as kernel function (Koshizuka et al. (1998) [6], Ataie-Ashtiani and Farhadi (2006) [7]), gradient model (Koshizuka et al. (1998) [6], Khayyer and Gotoh (2008) [8], Tsuruta et al.(2013) [9]), Laplacian model (Khayyer and Gotoh (2012) [10], Ikari et al. (2015) [11]), pressure Poisson equation (Khayyer and Gotoh (2009) [12], Tanaka and Masunaga (2010) [13], Kondo and Koshizuka (2011) [14]) and free surface detection (Khayyer and Gotoh (2009) [12], Tanaka and Masunaga (2010) [13]). In the past years, MPS is usually applied to simulate the two-dimensional problems because of the low computational efficiency. The refined particles methods like multi-resolution (Tang et al. (2016) [15-16]) and overlapping (Shibata et al. (2012) [17], Tang et al. (2016) [18]) are typical numerical acceleration technologies to reduce the calculation amount. In addition, many researchers use CPU parallel technique to accelerate the calculation of MPS (Ikari and Gotoh (2008) [19], Iribe et al. (2010) [20]).

By using CPU parallel technique, it is found that the computation time of MPS is reduced with the increase of calculation cores. The graphics processing unit (GPU) whose remarkable feature is multi cores have been produced with the development of industry. Based on GPU acceleration technique, many meshless methods are applied to simulate massive problems. The application of GPU technique in SPH is more mature than MPS. Harada et al. (2007) developed one search method for neighboring particles in order to implement the SPH entirely on GPU. By the limit of GPU card capacity, the maximum particle number is four million and the maximum speedup is 28 [21]. Crespo et al. (2011) developed DualSPHysics solver based on GPU acceleration technique. They used this solver to simulate 3-D dam break problem with one million particles and achieved a speedup of 64 by comparing to one CPU core [22]. Then Domínguez et al. (2013) optimized DualSPHysics solver based on the characters of GPU and accelerated the SPH codes with a maximum speedup of 56.2 [23]. Mokos et al. (2015) developed two-phase GPU code to simulate 3-D dam break with obstacle and obtain high acceleration ratio on different GPU card [24]. Because pressure Poisson equation is solved implicitly, the acceleration effect of GPU for MPS is not remarkable and the research of this field is rare. Zhu et al. (2011) developed different versions of MPS code based on different GPU memories [25]. 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 [26]. 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 [27]. Gou et al. (2016) used GPU accelerated MPS to simulate the isothermal multi-phase fuel-coolant interaction [28].

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 and sloshing problems. The numerical results of GPU code such as fluid field, impact pressure, wave height and water front 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.

II. NUMERICAL METHOD

In this paper, the simulation of flow is calculated by our in-house particle solver MPSGPU-SJTU based on modified MPS method. The applied numerical models are introduced briefly in this section.

A. Governing Equations

The governing equations for viscous incompressible fluid contain continuity equation and Navier-Stokes equation.

$$\frac{1}{\rho}\frac{D\rho}{Dt} = \nabla \cdot \vec{V} = 0 \tag{1}$$

$$\frac{D\vec{V}}{Dt} = -\frac{1}{\rho}\nabla P + \nu \nabla^2 \vec{V} + \vec{g} \tag{2}$$

where ρ is the fluid density, t is the time, $\vec{V}$ is the velocity vector, P is the pressure, ν is the kinematic viscosity and $\vec{g}$ is the gravitational acceleration vector.

B. Kernel Function

In MPS method, the particle interaction is described by a kernel function. Zhang and Wan (2012) developed a modified kernel function in order to avoid the singularity at r=0 in original version [29].

$$W(r) = \begin{cases} \dfrac{r_e}{0.85r + 0.15r_e} - 1 & 0 \le r < r_e \\ 0 & r_e \le r \end{cases} \tag{3}$$

where r is the distance between two particles and r_e is the radius of the particle interaction.

C. Particle Interaction Models

The models of particle interaction include gradient model, divergence model and Laplacian model for MPS. These models can be written as:

$$<\nabla \phi>_i = \frac{D}{n^0}\sum_{j\neq i}\frac{\phi_j + \phi_i}{|\vec{r}_j - \vec{r}_i|^2}(\vec{r}_j - \vec{r}_i)\cdot W(|\vec{r}_j - \vec{r}_i|) \tag{4}$$

$$<\nabla \vec{V}>_i = \frac{D}{n^0}\sum_{j\neq i}\frac{(\vec{V}_j - \vec{V}_i)\cdot(\vec{r}_j - \vec{r}_i)}{|\vec{r}_j - \vec{r}_i|^2}\cdot W(|\vec{r}_j - \vec{r}_i|) \tag{5}$$

$$<\nabla^2 \phi>_i = \frac{2D}{n^0\lambda}\sum_{j\neq i}(\phi_j - \phi_i)\cdot W(|\vec{r}_j - \vec{r}_i|) \tag{6}$$

$$\lambda = \frac{\sum_{j\neq i} W(|\vec{r}_j - \vec{r}_i|)\cdot|\vec{r}_j - \vec{r}_i|^2}{\sum_{j\neq i} W(|\vec{r}_j - \vec{r}_i|)} \tag{7}$$

where D is the space dimension, n^0 is the initial particle number density, $\vec{r}$ is coordinate vector of particle, ϕ is any physical quantity and λ is applied to make sure that the increase of variance is equal to the analytical solution..

D. Model of Incompressibility

Lee et al. (2015) improved a mixed source term method (Tanaka and Masunaga(2010) [13]) combined with the velocity divergence-free condition and constant particle number density condition [30].

$$<\nabla^2 P^{k+1}>_i = (1-\gamma)\frac{\rho}{\Delta t}\nabla\cdot\vec{V}_i^* - \gamma\frac{\rho}{\Delta t^2}\frac{<n^*>_i - n^0}{n^0} \tag{8}$$

where γ is a variable parameter from 0 to 1, Δt is the time step, n^* is the temporal particle number density and defined as:

$$<n>_i = \sum_{j\neq i} W\left(\left|\boldsymbol{r}_j - \boldsymbol{r}_i\right|\right) \tag{9}$$

E. Free Surface Detection

In MPS method, the Dirichlet boundary condition is imposed by assigning zero pressure for surface particles. Zhang and Wan (2012) developed a modified surface particle detection method, which is based on the asymmetry arrangement of neighboring particles [29].

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

$$< |\vec{F}|>_i \; > \alpha \quad (11)$$

$$\alpha = 0.9 \left|\vec{F}\right|^0 \quad (12)$$

where $\vec{F}$ is a vector which represents the asymmetry of arrangements of neighbor particles, $\left|\vec{F}\right|^0$ is the initial value of $\left|\vec{F}\right|$.

F. Boundary Condition

In this work, multilayer particles are used to present the wall boundary. The wall particles are arranged at the boundary and the pressures of them are solved by PPE. Two layers of ghost particles are configured to fulfill the particle number density near the boundary so that the particle interaction can be properly simulated near the boundary. The pressure of ghost particle is obtained by interpolation.

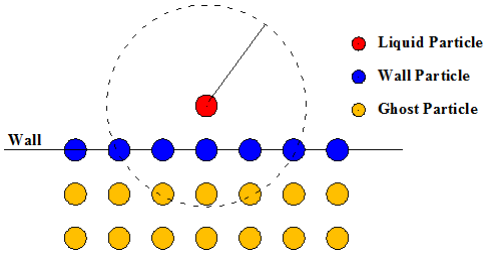

Figure 1. Schematic of boundary particles

III. GPU ACCELERATION

Based on the above brief introduction, one feature of MPS method can be found that the calculation of each particle is independent of the synchronous results of other particles except solving pressure Poisson equation. This feature determines that the calculation flow of MPS can be effectively parallelized. Comparing to CPU, GPU is designed possess more arithmetic logic units (ALU) in the same chip area. This hardware design makes GPU to own high floating point operations per second (FLOPS) and ability to process multi objects simultaneously.

CUDA is a parallel computing platform and programming model created by NVIDIA and implemented by GPU [31]. A CUDA program is divided into a host part and a device part. The host part runs on CPU while the device part runs on GPU. The host code includes instructions for setting parallelism and communicating data between host and device.

In order to accelerate the iteration of pressure Poisson equation, the open source library CUSP is applied in GPU solver. Cusp is a library for sparse linear algebra and graph computations based on Thrust. Cusp provides a flexible, high-level interface for manipulating sparse matrices and solving sparse linear systems [32].

The computational flow chart of MPS method is shown in Fig. 2. One time integration of MPS method is mainly composed of two steps. The first step corresponds to an explicit calculation considering the gravity and viscosity terms. The second step is an implicit calculation accounting for the pressure term. The pressure field of particles is obtained by solving pressure Poisson equation which is discretized into a linear system. The GPU implementation mainly consists of eight steps except the data exchange between GPU and CPU.

Step 1: Neighbor particle searching

Step 2: Explicit calculation of gravity and viscous force

Step 3: Updating for temporary velocity of particles

Step 4: Calculation of particle number

Step 5: Free surface detection

Step 6: Solving the pressure Poisson equation implicitly

Step 7: Calculation of pressure gradient

Step 8: Updating for velocity and position of particles finally

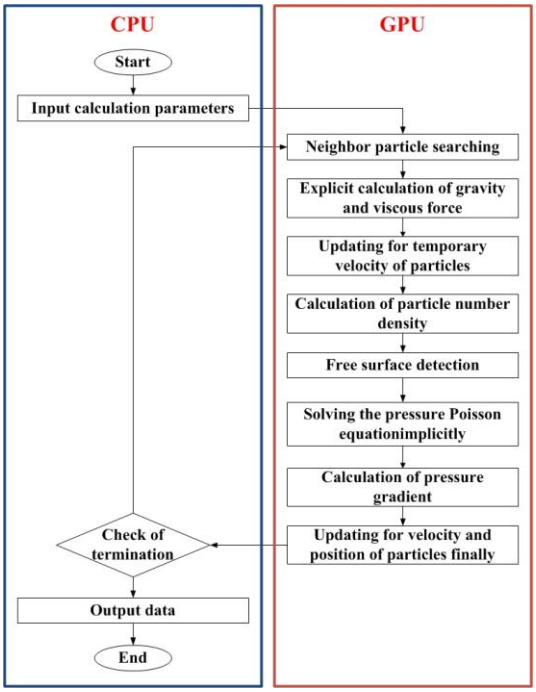

Figure 2. The flow chart of GPU implementation

IV. NUMERICAL SIMULATION

In this section, the results of GPU simulations are obtained by running MPSGPU-SJTU solver. In addition, another in-house CPU solver MLParticle-SJTU is used to

compare with GPU solver. The reliability of MLParticle-SJTU solver was validated by many violent flow cases in previous articles [2-5]. The comparisons between CPU and GPU include fluid field, monitoring data, computation time and so on. In this paper, all 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.

TABLE I. COMPUTATIONAL ENVIRONMENT OF CPU AND CPU

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

A. Dam Break Flow

Dam break flow is a typical violent free surface flow with complex phenomena such as the overturning of free surface, splashing and jet flow. In this sub-section, a 3-D dam break flow is numerically simulated by MPSGPU-SJTU solver and MLParticle-SJTU solver, respectively. The numerical model is the same as the experimental facility given by Colicchio (2001). Fig. 3 shows the sketch of computational domain. For fluid domain, the height of water column (H) is 0.6 m and the length is 1.2 m. One pressure probe and one wave gauge are placed in the tank to measure the impact pressure on lateral wall and wave height. The arrangements of monitoring points are listed in Table 2. In this case, the initial particle space is 0.01 m. In total, 1199205 particles with 712800 fluid particles are used to model. The time step is 2.5×10^{-4} s and the density of liquid is 1000 kg/m^3.

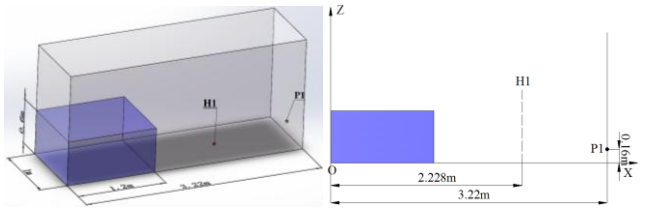

Figure 3. The sketch of model

TABLE II. ARRANGEMENTS OF PROBES

	X/m	Y/m	Z/m
H1	2.228	0	0
P1	3.22	0.5	0.16

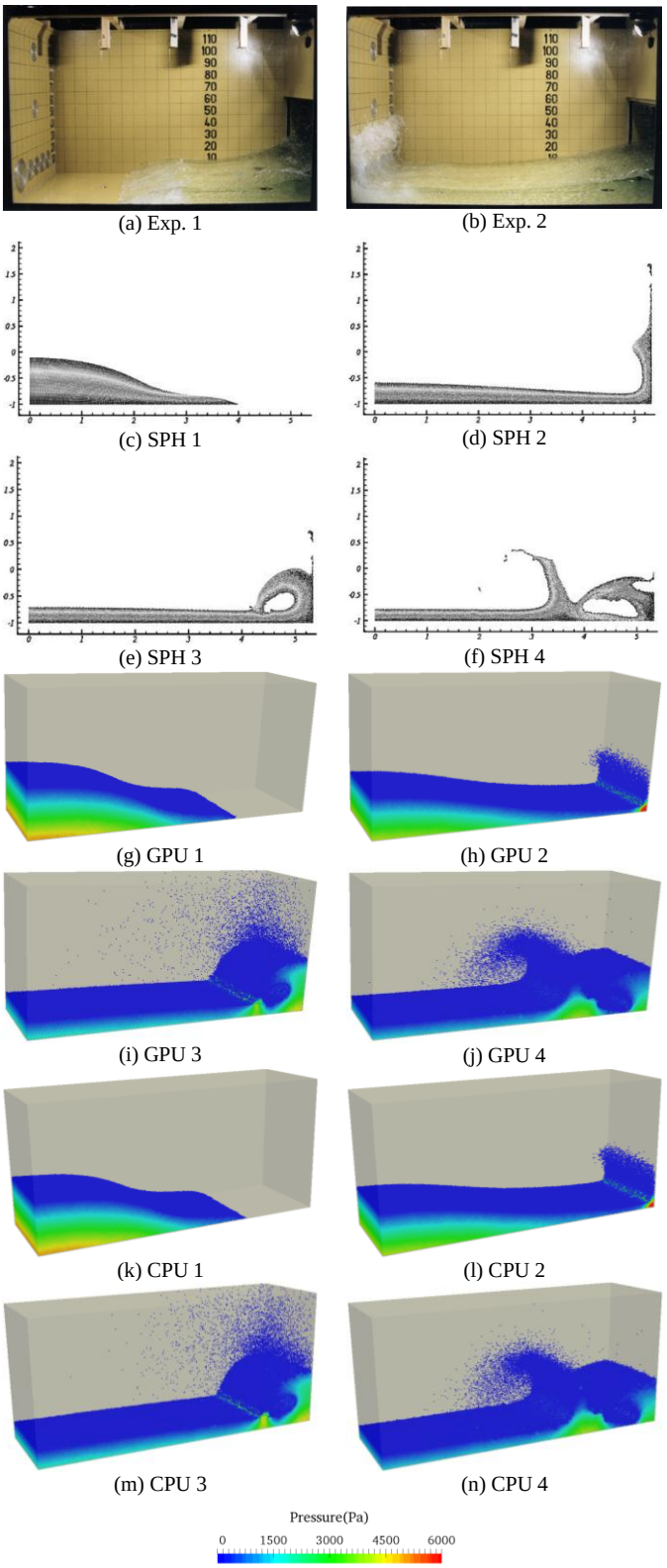

Figure 4. The flow fields of experiment, SPH, GPU and CPU

Fig. 4 shows some snapshots of numerical and experimental flow fields. After the water column is released, the water front firstly moves along the dry bottom of tank. The free surface is smooth and the pressure of fluid field is equal to hydrostatic pressure. Then the water front impacts the corner of tank and the pressure around the corner increases suddenly. The water front runs up along the lateral wall and a part of fluid splashes. Under the action of gravity, the subsequent fluid forms the overturning of free surface and falls into the lower fluid domain. The fallen water against the surface generates the second successive curling wave. From these figures, the numerical flow field of GPU simulation is in good agreement with CPU, SPH, BEM and experimental results.

In addition, some simulated results of GPU are also compared to the results of GPU, SPH, BEM and the data of experiment. Fig. 5 gives the propagation of water front. All the results show that the fluid accelerates smoothly and reaches to a stable velocity. The wave front propagation along the dry bottom of tank by GPU simulation is similar to the result of CPU and also in agreement with SPH and BEM. However, the experimental propagation speed of water front is slower than the results of numerical simulation.

Fig. 6 shows the comparison of impact pressure on the right wall among numerical methods and experiment. The overall tendency of pressure history by MPSGPU-SJTU solver shows a good agreement with CPU solver, SPH, BEM and experimental data. There are two peak values in the pressure history. After the process of water front moves along the bottom of tank, the first peak value of pressure is suddenly caused by the impact of fluid on the side wall. And the second pressure peak results from the impact of fallen overturning water on the free surface of lower fluid domain. There is a clear phase difference and value difference of the second pressure peak between numerical results and experimental data.

The variation of wave height by many numerical simulations and experiment is shown in Fig. 7. Because of the transition from dry-deck condition to wet-deck condition, the wave height increases suddenly. Then the wave height rises slowly with the process of fluid movement. Due to the overturning of water front, the wave height reaches to the maximum value in whole process. The peak value of wave height by numerical simulations is obviously higher than that of experiment. For these differences between numerical methods and experiment, one or two phase model is the possible reason mentioned by Colagrossi and Landrini (2003). In addition, BEM con not capture the phenomena of nonlinear stage such as overturning of free surface and splashing from recorded pressure and wave height. However, these phenomena can be qualitatively simulated by MPS and SPH.

Fig. 8 shows the computation time of every part in one step by GPU and CPU simulations. For MPS, solving pressure Poisson equation has the most amount of time which limits the computational efficiency. The computation time of every part decreases with the increase of CPU cores. The computational efficiency of GPU is remarkable and the speedup is up to 18 by comparing to one CPU core in Fig. 9.

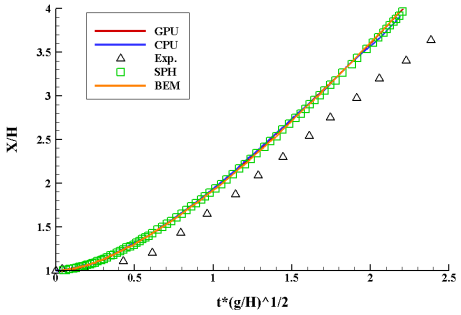

Figure 5. The water-front of GPU, CPU experiment, SPH and BEM

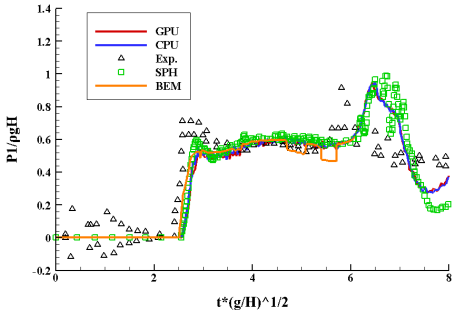

Figure 6. The impact pressure of GPU, CPU experiment, SPH and BEM

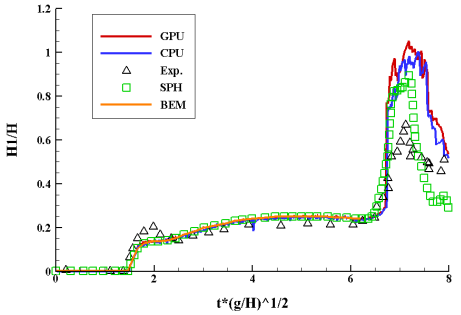

Figure 7. The wave height of GPU, CPU experiment, SPH and BEM

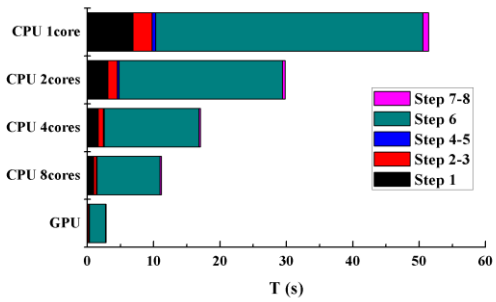

Figure 8. The computation times of GPU and CPU

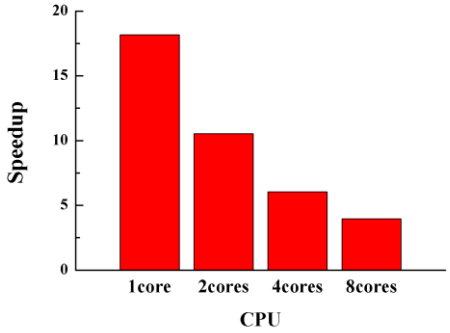

Figure 9. The speedup by GPU

B. *Sloshing Flow*

A 3-D liquid tank under horizontal excitation which is the same as the experimental model given by Song et al. (2013) is selected as numerical model to simulate in this sub-section. The sketch and geometric parameters of the liquid tank are shown in Fig. 10. The length of tank is 0.79 m, the width and the height are 0.48 m. The filling level is 30% and the corresponding depth of water is 0.144 m. Two pressure probes are placed at the lateral wall to measure the variation of impact pressure. The whole sloshing system is subject to move by the external surge excitation:

$$x = A \cdot \sin(\omega \cdot t) \tag{13}$$

where A is the amplitude of excitation with the value of 0.0575 m and ω is the excitation frequency which is set to 4.49 rad/s. The initial particle space is 0.005 m, the time step is 2×10^{-4} s and the density of liquid is 1000 kg/m^3. In this case, total 678373 particles including 432535 fluid particles are used to simulate this model.

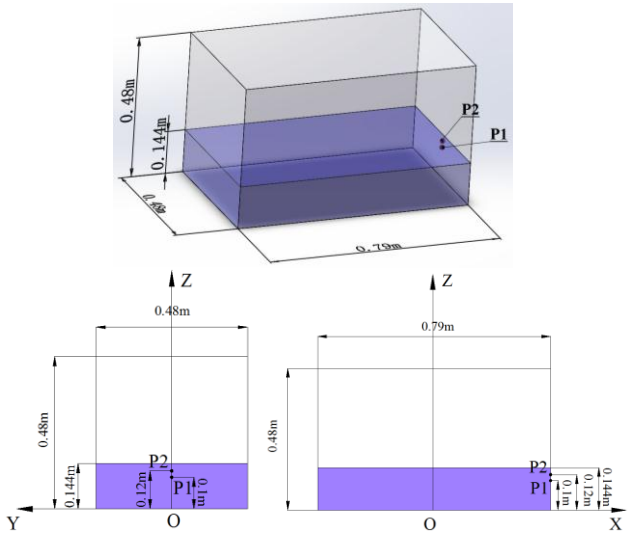

Figure 10. The sketch of model

TABLE III. ARRANGEMENTS OF TWO PRESSURE PROBES

	X/m	Y/m	Z/m
P1	0.395	0	0.1
P2	0.395	0	0.12

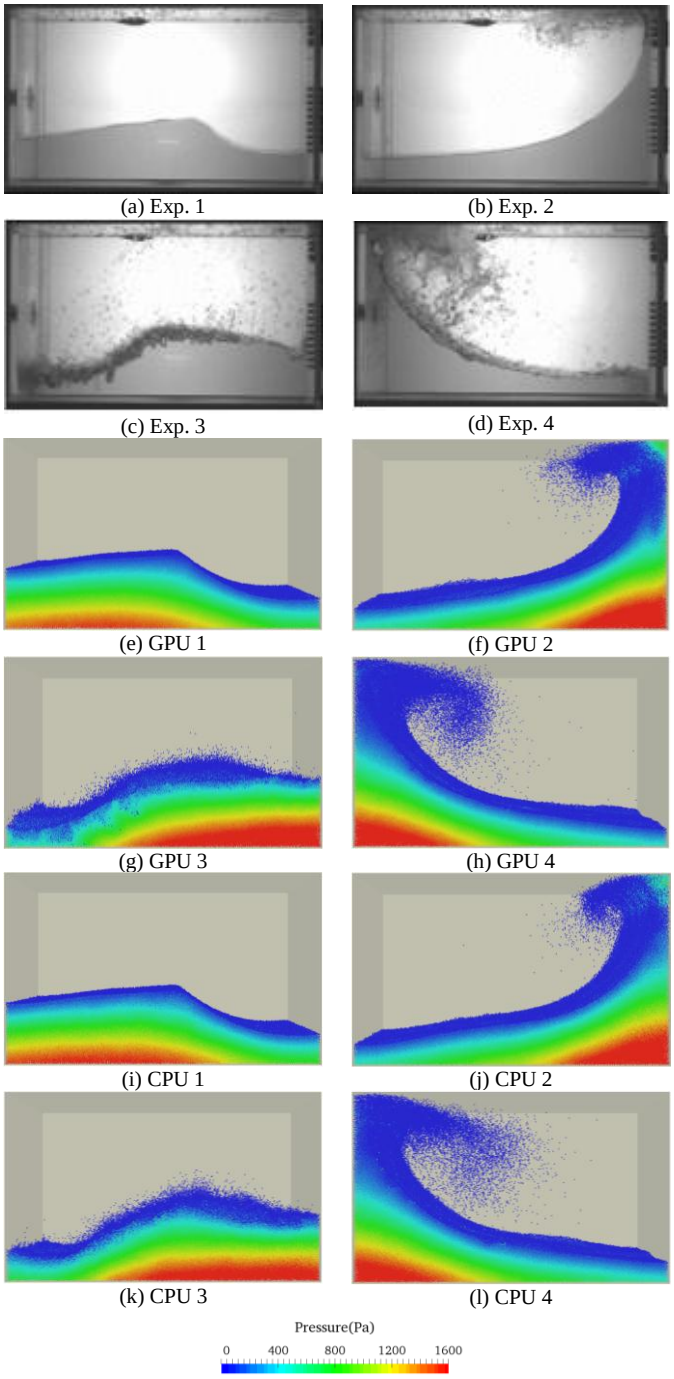

Figure 11. The flow fields of experiment, GPU and CPU

Some snapshots of experimental and numerical flow fields are shown in Fig. 11. The GPU simulation is in good agreement with the flow fields of CPU simulation and experiment. The fluid field is forced to move by the

movement of tank. The sloshing wave travels to the right walls when the tank moves right. Then the water front impacts and climbs along the lateral wall. And the front of fluid hits the corner and forms jet flow which still spreads along the ceiling of tank. Under the action of gravity, the jet flow falls into the lower fluid field and moves to the left wall. From Fig. 11, the obvious nonlinear phenomena such as overturning of water and liquid splash can be observed. These large deformation and nonlinear fragmentation of free surface are simulated by both MLParticl-SJTU solver and MPSGPU-SJTU solver.

In addition, Fig. 12 shows the numerical pressure histories of CPU and GPU by comparing to the experimental data. A typical impact pressure pattern "church roof" can be observed in the figure. When the liquid tank reaches its maximum position on the right and starts to move left, the fluid field still moves to and impinges the right wall, which results in an instant pressure peak. The sloshing flow runs up along the right wall with the decrease of pressure. The fallen fluid which drops down on the free surface due to the gravity causes the second pressure peak. Finally, the pressure is less than zero, because the depth of water near right-side wall is below pressure probes while the fluid field moves to the left side wall.

Fig .13 also gives the comparison of computation time between GPU and CPU. The performance of GPU is outstanding in the conditions of guaranteeing the accuracy. The GPU solver can reduce the computation time of every part up to one order. In addition, the acceleration ratio between GPU and one CPU core is up to 35 from Fig. 14.

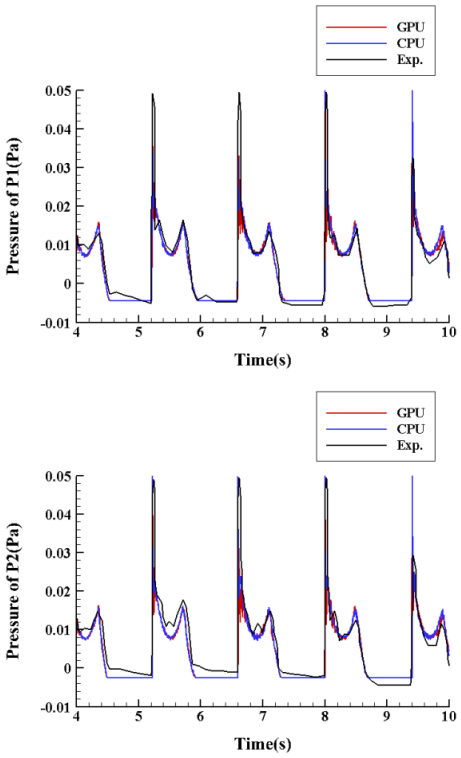

Figure 12. The impact of experiment, GPU and CPU

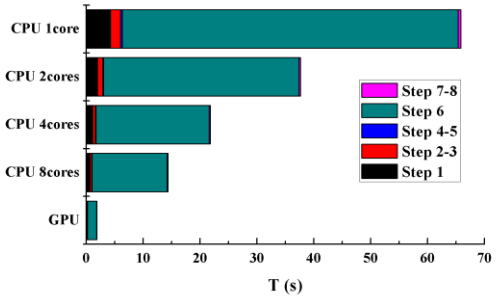

Figure 13. The computation times of GPU and CPU

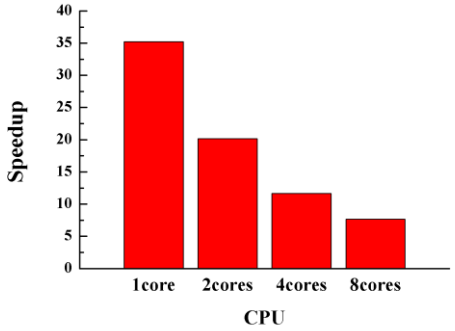

Figure 14. The speedup by GPU

V. CONCLUSIONS

In this work, the in-house solver MPSGPU-SJTU based on modified MPS method is developed to simulate the three-dimensional violent flows such as dam break and sloshing. The large deformation and nonlinear fragmentation of free surface, like overturning wave, jet flow, splashing and so on, can be observed clearly in these numerical simulations. The numerical results such as impact pressure on the lateral wall, wave height and water front of GPU simulation shows a good agreement with CPU calculation, SPH, BEM and experiment. These comparisons demonstrate the validity of MPSGPU-SJTU solver. In addition, the GPU solver can dramatically reduce the computation time and improve the computational efficiency. The speedup of every calculation step between GPU and CPU solvers is up to 35.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (51379125, 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.

REFERENCE

[1] S. Koshizuka and Y. Oka. "Moving-particle semi-implicit method for fragmentation of incompressible fluid," Nuclear Science and Engineering, vol. 123(3), pp.421–434, July 1996.

[2] C. Zhang, Y. X. Zhang, and D. C. Wan. "Comparative study of SPH and MPS methods for numerical simulations of dam breaking problems," Chinese Journal of Hydrodynamics, vol. 26, pp.736–746, June 2011.

[3] Y. Q. Yang, Z. Y. Tang, Y. L. Zhang, and Wan, DC. "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, June 2015.

[4] X. Chen, C. P. Rao, and D. C. Wan. "Numerical simulation of water entry for two-dimensional wedge by MPS," Chinese Journal of Computational Mechanics, vol. 34, pp.356–362, June 2017.

[5] Y. L. Zhang, Z. Y. Tang, and D. C. Wan. "Numerical Investigations of Waves Interacting with Free Rolling Body by Modified MPS Method," International Journal of Computational Methods, vol. 13: 1641013–1–1641013–14, August 2016.

[6] B. Ataie-Ashtiani and L. Farhadi. "A stable moving-particle semi-implicit method for free surface flows," Fluid Dynamics Research, vol. 38, pp.241–256, April 2006.

[7] S. Koshizuka, A, Nobe, and Y. Oka. "Numerical analysis of breaking waves using the moving particle semi-implicit method," International Journal for Numerical Methods in Fluids, vol. 26, pp.751–769, April 1998.

[8] A. Khayyer and H. Gotoh. "Development of CMPS method for accurate water-surface tracking in breaking waves," Coastal Engineering Journal, vol. 50, pp.179–207, June 2008.

[9] N. Tsuruta, A. Khayyer, and H. Gotoh . "A short note on dynamic stabilization of moving particle semi-implicit method," Computers & Fluids, vol. 82, pp.158–164, August 2013.

[10] A. Khayyer and H. Gotoh. "A 3D higher order Laplacian model for enhancement and stabilization of pressure calculation in 3D MPS-based simulations," Applied Ocean Research, vol. 37, pp.120–126, August 2012.

[11] H. Ikari, A. Khayyer and H. Gotoh. "Corrected higher order Laplacian for enhancement of pressure calculation by projection-based particle methods with applications in ocean engineering," Journal of Ocean Engineering and Marine Energy, vol. 1, pp.361–376, November 2015.

[12] A. Khayyer and H. Gotoh. "Modified moving particle semi-implicit methods for the prediction of 2D wave impact pressure," Coastal Engineering, vol. 56, pp.419–440, April 2009.

[13] M. Tanaka, and T. Masunaga. "Stabilization and smoothing of pressure in MPS method by quasi-compressibility," Journal of Computational Physics, vol. 229, pp.4279–4290, June 2010.

[14] M. Kondo, and S. Koshizuka. "Improvement of stability in moving particle semi-implicit method," International Journal for Numerical Methods in Fluids, vol. 65, pp.638–654, February 2011.

[15] Z. Y. Tang, Y. L. Zhang, and D. C. Wan. "Multi-resolution MPS method for free surface flows," International Journal of Computational Methods, vol. 13: 1641018, August 2016.

[16] Z. Y. Tang, D. C. Wan, G. Chen, and Q. Xiao. "Numerical Simulation of 3D Violent Free-Surface Flows by Multi-Resolution MPS Method," Journal of Ocean Engineering and Marine Energy, vol. 2, pp.355–364, July 2016.

[17] K. Shibata, S. Koshizuka, M. Sakai, and K. Tanizawa. "Lagrangian simulations of ship-wave interactions in rough seas," Ocean Engineering, vol. 42, pp.13–25, March 2012.

[18] Z. Y. Tang, Y. L. Zhang, and D. C. Wan. "Numerical simulation of 3D free surface flows by overlapping MPS," Journal of Hydrodynamics, vol. 28, pp.306–312, July 2016.

[19] H. Ikari, and H. Gotoh. "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, September 2008.

[20] T. Iribe, T. Fujisawa, and S. Koshizuka. "Reduction of communication in parallel computing of particle method for flow simulation of seaside areas," Coastal Engineering Journal, vol. 52, pp.287–304, December 2010.

[21] T. Harada, S. Koshizuka, and Y. Kawaguchi. "Smoothed Particle Hydrodynamics on GPUs," Structure, vol. 4, pp.671–691, January 2007.

[22] A. J. C. Crespo, J. M. Dom ínguez, A. Barreiro, M. G ómez-Gesteira, and B. D. Rogers. "GPUs, a new tool of acceleration in CFD: efficiency and reliability on smoothed particle hydrodynamics methods," PLoS One, vol. 6: e20685, June 2011.

[23] J. M. Dom ínguez, A. J. C. Crespo, and M. G ómez-Gesteira. "Optimization strategies for CPU and GPU implementations of a smoothed particle hydrodynamics method," Computer Physics Communications, vol. 184, pp.617–627, March 2013.

[24] A. Mokos, B. D. Rogers, P. K. Stansby, and J. M. Dom ínguez. "Multi-phase SPH modelling of violent hydrodynamics on GPUs," Computer Physics Communications, vol. 196, pp.304–316, November 2015.

[25] X. S. Zhu, L. Cheng, L. Lu, and B. Teng. "Implementation of the moving particle semi-implicit method on GPU," Science China, vol. 54, pp.523–532, March 2011.

[26] C. Hori, H. Gotoh, H. Ikari, and A. Khayyer. "GPU-acceleration for moving particle semi-implicit method," Computers & Fluids, vol. 51, 174–183, December 2011.

[27] H. Z. Li, Y. L. Zhang, and D. C. Wan. "GPU Based Acceleration of MPS for 3D Free Surface Flows," Proceedings of the 9th International Workshop on Ship and Marine Hydrodynamics, Glasgow, UK, August 2015.

[28] W. Gou, S. Zhang, and Y. Zheng. "Simulation of isothermal multi-phase fuel-coolant interaction using MPS method with GPU acceleration," Kerntechnik, vol. 81, pp.330–336, June 2016.

[29] Y. X. Zhang, and D. C. Wan. "Numerical simulation of liquid sloshing in low-filling tank by MPS," Journal of Hydrodynamics, vol. 27, pp.101–107, January 2012.

[30] B. H. Lee, J. C. Park, M. H. Kim, and S. C. Hwang. "Step-by-step improvement of MPS method in simulating violent free-surface motions and impact-loads," Computer Methods in Applied Mechanics and Engineering, vol. 200, pp.1113–1125, February 2011.

[31] CUDA Toolkit Documentation v8.0.61. http://docs.nvidia.com/cuda/. 2017.

[32] CUSP. https://developer.nvidia.com/cusp. 2017.

[33] G. Colicchio, A. Colagrossi, M. Greco, and M. Landrini. "Free surface flow after a dam break a comparative study," 4th Numerical Towing Tank Symposium, Germany, September 2001.

[34] Y. K. Song, K. A. Chang, Y. Ryu, and S. H. Kwon. "Experimental study on flow kinematics and impact pressure in liquid sloshing," Experiments in Fluids, vol. 54, pp.1–20, September 2013.