Proceedings of the ASME 2018 37th International
Conference on Ocean, Offshore and Arctic Engineering
OMAE2018
June 17-22, 2018, Madrid, Spain

OMAE2018-78369

GPU BASED ACCELERATION OF MPS METHOD FOR THREE-DIMENSIONAL DAM-BREAK FLOWS

Xiang Chen
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, 200240, China

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, 200240, China
*Corresponding author: dcwan@sjtu.edu.cn

ABSTRACT

The Moving Particle Semi-implicit (MPS) method has been proven effective to simulate violent flows such as dam-break flow, liquid sloshing and so on. But the low computational efficiency is one disadvantage of MPS. In the field of scientific computations, GPU based acceleration technique is widely applied to reduce the computation time of various numerical methods. In this paper, an in-house solver MPSGPU-SJTU is developed based on modified MPS method and GPU acceleration technique. A three-dimensional (3-D) dam-break flow is simulated by present solver and the validity and accuracy of GPU code are investigated by comparing the results with those by other researches. By comparisons, the flow field of GPU-based calculation is in better agreement with the experiment. In addition, the computation times of GPU and CPU solvers are compared to demonstrate the effect of GPU acceleration technique on the computational efficiency of MPS method.

INTRODUCTION

The dam-break flow caused by a sudden break of a holding barrier may generate violent impact on the downstream structure. Therefore, the evolution of dam-break flow has become a focus of researchers. Dressler (1958) introduced an analytical solution and conducted the dam break experiments to verify the solution [1]. Kleefsman et al. (2005) used the Volume-of-Fluid (VOF) method to displace the free surface of dam-break flow [2]. The adapting hierarchical grids technique was applied by Greaves (2006) to simulate water column collapse [3]. However, the dam-break flow usually accompanies the nonlinear phenomena such as splashing and breaking waves, which are difficult to be modeled by theoretical solution and grid methods. The meshfree methods can easily track free surface and effectively simulate the nonlinear phenomena in the dam-break flow problem. Chang et al. (2011) simulated the shallow-water dam break flows in open channels by using Smoothed Particle Hydrodynamics (SPH) method [4]. Another lagrangian method, Consistent Particle Method (CPM) was also employed to model large dam break problem [5]. Tang et al. developed the multi-resolution MPS to simulate a two-dimensional (2-D) dam break [6].

For meshfree methods, most simulations are limited to 2-D problems. However, it is necessary to use more than one million particles to simulate a practical 3-D problem. The low computational efficiency is the most problem in practical use of the particle method. One feature of meshfree methods is that the calculation of each particle is independent on the synchronous results of other particles. This feature determines that the calculation flow of meshfree methods can be effectively parallelized. At the same time, GPU (Graphics Processing Unit), a multi-processor, is designed to optimize for the execution of massive number of threads. Because of more arithmetic logic units (ALU) in the same chip area than CPU, GPU owns high floating point operations per second (FLOPS) and ability to process multi objects simultaneously. Therefore, applying GPU acceleration technique in meshfree methods is inevitable and feasible.

Due to the explicit time iteration, the application of GPU acceleration technique in SPH is earlier than MPS. Hérault et al. (2010) used CUDA to implement SPH on GPU. By simulating the dam-break flow, the computation times of three main components: neighbor list construction, force computation, and integration of the equation of motion are reduced significantly [7]. Based on GPU acceleration technique, Crespo et al. (2011) developed DualSPHysics solver to simulate 3-D dam-break flow problem with one million particles and achieved a speedup of 64 [8]. Taking advantage of powerful GPU parallel computing ability, Wu et al. used SPH to simulate large-scale dam-break flow in complex urban underground spaces [9]. Though the pressure field of MPS method is more stable and accurate than SPH by solving pressure Poisson equation, it is difficult to implement on GPU because of the semi-implicit algorithm. Hori et al. (2011) firstly used CUDA language to develop a GPU-based MPS code and simulated 2-D dam break with 7 times speedup [10]. Kakuda et al. (2012) used GPU-based MPS to calculate 2-D dam break problem and the speedup is 12 times [11]. Li et al. investigated the speedups of neighbor particle list and pressure Poisson equation in 2015 [12]. Overall, there are less research on GPU accelerated MPS.

In this paper, an in-house solver MPSGPU-SJTU is developed by applying GPU acceleration technique in modified MPS method. In order to verify the validity and accuracy of this solver, a dam-break flow benchmark is used to simulate and research. The flow field of calculation is compared to the experimental data and other numerical studies. And the calculated results such as the propagation of water front and the height of water column are analyzed. Furthermore, the acceleration performance of GPU parallel technique is investigated by the comparison of computation time between GPU-based and CPU-based codes.

MODIFIED MPS METHODOLOGY

Governing Equations and Solution Algorithms

For viscous incompressible fluid, the Navier-Stokes equations including mass and momentum conservation equations are used to describe the flow motion.

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

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

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

The overall calculation algorithm of MPS is described in Figure 1. Each time integration is mainly composed of two parts including eight steps. In the first part, the prediction of velocity is calculated by considering the gravity and viscosity terms form Equations. (1-2). The temporary velocity of particle is obtained as:

$$\vec{V}_i^* = \vec{V}_i^k + \Delta t(\nu\nabla^2\vec{V}_i^k + \vec{g}) \tag{3}$$

where $\vec{V}_i^*$ is the temporary velocity vector of particle i, $\vec{V}_i^k$ is the velocity vector of particle i at step k and Δt is the time step.

The second part is the correction step by accounting for the pressure term. The velocity and position of particle at the next time step can be written as:

$$\vec{V}_i^{k+1} = \vec{V}_i^* - \Delta t\frac{1}{\rho}\nabla P^{k+1} \tag{4}$$

$$\vec{r}_i^{k+1} = \vec{r}_i^k + \Delta t \cdot \vec{V}_i^{k+1} \tag{5}$$

Kernel Function

A particle interacts with others in its action range with kernel function. In order to avoid the singularity at r=0 in original version, a modified kernel function is developed by Zhang and Wan (2012) [13], where r is the distance between two particles.

$$W(r) = \begin{cases} \dfrac{r_e}{0.85r + 0.15r_e} - 1 & 0 \le r < r_e \\ 0 & r_e \le r \end{cases} \tag{6}$$

where r is the distance between two particles and r_e is the radius of the particle interaction. For the particle number density and gradient model, r_e is equal to $2.1l_0$. And r_e values as $4.0l_0$ in the Laplacian model. l_0 is the initial particle spacing.

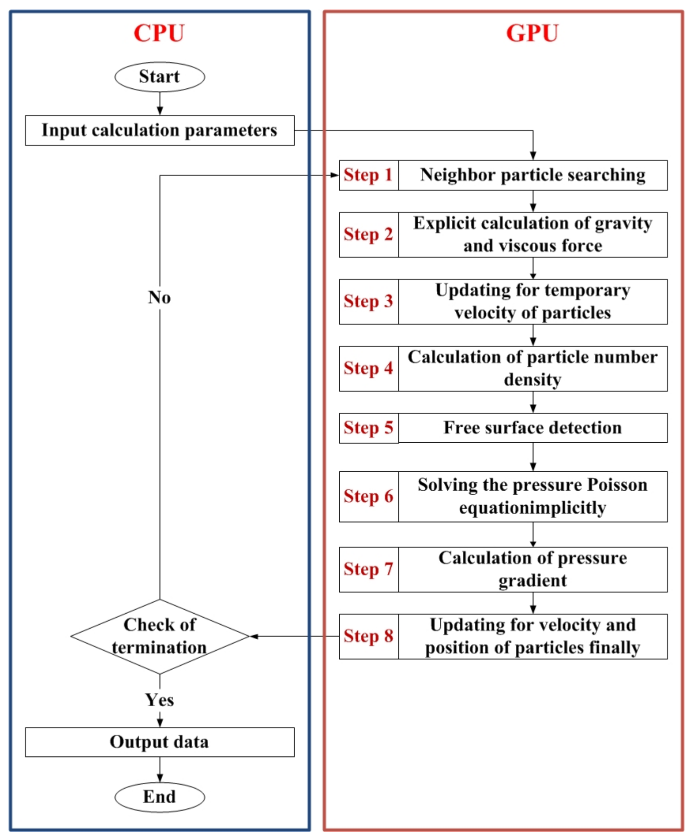


Figure 1. The calculation flow chart of MPS

Particle Interaction Models

The particle interaction models including gradient model, divergence model and Laplacian model can be written as:

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

$$< \nabla \vec{V} >_i = \frac{D}{n^0} \sum_{j \neq i} \frac{(\vec{V}_j - \vec{V}_i) \cdot (\vec{r}_j - \vec{r}_i)}{|\vec{r}_j - \vec{r}_i|^2} \cdot W(|\vec{r}_j - \vec{r}_i|) \tag{8}$$

$$< \nabla^2 \phi >_i = \frac{2D}{n^0 \lambda} \sum_{j \neq i} (\phi_j - \phi_i) \cdot W(|\vec{r}_j - \vec{r}_i|) \tag{9}$$

$$\lambda = \frac{\sum_{j \neq i} W(|\vec{r}_j - \vec{r}_i|) \cdot |\vec{r}_j - \vec{r}_i|^2}{\sum_{j \neq i} W(|\vec{r}_j - \vec{r}_i|)} \tag{10}$$

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

Model of Incompressibility

A mixed source term method combined with the velocity gradient and particle number density is used in this work [14].

$$< \nabla^2 P^{k+1} >_i = (1-\gamma) \frac{\rho}{\Delta t} \nabla \cdot \vec{V}_i^* - \gamma \frac{\rho}{\Delta t^2} \frac{< n^* >_i - n^0}{n^0} \tag{11}$$

$$< n >_i = \sum_{j \neq i} W\left(\left|\boldsymbol{r}_j - \boldsymbol{r}_i\right|\right) \tag{12}$$

where γ is a variable parameter from 0 to 1, n^0 is the initial particle number density and n^* is the temporal particle number density.

Surface Particle Detection

According to the characteristic of asymmetry particle arrangement, Zhang and Wan (2012) proposed a modified surface particle detection method [13].

$$< \vec{F} >_i = \frac{D}{n^0} \sum_{j \neq i} \frac{1}{|\vec{r}_i - \vec{r}_j|} (\vec{r}_i - \vec{r}_j) W(r_{ij}) \tag{13}$$

$$< |\vec{F}| >_i \; > \; 0.9 \left|\vec{F}\right|^0 \tag{14}$$

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 $\left|\vec{F}\right|$.

Boundary Condition

In order to ensure that the particle interaction can be properly simulated near the boundary, three layers of particles are used to present the wall boundary in MPS. One layer of wall particles are placed at the boundary and the pressures of them are solved by PPE. In addition, two layers of ghost particles are arranged to fulfill the particle number density near the boundary and the pressures of ghost particles are obtained by interpolation.

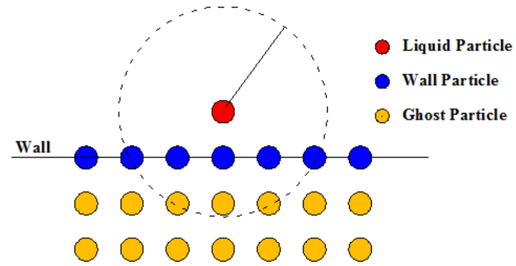


Figure 2. Schematic of boundary particles

GPU technique

As shown in Figure 1, all calculation steps are implemented on GPU device except the data exchange between CPU and GPU. In this paper, CUDA C/C++ is used to write programs. CUDA is a parallel computing platform and programming model created by NVIDIA and implemented by GPU [15]. A CUDA program is divided into two parts, a host part runs on CPU and a device part runs on GPU. The host code includes the instructions for setting parallelism and communicating data between host and device. And the device code specifically implements parallel computing of calculation algorithm.

It is well known that the most computation time of MPS is consumed to solve pressure Poisson equation. The pressures of free surface particles and ghost particles are set to zero as the dynamic free surface conditions. Therefore, the coefficient matrix of PPE is a typical sparse symmetry matrix. The open source library CUSP is applied to accelerate the iteration of pressure Poisson equation. Cusp is a library for sparse linear algebra and graph computations based on Thrust [16]. In addition, the compressed sparse row (CSR) data storage format is employed to save the coefficient matrix and the preconditioned conjugate gradient method (PCG) is used to solve PPE.

RESULTS AND DISCUSSIONS

In this section, the in-house solver MPSGPU-SJTU is used to simulate the problem of dam-break flow. The hardware used for the simulation includes a GPU card NVIDIA Tesla K40M, which has 2880 CUDA cores with 12GB graphics memory. And the numerical data are saved by double precision floating point. The configuration of GPU card is shown in Table 1.

Table 1. Configuration of GPU

Configuration	Value
Card	Tesla K40M
Graphics Memory	12GB
Max Cores	2880
Programming Language	CUDA C/C++
Compiler	CUDA 7.0

The dam-break flow is a typical violent flow with complex nonlinear phenomena such as splashing, jet flow and the overturning of free surface. In this sub-section, a 3-D dam-break flow the same as the experimental benchmark test [17-18] is numerically simulated by MPSGPU-SJTU solver. And this dam-break model is also numerically simulated by VOF method (Cao et al., 2013 [19]) and MPS method (Zhang and Wan, 2011 [20]). In the research of Zhang and Wan, the particle spacing is 0.0073 m, so only 71495 particles with 15200 fluid particles and 56295 wall particles are used. In present calculation, the initial particle spacing is selected as 0.002 m with the help of GPU acceleration technique. More than one million particles are used to finely model this problem and capture the details of dam-break flow. The sketch of dam-break flow model is shown in Figure 3. For fluid domain, the height of water column (H) is 0.292 m, the length (L) and the width (B) are all 0.146 m. The initial water column is blocked by a removable board in the experiment. The detailed computational parameters for simulation are shown Table 2.

Table 2. Computational parameters

Parameters	Value
Water Density	1000 kg/m^3
Kinematic Viscosity	1×10^{-6} m^2/s
Gravitational Acceleration	9.81 m/s^2
Particle Spacing	0.002 m
Fluid Number	767376
Wall Number	715986
Total Number	1483362
Time Step	0.0001 s

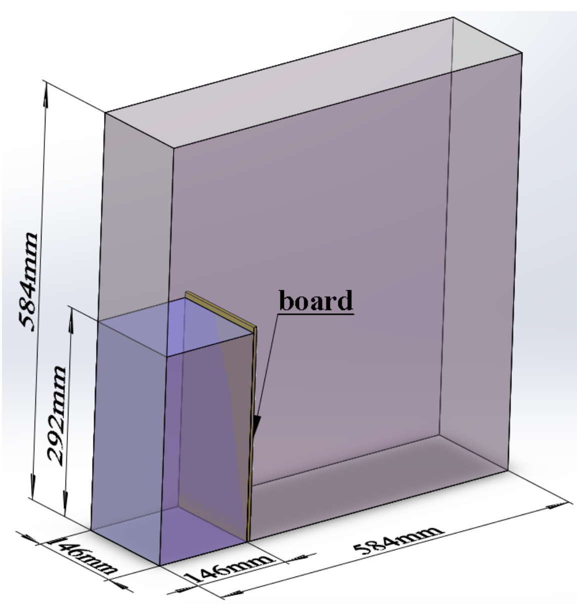
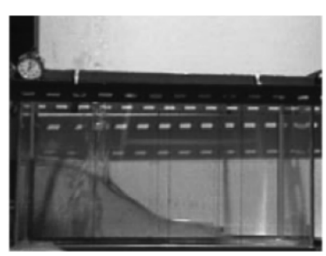


Figure 3. The sketch of model

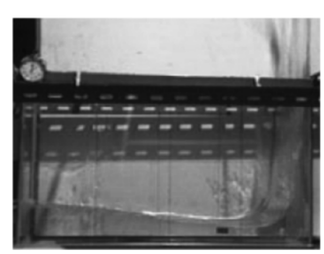
Figures 4-7 show the flow fields of different numerical simulations and experiment at some instants. After the initial water column is released, the collapsing water is moving along the bottom of tank at 0.2 s. At this stage, the pressure field is smooth and the free surface is sequential. Then the front of water impacts on the right vertical wall and the pressure around the corner increases suddenly. From Figures 4(b)-7(b), the water front runs up along the lateral wall and the height of water is above the initial column height at 0.4 s. Then the fallen water under the action of gravity joins the subsequent ascending fluid, which generates a heave on free surface in Figures 4(c)-7(c).

In the previous process of dam-beak flow, there is no difference between the results of other simulations and present work. But the flow fields of three numerical works present different details in the following time. In Figure 6(d), the fallen fluid blends into the bottom water and moves to the left wall. However, the falling water in the experiment, Cao's calculation and present simulation impacts fiercely the bottom free surface and is reflected in the upper-left direction. From Figures 4(d), 5(d) and 7(d), the two successive overturning waves and cavitation in fluid can be observed obviously. In Figure 6(e), the dam-break flow impacts the left lateral wall and the water front runs along the wall sequentially. But the nonlinear phenomena such as the fragmentation of free surface, overturning wave and splashing liquid are obvious in the experiment and simulation of MPSGPU-SJTU solver. From Figure 6(e) and 7(e), another overturning wave impacts the left lateral wall and produce local high pressure. In addition, there is a big cavitation near the left wall and a portion of liquid still drops down near the right lateral wall. Because of the limited particles, all these detailed phenomena of flow field are neglected in Figure 5. By applying GPU acceleration technique, the calculated flow field can be more real with the increase of particle number. Moreover, MPS method can effectively simulate nonlinear fragmentation of free surface such as splashing and breaking wave by comparing to VOF method.

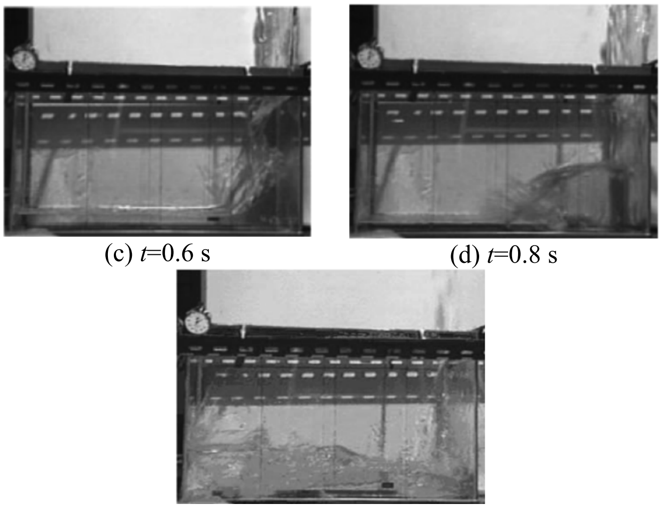
From Figures 6(f) and 7(f), the fluid impacts the left lateral wall and splashes in the upper-right direction. And the height of climbing water at 1.2 s is significantly less than the height at 0.4 s. Comparing to Figure 6(f), some small cavitations can be investigated in the flow field in Figure 7(f). Then the water front moves to the right later wall again and the free surface is more rugged than former flow. In addition, the water front is going to overturn in the process of moving forward because of the bottom water.



(a) t=0.2 s



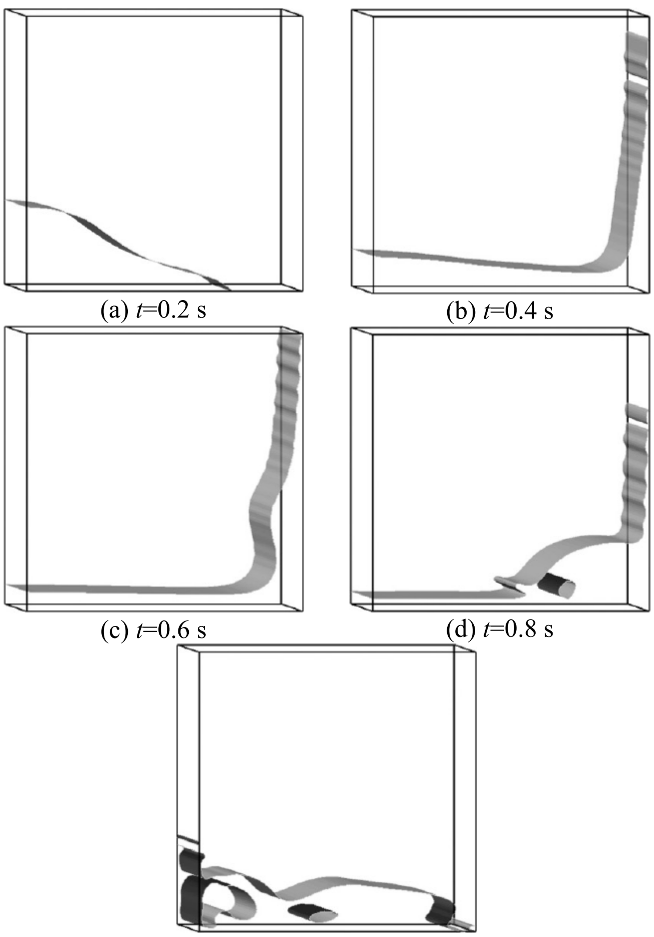
(b) t=0.4 s



(c) t=0.6 s (d) t=0.8 s

(e) t=1.0 s

Figure 4. Experimental dam-break flow (Koshizuka and Oka, 1996)

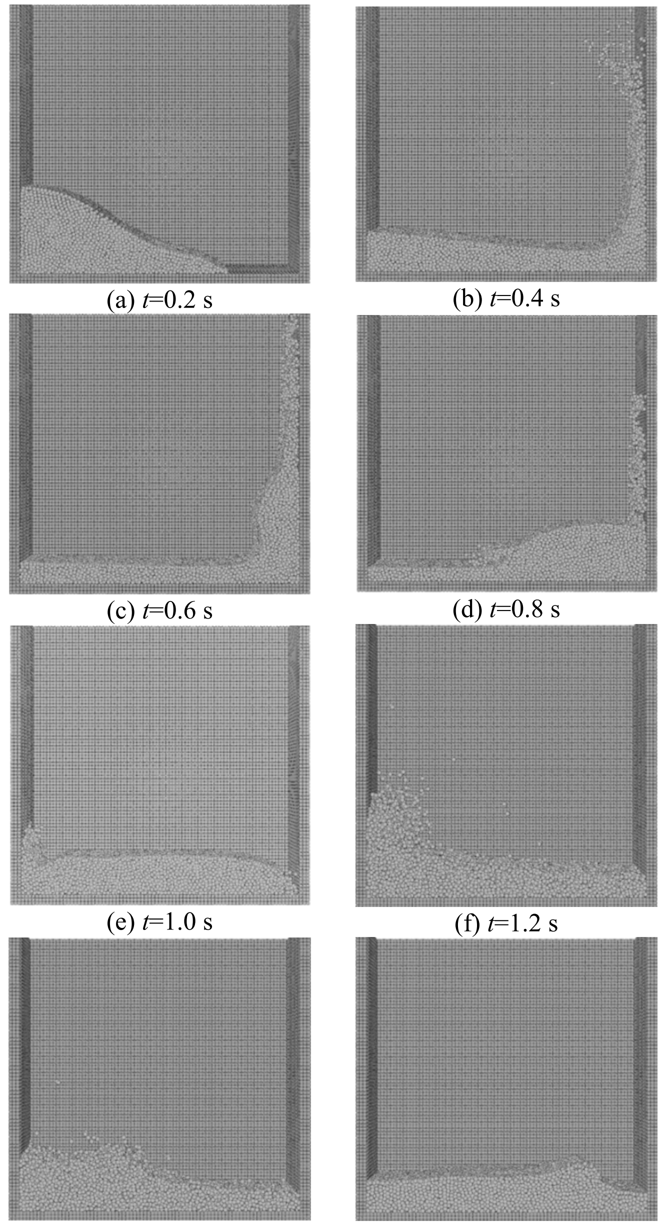


(a) t=0.2 s (b) t=0.4 s

(c) t=0.6 s (d) t=0.8 s

(e) t=1.0 s

Figure 5. Simulated dam-break flow by VOF (Cao et al., 2013)



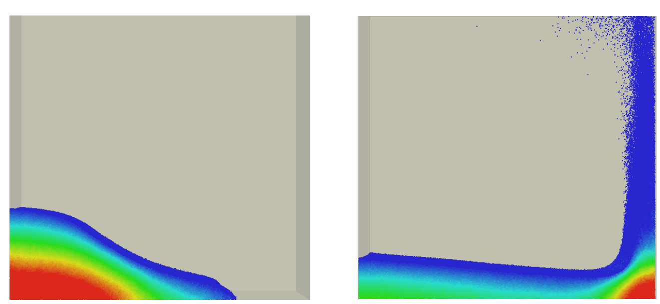
(a) t=0.2 s (b) t=0.4 s

(c) t=0.6 s (d) t=0.8 s

(e) t=1.0 s (f) t=1.2 s

(g) t=1.4 s (h) t=1.6 s

Figure 6. Simulated dam-break flow by MPS (Zhang and Wan, 2011)



(a) t=0.2 s (b) t=0.4 s

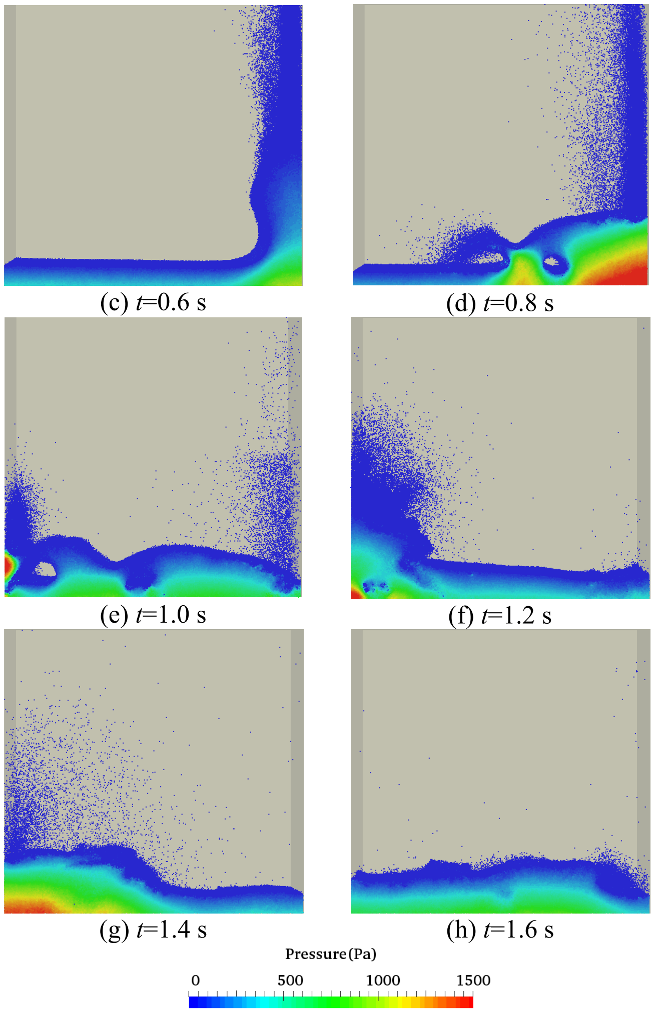


Figure 7. Simulated dam-break flow of present work

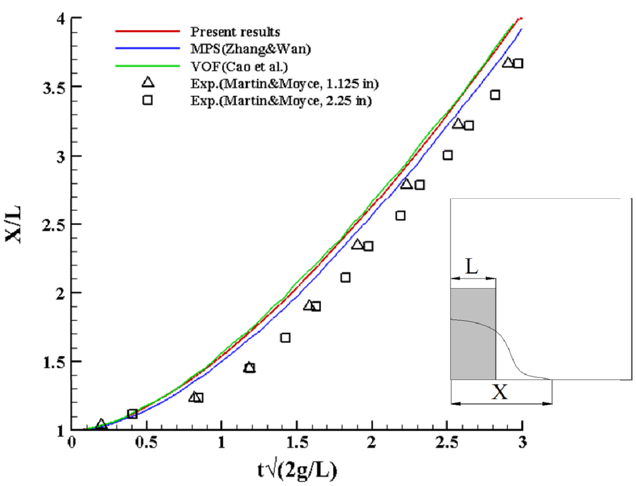


Figure 8. The propagation of water front

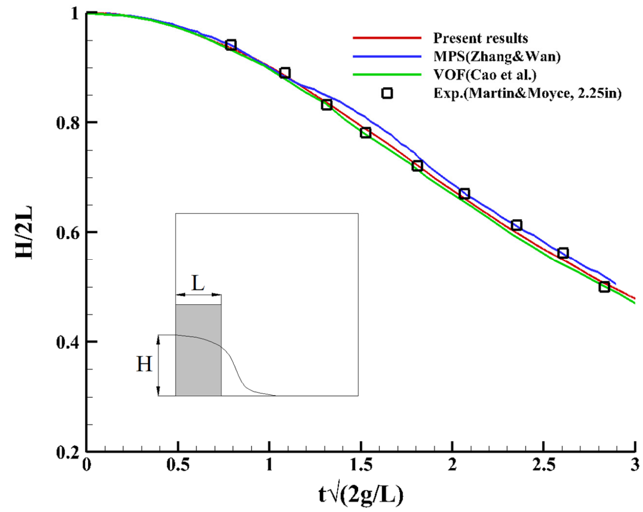


Figure 9. The height of water column

Some present simulated results are also compared to the results of other researches and the data of experiment. Figure 8 shows the propagation of water front. At the early stage of dam-break flow, the simulated wave front propagation by MPSGPU-SJTU solver is similar to the result of other researches. The collapsing water column accelerates smoothly along the bottom of tank and reaches to a stable velocity. However, the numerical propagation speed of water front is faster than the data of experiment. The height of water column by numerical researches and experiment is shown in Figure 9. The height of water column declines slowly with the process of dam-break water. And all numerical results are in good agreement with experimental data. The validity of in-house solver MPSGPU-SJTU is confirmed by these comparisons.

In order to demonstrate the effect of GPU acceleration technique, the comparison of computation time between GPU and CPU is conducted here. Another in-house solver MLParticle-SJTU valid to simulate violent flow problems in the previous works of our group is performed on high performance computing (HPC) with CPU core of Intel(R) Xeon(R) E5-2680 v2, 2.80 GHz. The computation time of one thousand steps from 0.8 s to 0.9 s are selected to compare. The detailed computation time of every step in each time iteration is listed in Table 3. And Figure 10 shows the speedup of steps 2-8 between GPU and CPU. Due to different strategies of neighbor particle searching in step 1, the calculation time of GPU solver is much less than that of CPU solver. And solving pressure Poisson equation always costs the most time in MPS method and play a decisive role in improving computational efficiency. From Table 3 and Figure 10, there is a significant reduction in the computation time of every step by GPU simulation. It may cost 14 days to simulate the whole process of dam-break flow by using CPU-based solver while only 15 hours by GPU code. The acceleration ratio between GPU and CPU is up to 22.3, which proves that GPU acceleration technique can significantly improve computational efficiency.

Table 3. The computation times of CPU and GPU

	GPU	CPU
Step 1	0.004 s	7.516 s
Step 2-3	0.181 s	3.133 s
Step 4-5	0.100 s	0.662 s
Step 6	2.940 s	62.692 s
Step 7-8	0.100 s	0.914 s
Total	3.361 s	74.917 s

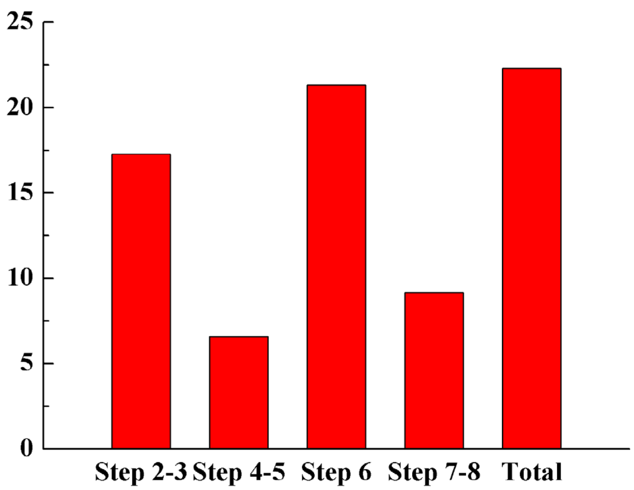


Figure 10. The speedup of GPU

CONCLUSIONS

An in-house solver MPSGPU-SJTU based on modified MPS and GPU acceleration technique is developed in this work. And this solver is used to simulate the typical violent dam-break flow problem. Comparing to other numerical calculations, the results of present work are more agreement with the experiment. The present simulation can capture nonlinear fragmentation of free surface like splashing and breaking wave, which is harder to simulate by grid method. With the increase of particles, MPSGPU-SJTU solver can model more real and complex flow field such as successive overturning waves and cavitation in fluid. The validity and accuracy of present solver is confirmed by comparing the propagation of water front and the height of water column between different studies. In addition, the computation time of GPU solver is reduced significantly and the speedup of each time iteration is up to 22.3. These results demonstrate that it would be applicable to simulate large-scale violent flow by GPU accelerated MPS method.

ACKNOWLEDGMENTS

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

[1] Dressler, R.F. Comparison of theories and experiments for the hydraulic dam-break wave [J]. Int. Assoc. Sci. Hydrology, 1954, 3(38): 319-328.

[2] Kleefsman, K.M.T., Fekken, G., Veldman, A.E.P., Iwanowski, B., Buchner, B. A Volume-of-Fluid based simulation method for wave impact problems [J]. Journal of Computational Physics, 2005, 206(1): 363-393.

[3] Greaves, D.M. Simulation of viscous water column collapse using adapting hierarchical grids [J]. International for Numerical Methods in Fluids, 2006, 50(6): 693-711.

[4] Chang, T.J, Kao, H.M., Chang, K.H., Hsu, M.H. Numerical simulation of shallow-water dam break flows in open channels using smoothed particle hydrodynamics [J]. Journal of Hydrology, 2011, 408(1-2): 78-90.

[5] Luo, M., Koh, G.G., Bai, W., Gao, M. A particle method for two-phase flows with compressible air pocket [J]. International Journal for Numerical Methods in Engineering, 2016, 108(7): 695-721.

[6] Tang, Z.Y., Zhang, Y.L., Wan, D.C. Multi-Resolution MPS Method for Free Surface Flows [J]. International Journal of Computational Methods, 2016, 13(4): 1641018-1-1641018-17.

[7] Hérault, A., Bilotta, G., Dalrymple, R.A. SPH on GPU with CUDA [J]. Journal of Hydraulic Research, 2010, 48(Extra): 74-79.

[8] Crespo, A.J.C., Domínguez, J.M., Barreiro, A., Gómez-Gesteira, M., Rogers, B.D. GPUs, a new tool of acceleration in CFD: efficiency and reliability on smoothed particle hydrodynamics methods [J]. PLoS One, 2011, 6: e20685.

[9] Wu, J.S., Li, N., Liu, W.Y., Zhang, H. GPU-based SPH modeling of flood with floating bodies in urban underground spaces[C]. 2017 SPHERIC Beijing International Workshop, Beijing, China, October 18-20, 2017: 149-156.

[10] Hori, C., Gotoh, H., Ikari, H., Khayyer, A. GPU-acceleration for moving particle semi-implicit method [J]. Computers & Fluids, 2011, 51: 174-183.

[11] Kakuda, K., Nagashima, T., Hayashi, Y., Obara, S., Toyotani, J., Katsurada, N., Higuchi, S., Matsuda, S. Particle-based fluid flow simulations on GPGPU using

CUDA. Computer Modeling in Engineering & Sciences, 2012, 88: 17-28.
[12] Li, H.Z., Zhang, Y.L., Wan, D.C. GPU Based Acceleration of MPS for 3D Free Surface Flows [C]. Proceedings of the 9th International Workshop on Ship and Marine Hydrodynamics, Glasgow, UK, 26-28 August, 2015.
[13] Zhang, Y.X., Wan, D.C. Numerical simulation of liquid sloshing in low-filling tank by MPS [J]. Journal of Hydrodynamics, 2012, 27: 101-107.
[14] Lee, B.H., Park, J.C., Kim, M.H., Hwang, S.C. Step-by-step improvement of MPS method in simulating violent free-surface motions and impact loads [J]. Computer Methods in Applied Mechanics and Engineering, 2011, 200: 1113-1125.
[15] CUDA Toolkit Documentation v8.0.61. 2017 http://docs.nvidia.com/cuda/
[16] CUSP. https://developer.nvidia.com/cusp. 2017.
[17] Martin, J.C., Moyce, W.J. An experimental study of the collapse of liquid columns on a rigid horizontal plane [J]. Philosophical Transaction of the Royal Society of London, 1952, 244(882): 312-324.
[18] Koshizuka, S., Oka, Y. Moving-particle semi-implicit method for fragmentation of incompressible fluid [J]. Nuclear Science and Engineering, 1996, 123: 421-434.
[19] Cao, H.J., Wan, D.C., Yang, C. Numerical simulation of violent flow of 3-D dam-breaking wave around square cylinder [J]. Chinese Journal of Hydrodamics, 2013, 28(4): 414-422.
[20] Zhang, Y.X., Wan, D.C. Application of MPS in 3D dam breaking flows [J]. SCIENTIA SINICA Phys, Mech & Astron, 2011, 41(2): 140-154.