11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
E. Oñate, J. Oliver and A. Huerta (Eds)

SOME MODIFICATIONS OF MPS METHOD FOR INCOMPRESSIBLE FREE SURFACE FLOW

ZHE SUN[*], KAMAL DJIDJELI[*], JING T. XING[*], FAI CHENG[†] AND ALI JAVED[*]

[*] CED/FSI Group, FEE, University of Southampton
Southampton SO17 1BJ, UK
e-mail: zs2g12@soton.ac.uk, kdd@soton.ac.uk, jtxing@soton.ac.uk, A.Javed@soton.ac.uk

[†] Lloyd's Register
London EC3M 4BS, UK
email: Fai.Cheng@lr.org

Key Words: *Moving Particle Semi-implicit (MPS) Method, Particle Method, Free Surface Problem, Sloshing, Dam-break.*

Abstract. As a Lagrangian mesh-free method, the Moving Particle Semi-implicit (MPS)[1] method is very suitable for simulating violent flows, such as breaking waves on free surface. However, despite its wide range of applicability, the original MPS algorithm suffers from some inherent difficulties in obtaining an accurate fluid pressure in both spatial and time domain. Different modifications to improve the method have been proposed [2-5] in the literature. In this paper, the authors developed a particle position shifting and collision handling technique which could effectively suppress the pressure fluctuation. In addition, a new version of "cell-link" neighbour particle searching strategy, which reduces about 7/9 (~78%) of the searching area compared with traditional "cell-link" algorithm, is proposed.

The developed MPS method with the proposed modifications has been tested on two free surface flow problems: 2D dam break and liquid sloshing. The numerical results obtained are found to be in good agreement with the available numerical and experimental results. With the proposed modifications, the stability and accuracy of the pressure field are improved in spatial and time domains.

1 INTRODUCTION

For the numerical simulation of marine engineering problems, the capturing of highly-deformed nonlinear free surface phenomena and predicting the consequent impact force to floating structures are very important and also quite challenging. In traditional mesh-based CFD approaches, the popular approaches to handle the free surface flow include VOF [6] (Volume of Fluid), LS [7] (Level Set), CIP[8] (Constrained Interpolation Profile) etc. These methods have been successfully applied to various problems. Although, it is also reported that they tend to suffer from the numerical diffusion issues[9].

On the other hand, the emerging of the so-called particle methods such as SPH (Smoothed Particle Hydrodynamics) [10] and MPS (Moving Particle Semi-implicit)[1] provide an alternative to simulate the free surface flow with Lagrangian frame. The using of meshless

approach makes it more convenient to describe the violent fluid deformation and could avoid the distortion of mesh in grid-based methods. Additionally, the Lagrangian frame will also avoid the spatial discretization of the convection term in N-S equations, which will prevent the consequent diffusion.

The original MPS method was proposed by koshizuka[1] to calculate the incompressible flow. It has been successfully applied to various problems[2, 4]. However, it suffers from some problems such as the non-physical pressure fluctuation and the falsely detected free surface particles. These defects hinder the application of MPS method to fluid-structure interaction simulations. Following the previous improving work done by other researchers[2-5], the present study would illustrate some new modifications to remedy the standard MPS method, especially in the sense of suppressing the pressure fluctuation in both time and spatial domain.

2 GOVERNING EQUATIONS

The problems investigated in this paper are all marine related violent and rapid changing physical processes, which mean the viscosity effect is quite small. As a consequence, the Lagrangian form of incompressible and inviscid Navier-Stokes equations are employed here as the governing equations of the flow.

$$\begin{aligned} \frac{D\boldsymbol{u}}{Dt} &= \boldsymbol{g} - \frac{\nabla p}{\rho} \\ \nabla \cdot \boldsymbol{u} &= 0 \end{aligned} \tag{1}$$

where, $\boldsymbol{u}$, p and ρ are the fluid velocity, pressure and density respectively, $\boldsymbol{g}$ is the acceleration of gravity.

For the free surface particles, the pressure is taken as the atmospheric pressure ($p = p_0$=0)

The solid boundary condition is described in Section 3.1.

3 THE MPS METHODOLOGY

In this section, the MPS method[1] is briefly described, containing the particle interaction model and time stepping procedure to enforce the incompressibility.

3.1 Enforcing incompressibility---Projection method

As a typical approach for the incompressible fluid computation, the two-step projection method, which is introduced by Chorin[11], is adopted here to decouple the velocity and pressure calculation:

The first step is to calculate the intermediate velocity without considering pressure, and then move the particles to the intermediate location accordingly:

$$\begin{cases} \boldsymbol{u}^* = \boldsymbol{u}^n + \Delta t \boldsymbol{g} \\ \boldsymbol{r}^* = \boldsymbol{r}^n + \boldsymbol{u}^* \Delta t \end{cases} \tag{2}$$

A pressure Poisson equation is then derived as follows to solve the pressure field:

$$\nabla^2 p^{n+1} = \rho_0 \frac{n_0 - n^*}{n_0 \Delta t^2} \tag{3}$$

here, the term n_0 and n^* are called "particle density". The superscripts 0, *n* and * indicate the initial, n^{th} and intermediate states respectively. They are proportional to the physical density and the definition is provided in the next section.

Similar to [4], the authors coupled the accumulated absolute density variation and the rate of density variation at the last time step to formulate a density error compensation term which is added in the *r.h.s* of Eqn. (3).

For the solid boundary condition, the solid particles are also included in the pressure calculation. As a consequence, its pressure will repel the fluid particle which is too close to the solid, and this would avoid the penetrating of fluid particles into solid boundary. To compensate the deficiency of neighbour particles for the "near- solid" particles when calculating n^* , two additional layers of dummy particles are placed just outside the inner solid particle layer. In standard MPS method, these particles are only involved in the particle density calculation and do not take these dummy particles into account.

For some complex geometrical boundaries, the use of dummy particles could make the particle distribution a bit tricky. In this paper, the following Neumann condition Eqn. (4) is applied on the solid boundary instead of Eqn. (3). The gradient of pressure is calculated between boundary particle and its nearest fluid particle, which means the particle density is not required to be calculated for boundary particles and consequently the layer of dummy particle is not necessary (although the support domain of fluid particles that are very close to boundary could also exceed the solid boundary layer; its particle density calculation would not be affected a lot, since the value of the weight function, i.e. Eqn. (9) drops rapidly for relatively far distance).

$$\boldsymbol{n} \cdot \nabla p^{n+1} = \rho(\boldsymbol{n} \cdot \boldsymbol{g} - \boldsymbol{n} \cdot \dot{\boldsymbol{U}}^{n+1}) \tag{4}$$

where $\dot{\boldsymbol{U}}$ is the acceleration of the boundary. When the motion of the boundary is determined by the pressure of the surrounding fluid, the acceleration of next time step $\dot{\boldsymbol{U}}^{n+1}$ is unknown since the pressure has not been solved yet. As an approximation, the value of last time step $\dot{\boldsymbol{U}}^{n}$ (or last iteration when iterative process is involved in the fluid structure interaction) is adopted instead.

For the free surface condition, in order to identify the free surface particles, all the fluid particles are examined by the following equation.

$$n_i^* < \beta n_0 \tag{5}$$

where β is a parameter slightly smaller than 1. Because of the deficiency of neighbour particles, particle density of free surface particles will drop dramatically, which means they will be selected out from this checking process.

In this study, a simplified version of the method used by C. G. Koh et al [12] is adopted. Specifically, each particle is allocated a virtual circle, If the "circle" is completely covered by its neighbours, then it is recognized as an inner fluid particle, otherwise it is a free surface particle. The circle is discretized by 360 points which locate evenly along it. If all these points are covered, the circle is then regarded as being covered.

After obtaining the pressure, the velocity and location are then updated as:

$$\begin{cases} \boldsymbol{u}^{n+1} = \boldsymbol{u}^{*} - \Delta t \cdot \dfrac{\nabla p^{n+1}}{\rho_0} \\ \boldsymbol{r}^{n+1} = \boldsymbol{r}^{n} + \Delta t \cdot \boldsymbol{u}^{n+1} \end{cases} \tag{6}$$

3.2 Particle interaction model

The gradient and Laplacian operator are discretized by a weighted average approach:

$$\nabla u(\boldsymbol{x}_i) = \frac{d}{n_0} \sum_{j \neq i}^{M} \frac{u(\boldsymbol{x}_j) - u(\boldsymbol{x}_i)}{{r_{ij}}^2} (\boldsymbol{x}_j - \boldsymbol{x}_i) w(r_{ij}) \tag{7}$$

$$\nabla^2 u(\boldsymbol{x}_i) = \frac{2d}{n_0 \lambda} \sum_{j \neq i}^{M} [u(\boldsymbol{x}_j) - u(\boldsymbol{x}_i)] w(r_{ij}) \tag{8}$$

where d is the number of space dimension, M is particles number in the support domain. $w(r_{ij})$ is the weight function.

$$w(r_{ij}) = \begin{cases} \dfrac{r_e}{r_{ij}} - 1 & 0 \leq r_{ij} \leq r_e \\ 0 & r_{ij} \geq r_e \end{cases} \tag{9}$$

where r_e is the radius of local support domain. The parameter λ and particle density n_i are both related to $w(r_{ij})$ and are defined as:

$$n_i = \sum_{k \neq i}^{M} w(r_{ik}) \tag{10}$$

$$\lambda = \frac{\sum_{j \neq i}^{M} w(r_{ij}) {r_{ij}}^2}{\sum_{j \neq i}^{M} w(r_{ij})}$$

4 THE PROPOSED MODIFICATIONS

4.1 Particle shifting and collision handling

The disorder of particle distribution is one of the main sources of pressure fluctuation suffered by particle method. Many researchers have developed different techniques to handle this problem [3,5, 13-16]. Among these improvements, rearranging the particle positions after each time step is considered to be a very effective one. It could stabilize the pressure calculation in both spatial and temporal domain[5, 15]. Here, a particle shifting method is proposed to improve the stability of computation.

After each time step, the positions of particles are slightly shifted to regularize their distribution. Actually, this technique could be regarded as a re-meshing procedure. Moreover, because the amount of shifting is very small, not mapping the value onto the new positions will not corrupt the result. The amount of shifting is decided as:

$$\delta \boldsymbol{r}_i = \sum_{j \neq i} \frac{r_0 - |r_{ij}|}{2} \cdot \frac{\boldsymbol{r}_i - \boldsymbol{r}_j}{|r_{ij}|} \quad , when \ |r_{ij}| \leq r_0 \tag{11}$$

where r_0 normally is set to be 99% of the initial particle distance.

For the free surface particles which are far away from the main fluid body, their motion will barely be affected by pressure. Under some circumstances, they may get extremely close. This unusual and "suddenly-formed" very short distance between fluid particles will cause singularity problem when solving pressure Poisson Equations. This situation will not completely be eliminated by the aforementioned particle shifting. For example, the current distance between two particles are not very small (which will not activate the particle shifting

scheme), but they have large relative velocity which means they will get very close after prediction step. Therefore, similar to [12], a simple collision handling technique is applied here. The basic idea of this approach is that the relative velocities between particles are set to be zero when they are forecasted to be closer than the threshold before the prediction step. Accordingly, before the calculation of each time step, we apply the following velocity manipulation for each fluid particle:

$$\delta \boldsymbol{v}_i = \sum_{j \neq i} -\epsilon(r_{ij})\, \boldsymbol{v}_{t_{ij}} \quad , \qquad when \quad (r_{ij} - \boldsymbol{v}_{\tau_{ij}} \Delta t) \leq r_{min} \tag{12}$$

where $v_{\tau_{\mathrm{ij}}}$ is the tangential relative velocity between particle i and j. And r_{min} is the threshold to activate the scheme. It is selected as roughly 30% of the initial particle distance in this study. Parameter ϵ depends on the property of particle j. If particle j is a fluid particle, ϵ is equal to 0.5, otherwise, i.e. if it is a solid boundary particle, ϵ is equal to 1.0. This kind of setting is to make sure that the solid particles velocity involved will not be affected while the relative velocity between its neighbour fluid particle will still be set to be zero.

4.2 Neighbour particle searching strategy

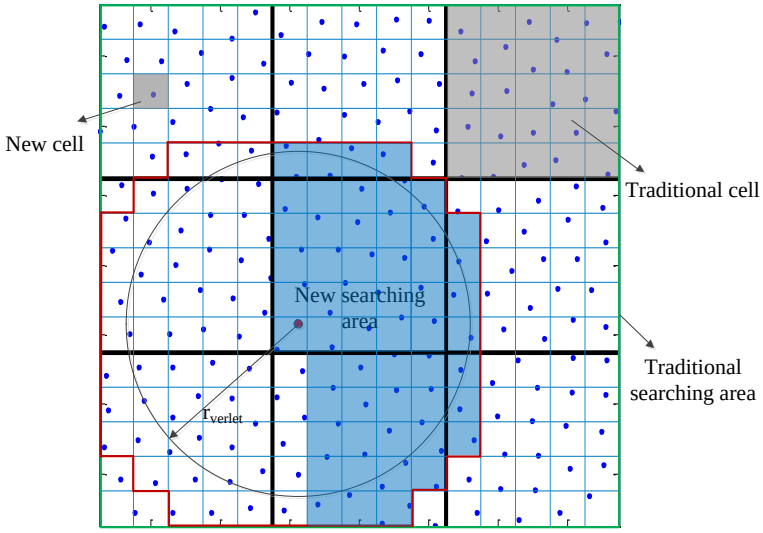

Figure 1: Demonstration of the neighbour particle searching strategy

This neighbour particle searching (which is required when discretizing gradient and Laplacian operators) could be very time-consuming if not properly conducted. Traditionally, there are two ways to accelerate the generation of the neighbour particle list[17], instead of applying the primitive "all-pair" searching strategy. They are Cell-linked algorithm and Verlet list algorithm. In cell-linked method, all particles are distributed into a set of regular square cells which cover the entire computation domain. The length of the cell side is at least the cut-off distance of supporting domain for Laplacian operator, i.e. four times of the initial particle distance. As a consequence, the neighbour searching for a particular particle could be conducted just within the surrounding cells (nine cells in 2D, i.e. area in black lines in Fig. 3). Alternatively, the Verlet list algorithm establishes a neighbour candidates list for each particle. This list contains all the particles with a larger distance from the concerned particle than the exact cut-off length of the Laplacian supporting domain. If this distance is chosen

properly, this list could be used for several time steps without the need of updating.

Actually, the combination of these two methods is the most common practice. Namely, the cell-linked method with a larger cell length (e.g. five times of initial particle distance) is first employed to generate the Verlet list, and then the neighbour particle searching is just conducted based on the list, instead of the cells, for several following time steps. An optimal length of the cell (and correspondingly the time steps during which the cell do not need to be updated) could be found based on numerical practice for a particular particle number involved in the computation.

The reason why this combination could improve the efficiency is that there are fewer candidates needed to be checked in the verlet list than in the nine cells in 2D. More specifically, the particles in the Verlet list equivalently lie within the inscribed circle of four adjacent cells (i.e. the circle in Fig. 3), which means it contains fewer particles.

In this study, in order to reduce the computation burden in the process of Verlet list generation, this principle is further explored by making the cell smaller than the traditional one which is the initial particle distance, as shown in Fig. 1. This change means the searching for Verlet list could be performed just within the red line cover area instead of the green line covered area in Fig. 1. This reduces almost 5/9 of the searching area compared with the traditional cell.

Another strategy[18] was also developed to avoid repetitive checking of pair. The core idea is that if particle *j* is in the Verlet list of particle *i*, particle *i* is obviously also in the Verlet list of particle j. Hence, the repeating of pair interaction could be avoided if the Verlet list is updated simultaneously for both of the particle pair when one of them is currently concerned as centre particle. And of course, this centre particle is then excluded during the following list generation process for the rest of particles. That means if the checking is conducted cell by cell (i.e. after the establishment of Verlet list is finished for all the particles in one cell, then move to the next cell), only the cells with higher indexes in the related neighbour cells are needed to be checked (the particles in the lower-index-cells have already been checked previously). This idea is also applicable to the new cell model aforementioned. If the cells are indexed vertically from bottom to top, the generation of Verlet list could be conducted just in the area covered by blue color in Fig. 1. This means the computation burden is further reduced by half.

Over all, the searching area required by the proposed neighbour searching strategy which consists of the new smaller cell and the non-repeating particle pair checking is only about 2/9 of the traditional cell-linked combined with Verlet list strategy.

5 NUMERICAL RESULTS

5.1 Efficiency test of neighbour searching strategy

The efficiency of the proposed cell model is tested on the 2D dam-break with different particle numbers. The neighbour particle searching process for both new and traditional strategy in one single time step consists of the following steps:

First the cell is established over the entire computation domain; then the Verlet list is generated for all the particles based on the neighbour cells. Finally, the neighbour particles list for each particle is generated twice, before and after the prediction step, by refining the searching based on the Verlet list.

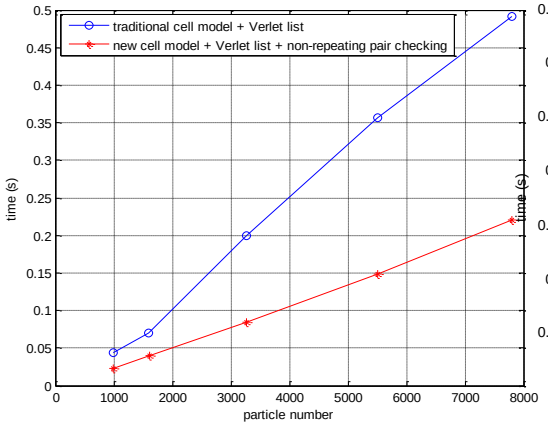

Figure 2: Computation time per time step comparison of new and traditional neighb-our search strategy

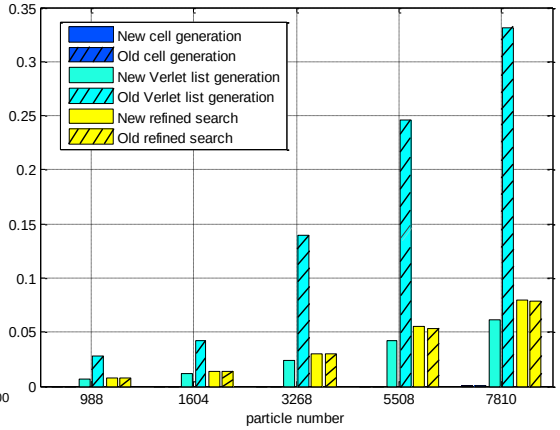

Figure 3: Proportion of time consumed for each part of the new & traditional neighbour search strategy

The simulations were conducted on a computer with Intel(R) Core(TM) i5-2400 (duo 3.1GHz) CPU, RAM 4.0 GB. As can be seen in Fig.2, using the new searching strategy, the computation time in one single time step has been reduced by 47.7%~55.3% compared with the traditional one. The efficiency of the neighbour particle searching has been improved remarkably. The proportion of the time cost by each part of the new and traditional neighbour search strategy is illustrated in Fig. 3. As is shown in Fig. 3, with the new cell model and the non-repeating pair checking technique, the time of Verlet list generation in the new strategy is only 17%~25% of the traditional one. This is consistent with the fact that the new searching area is about 22.2% (i.e. 2/9) of the traditional one. Moreover, as the cell generation part is almost neglectable in term of time consumed and the Verlet list generation part is much smaller than the refined neighbour particle searching part, the overall searching time could be further reduced by conducting two times cell updating and direct neighbour searching based on the proposed cell model and non-repeating pair checking (instead of aforementioned procedure).

5.2 Dam-break simulation

Table 1: Computation conditions for numercial simulations

	Fluid particle Number	Initial particle spacing	Max Time interval
Dam-break (without obstacle)	7200	0.005 m	0.001 s
Sloshing	2440	0.005 m	

Dam-break problem is a common testing case to verify particle method. Probably because it includes various rapid free surface deformation situations such as splashing, water re-entry etc. Additionally, it also involves the impact between moving water and the wall, which is an important phenomenon in marine engineering. In this section, the 2D Dam-break model set-up is shown in Fig. 4. The computation conditions (in all the following cases) are given in Table 1. For the time step, the CFL condition is applied with a maximum value of 0.001s. And it is selected in the same way for the following cases.

Fig. 5 shows the pressure time history at four monitor point compared between experiment

by Lobovsky *et al*. 2013[19] and MPS with the proposed modifications (with and without particle shifting). The solutions obtained using the proposed modifications (i.e. with the Neumann pressure boundary condition, additional source term in Poisson equation and particle shifting) agree well with the experiments results. Although there are still some fluctuations in the early period just after the impact, generally it is smooth enough to be used in the fluid structure interaction computation in the future research. Moreover, the results without particle shifting show a larger fluctuation, which demonstrates that the particle shifting technique is quite effective in reducing the non-physical, pressure fluctuation.

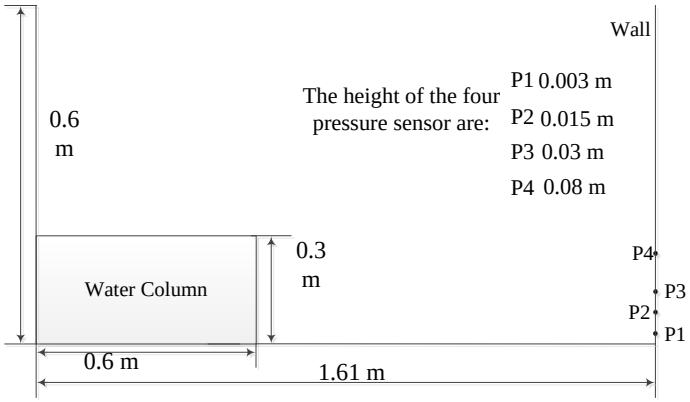

Figure 4: Sketch of the Dam-break calculation model

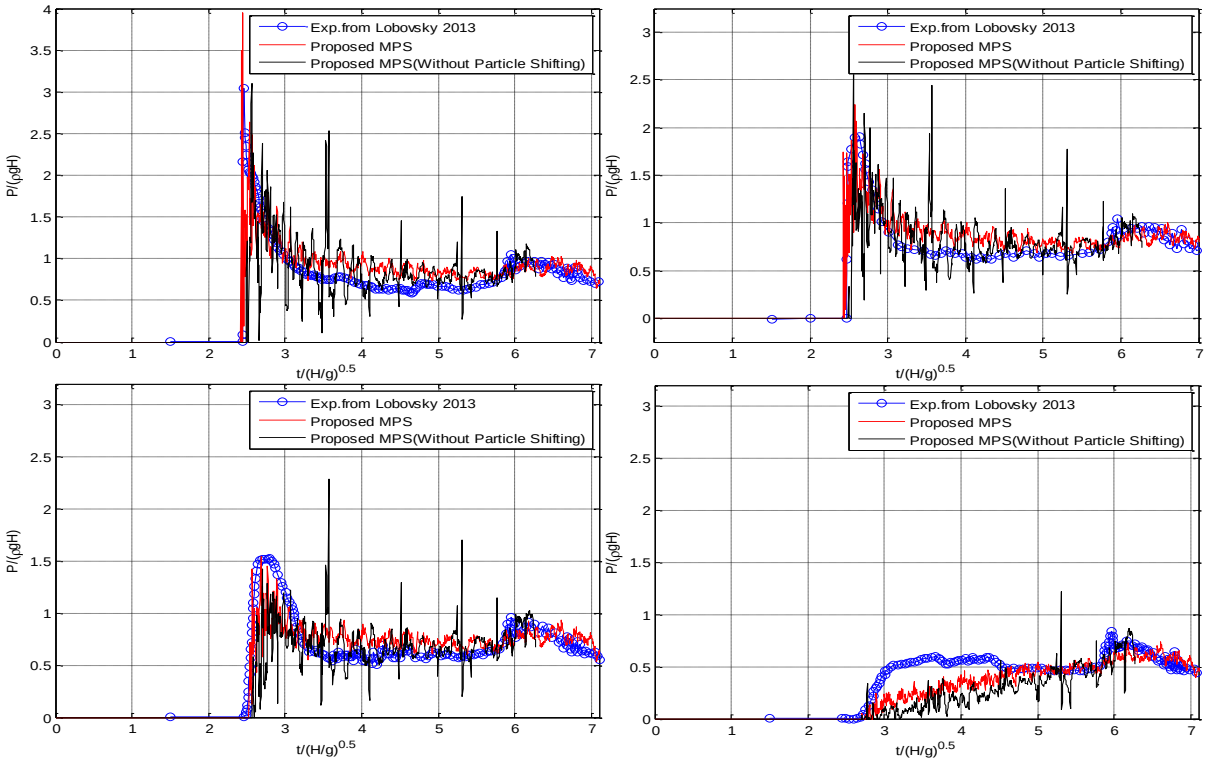

Figure 5: Pressure history monitored at P1~P4

The impact time and pressure peak value calculations match well with the experiments for points P1~P3. For sensor point P4, the peak pressure does not occur during the impact period, instead, it is caused by the falling back of the roll-up water along the wall in the later time, which is successfully captured by the computation. During the impact process, the computational pressure is found to be smaller than the experiment one at P4.

5.3 Sloshing simulation

A 2D sloshing phenomenon in partially filled tank is simulated in this section. The physical dimensions are shown in Fig.6.

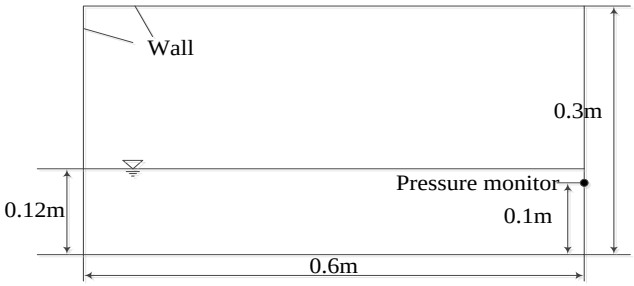

Figure 6: Sketch of the sloshing model

The tank moves sinusoidally in horizontal direction as: $X = Asin(\omega t)$, where A is the amplitude of motion and ω is the circular frequency of the excitation. In this simulation, the frequency ω= 4.8332 rad/s (period T is 1.3s) and the amplitude A= 0.05m. In order to simplify the coding, the equivalent acceleration, which is equal to the tank acceleration, is added into the right hand side of the governing equation (Eqn. (1)). And the benefit is all the boundaries remain stationary.

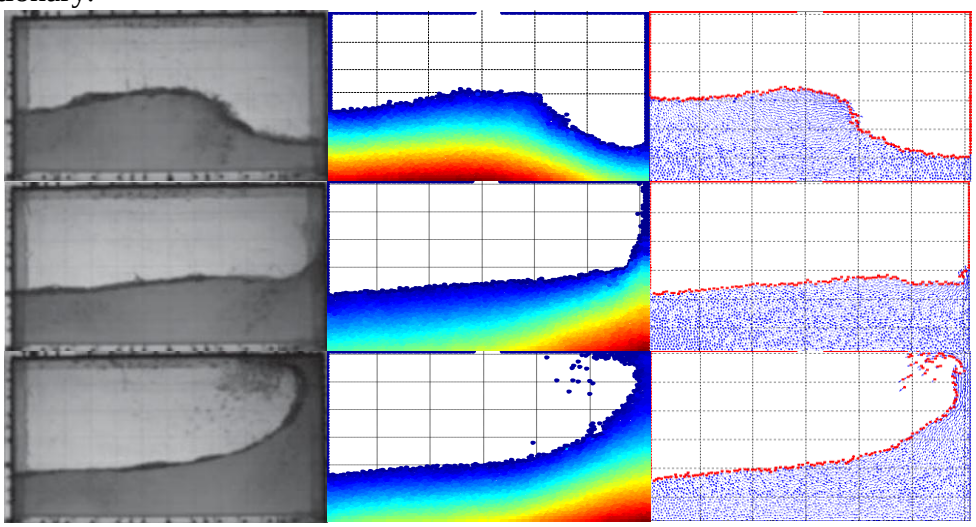

Figure 7: Comparison between experimental and numerical results at 0.1T, 0.2T and 0.3T

Fig. 7 shows the free surface profile, pressure contour of the numerical simulation and the comparison with experimental result at three time instants 0.1T, 0.2T and 0.3T (T is the period of sloshing). Fig. 8 shows the comparison of pressure history monitored at the location

shown in Fig.6 between original MPS, experiment results and Improved MPS. The original MPS results are scanned from Ref[3]. And the experiment data is extracted from the paper of Kishev *et al*[20]. It is obvious that the fluctuation of pressure in the original MPS method is too large to be used for FSI application. In contrast, the Improved MPS could successfully capture the typical pressure characters. The period of the results also match well, although a shifting manipulation (also in Ref[3, 20]) is made to align the first impulse. This could be because the starting of the measuring time in the experiment is not exactly the start of the tank motion. The peak values of each impulse are not exactly the same as those in experiment results, but the overall maximum value, which is about 7000 Pa at around 2s and 10s, is successfully captured.

6 CONCLUSIONS

- Two efficient modifications have been proposed to improve the performance of standard MPS method, including a particle shifting technique and a more efficient neighbour particle searching method.
- In order to show the effect of the aforementioned modifications, some 2D numerical examples, such as Dam-break simulation and sloshing are tested. The numerical results are also compared against numerical and experimental results from other researchers. As has been shown, the proposed modifications are found to be capable of producing smooth and stable velocity and pressure field for various free surface flow cases tested in this paper.

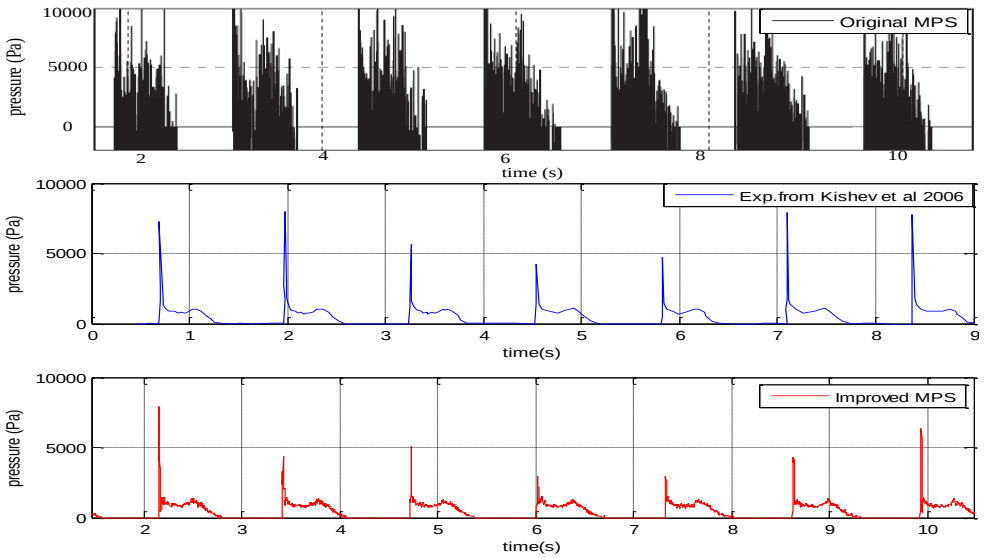

Figure 8: Pressure comparison with experiment of Kishev et al[20] and original MPS from B. H. Lee et al[3]

ACKNOWLEDGEMENT

This research is co-sponsored by Lloyds Register, University of Southampton and China Scholarship Council. The authors would like to express the sincere thanks for their supporting.

REFERENCES

[1] S. Koshizuka and Y. Oka, "Moving-Particle Semi-Implicit Method for Fragmentation of incompressible Fluid," *Nuclear Science and Engineering,* vol. 123, pp. 421-434, 1996.

[2] 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, 2009.

[3] 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, 2011.

[4] A. Khayyer and H. Gotoh, "Enhancement of performance and stability of MPS mesh-free particle method for multiphase flows characterized by high density ratios," *Journal of Computational Physics,* vol. 242, pp. 211-233, 2013.

[5] N. Tsuruta, A. Khayyer, and H. Gotoh, "A Short Note on Dynamic Stabilization of Moving Particle Semi-implicit Method," *Computers & Fluids,* 2013.

[6] C. W. Hirt and B. D. Nichols, "Volume Of Fluid (VOF) Method for the Dynamics of Free boundaries," *Journal of Computational Physics,* vol. 39, pp. 201-225, 1981.

[7] S. OSHER and R. FEDKIW, "Level Set Methods: An Overview and Some Recent Results," *Journal of Computational Physics,* vol. 169, pp. 463-502, 2001.

[8] C. H. Hu and M. Kashiwagi, "A CIP-based method for numerical simulation of violent free-surface flows," *Journal of Marine Science and Technology,* vol. 9, pp. 143-157, 2004.

[9] X. Y. Zhu, "Application of the CIP Method to Strongly Nonlinear Wave-Body Interaction Problems," PhD, Faculty of Engineering Science and Technology Department of Marine Technology, Norwegian University of Science and Technology, Norway, 2006.

[10] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics theory and application to non-spherical stars," *Monthly Notices of the Royal Astronomical Society,* vol. 181, pp. 375-389, 1977.

[11] A. J. Chorin, "A numerical method for solving incompressible viscous flow problems," *Journal of Computational Physics,* vol. 2, pp. 12-26, 1967.

[12] C. G. Koh, M. Gao, and C. Luo, "A new particle method for simulation of incompressible free surface flow problems," *International Journal for Numerical Methods in Engineering,* vol. 89, pp. 1582-1604, 2012.

[13] R. Xu, "An Improved Incompressible Smoothed Particle Hydrodynamics Method and Its Application in Free-Surface Simulations," PhD, University of Manchester, UK, 2010.

[14] M. S. Shadloo, A. Zainali, M. Yildiz, and A. Suleman, "A robust weakly compressible SPH method and its comparison with an incompressible SPH," *International Journal for Numerical Methods in Engineering,* vol. 89, pp. 939-956, 2012.

[15] S. Park and G. Jeun, "Coupling of rigid body dynamics and moving particle semi-implicit method for simulating isothermal multi-phase fluid interactions," *Computer Methods in Applied Mechanics and Engineering,* vol. 200, pp. 130-140, 2011.

[16] A. K. Chaniotis, D. Poulikakos, and P. Koumoutsakos, "Remeshed Smoothed Particle

Hydrodynamics for the Simulation of Viscous and Heat Conducting Flows," *Journal of Computational Physics,* vol. 182, pp. 67-90, 2002.

[17] S. Fanfan, "Investigation of Smoothed Particle Hydrodynamics Method for Fluid-Rigid Body Interactions," PhD, Univeristy of Southampton, 2013.

[18] A. J. C. Crespo, "Application of the Smoothed Particle Hydrodynamics model SPHysics to free-surface hydrodynamics," UNIVERSIDADE DE VIGO, 2008.

[19] L. Lobovsky, E. Botia-Vera, F. Castellana, J. Mas-Soler, and A. Souto-Iglesias, "Experimental investigation of dynamic pressure loads during dam break," *Journal of Fluids and Structures,* 2013.

[20] Z. R. Kishev, C. Hu, and M. Kashiwagi, "Numerical simulation of violent sloshing by a CIP-based method," *Journal of Marine Science and Technology,* vol. 11, pp. 111-122, 2006.