Int J Interact Des Manuf (2017) 11:633–640
DOI 10.1007/s12008-016-0338-z

ORIGINAL PAPER

Flow behaviour over a 2D body using the moving particle semi-implicit method with free surface stabilisation

Carlos A. Perez[1] · Manuel J. Garcia[1]

Received: 17 June 2016 / Accepted: 27 June 2016 / Published online: 5 July 2016

Abstract The Moving Particle Semi-implicit (MPS) method is a Lagrangian particle method based on the prediction–correction calculation of the velocity field and the Helmhotz–Hodge decomposition. Initially the predicted velocity is calculated with the viscous and external forces terms and then corrected by the gradient of the pressure which is obtained by the solution of the Poisson Pressure's equation. The MPS was developed for non-compressible bodies and it is adequate for free surface problems. However, when used to simulate fluid structure interaction problems, like ship resistance, the original formulation of the method can not accurately compute the pressure distribution over the bodies. This paper proposes a modified MPS method for modelling immerse bodies in an free surface flow. It was found that small variations in the source term of the Poisson Pressure's equation can destabilise simulations. Therefore, a reformulation of the Poisson pressure equation was developed. The results show that the proposed variation produced numerical stabilisation. The free surface particles behave in a good agreement with experimental observations. Also, although pressure fluctuations were still present, satisfactory results were obtained when comparing the drag coefficient with those reported values in the literature.

Keywords Meshless method · Particle method · MPS · Drag resistance · Free surface flow

✉ Carlos A. Perez
cperez@eafit.edu.co

Manuel J. Garcia
mgarcia@eafit.edu.co

[1] Departamento de Ingeniería Mecánica, Universidad EAFIT, Medellín, Colombia

1 Introduction

Free surface flows occur in a wide variety of physical phenomena, such as flooding, spillways, jets, gravity waves. They involve two fluids in two different states liquid and gas. Traditional methods like the Finite Volume method [3], and the finite element method [2], solving the Reynolds average Navier–Stokes' equations, have the following issues when modelling a free surface flow: As a grid is necessary to calculate spatial derivatives, mesh independence studies are necessary and time consuming, large deformations of the elements causes bad mesh qualities, numerical diffusion is originated on the advection term in the momentum equation, the initial free surface has to be defined and the mesh refined on the corresponding part of the domain. If the volume of fluid method is used [7], then at the post processing stage the free surface is not clearly defined as it changes depending on the volume fraction of air and water defined by the user. Sophisticated methods like using two different meshes have been developed to track the free surface [1,23].

Meshfree methods have emerged as a simulation alternative to mesh dependent Eulerian methods. A review of meshfree methods can be found in [21]. According to the physical modelling employed by the method, they can be classified in two classes: those based on probabilistic models such as lattice Boltzmann and molecular dynamics, and those based on deterministic models such as Smoothed Particle Hydrodynamics (SPH) and Moving Particle Semi-implicit methods [18]. The first methods represent macroscopic properties as statistical behaviour of microscopic particles. The second ones represent the fluid as a group of particle interactions. The main idea of a deterministic or Lagrangian method is to replace the fluid by a set of mathematical points named particles which follow the motion and carry fluid properties with them such as mass, momentum and pressure. The fluid dynamic equations are numerically solved for those particles.

One of the first Lagrangian mesh free particle methods was the Smoothed Particle Hydrodynamics, SPH, proposed by Gingold and Monaghan in 1977 [5] to determine the fluid dynamics in astrophysics. The term smooth refers to the use of a continuous weight function derivative to prevent large fluctuation in the calculated force [20]. Similarly, by the same time, Lucy in 1977 [22] proposed and experimented with an analogous method for gas dynamic problems in astronomy. The original method was thought for a compressible inviscid fluid and hence the reason for the derivation of the pressure from the equation of state. Monagham [25] showed its extension to simulation of free surface incompressible flows, presenting examples for the evolution of a drop, the dam break and the propagation of waves towards a beach.

The SPH method has been used in computer graphics to add realism to interactive simulations of fluids with free surface. The main focus of the interactive SPH simulation has been rendering realistic scenes rather than computing accurate engineering variables as velocities and pressures [6,26,27,38]. On the other hand, interactive CFD environments have been developed using different numerical methods as the lattice Boltzmann method [37] and the finite volume method [4]. MPS method have been applied to a variety of engineering problems as nuclear engineering [16], Environmental Hydraulics [28], bioengineering [34]. Although MPS has also been used for rendering purposes, like rendering of breaking waves [36], it was originally designed for accurate solution of incompressible fluids. This work aim to compute accurate solution of surface forces over submerged bodies.

The MPS method was originally proposed by Koshizuka and Oka [18] and it has been actively developed in the last years. For instance, Khayyer and Gotoh [9,10] proposed what they had named Corrected MPS, which consist of forcing the gradient to obey the linear momentum conservation by making forces between particles symmetric, this is, equal in magnitude and opposite in direction. Nonetheless, they have stated in 2011 [12] that non-exact momentum conservation give less numerical errors than those corresponding to approximation of the pressure gradient, and turn back to the original MPS gradient calculation with an additional Gradient Correction (GC). Further in 2013, Tsuruta, Khayyer and Gotoh [14] state clearly that previous corrections on the gradient implemented by themselves in 2011, referring to GC, do not resolve the maldistribution of particles and it requires a careful setting of calculation conditions. Thus, they proposed a Dynamically Stabilised (DS) gradient operator as a repulsive force to stabilise the calculation.

Moreover, in 2014, Hwang with Khayer, Gotoh and Park [8], did not use the DS gradient operator. Instead, they used the GC matrix for the Poisson Pressure Equation (PPE) and adopted the source term proposed by Tanaka and Masunaga [33]. This source term has two components: the divergence of the intermediate velocity plus the fraction of the particle number density deviation with a relaxation factor needed in order to stabilise the simulation. However, this source term was already criticised by Khayyer and Gotoh [12] because due to the discretisation of the source term, and the use of relaxation factors, simulations like the dam break still revealed considerable numerical noise and unrealistic flow behaviour. Hwang et al. [8] quoted Lee et al. [19] summarising that the particle method is not considered mature enough and fully reliable computational method due to non-physical pressure fluctuations and long computing time. The last issue is being compensated with nowadays parallel computing.

In the same way, Khayyer and Gotoh [12] criticised the use of unknown coefficients in the three components of the source term of the PPE proposed by Kondo and Koshizuka [15] as those coefficients were obtained by hydrostatic pressure calculations. Thus, Khayyer and Gotoh [12,14] used a three terms source for the PPE and called the Error Compensation Source term. The first term is a high source term proposed in 2009 [10] and the other two terms are dynamic instantaneous flow coefficients. However, Tamai and Koshizuka [32] reject their source term arguing that its accuracy is less than zero "since they adopted non-renormalised SPH divergence operator to approximate time derivative of particle number density." Hence, there have been different approaches to obtain a stable method. They work for specific situations if simulation parameters are carefully chosen. Therefore, there is a need for robust procedures that guarantee stability and improve the convergence of the method.

2 The MPS method

The MPS method uses the mass and momentum conservation laws. The following governing equations in Lagrangian form are considered respectively,

$$\frac{D\rho}{Dt} = 0 \tag{1}$$

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

where ρ is fluid density, t is time, $\boldsymbol{u}$ is the velocity field, P is pressure, ν is the kinematic viscosity and $\boldsymbol{g}$ is the gravity as external body force.

The domain Ω is discretized as a set of particles located at points $\boldsymbol{r}_i$. Each particle transport its mass and momentum. Let $\phi = \phi(\boldsymbol{r})$ be a scalar function defined over Ω. Then $\phi(\boldsymbol{r}_i) = \phi_i$ is the function evaluated at the particle located at $\boldsymbol{r}_i$. The gradient vector of a scalar quantity between two particles i and j can be approximated by the product of the variation of a physical quantity ϕ divided the distance times the unit direction, that is,

$$\left(\frac{\partial\phi}{\partial r}\right)_{r_j-r_i} \cong \frac{\phi_j-\phi_i}{\mid \boldsymbol{r}_j-\boldsymbol{r}_i \mid^2}(\boldsymbol{r}_j-\boldsymbol{r}_i) \tag{3}$$

In MPS, the gradient vector is approximated by a local weighted average of gradients between two particles. Therefore, for a particle located at $\boldsymbol{x}_i$ the gradient is defined as

$$\nabla\phi_i = \frac{d}{n^o}\sum_{j\neq i}\frac{\phi_j-\hat{\phi}_i}{\mid \boldsymbol{r_j}-\boldsymbol{r_i} \mid^2}(\boldsymbol{r_j}-\boldsymbol{r_i})W(\mid \boldsymbol{r_j}-\boldsymbol{r_i} \mid) \tag{4}$$

where d is the number of space dimension, $W(r)$ is the weight function and $\hat{\phi}_i$ is taken as the minimum value of ϕ_j within the interaction ratio. The latter option improves numerical stability because the forces between particles are always repulsive as $\phi_j - \hat{\phi}_i$ is always positive [17]. The Laplacian operator is modelled using a transient diffusion equation, where part of the quantity ϕ in particle i is distributed to its neighbours particles j using a weight function as follows,

$$\nabla^2\phi_i = \frac{2d}{\lambda n^o}\sum_{j\neq i}(\phi_j-\phi_i)W(\mid \boldsymbol{r_j}-\boldsymbol{r_i} \mid) \tag{5}$$

where λ is a parameter to ensure that the variance increase is equal to the analytical solution [17,18], and it is defined by

$$\lambda = \frac{\sum_{j\neq i}\mid \boldsymbol{r}_j-\boldsymbol{r}_i \mid^2 W(\mid \boldsymbol{r}_j-\boldsymbol{r}_i \mid)}{\sum_{j\neq i}W(\mid \boldsymbol{r}_j-\boldsymbol{r}_i \mid)} \tag{6}$$

Equation (5) is used to model and discretise not only the Laplacian of the velocities from the Navier stokes equation (2) but also the Poisson Pressure Equation (PPE). This PPE can be obtained from the Navier–Stokes equations and it is used to enforce continuity [3].

2.1 The weight or kernel function

Interactions between particles are limited to a finite distance with the following weight function,

$$W(r) = \begin{cases} \frac{r_e}{r} - 1 & 0 \leq r \leq r_e \\ 0 & r_e \leq r \end{cases} \tag{7}$$

where r is the distance between two particles and r_e is the radius of interaction or cut-off radius. The weight function is infinite at $r = 0$ which is good for avoiding clustering particles [17]. However, according to Kondo and Koshizuka [15], this singularity is one of the causes of the MPS instabilities and they had adopted a different weight function without singularity. Nonetheless, the original weight function is still used in several studies as those published by Shibata and Koshizuka [29], Sueyoshi et al. [31], Khayyer and Gotoh [9–14], Shibata et al. [29,30], Tanaka and Masunaga [33], Tamai and Khoshizuka [32], Hwang et al. [8].

2.2 Particle number density

The particle number density, *pnd*, of a particle at point $\boldsymbol{r}_i$ is defined in terms of the weight function as,

$$n_i = \sum_{j\neq i}W(\mid \boldsymbol{r_j}-\boldsymbol{r_i} \mid) \tag{8}$$

The fluid density is proportional to the *pnd* and the continuity equation is satisfied if the particle number density is constant. Otherwise, if the *pnd* of a particle i, n_i, is less than a constant β times the initial *pnd* n^o, then the particle is identified as one on the free surface.

$$n_i < \beta n^o \tag{9}$$

and $0 < \beta < 1$.

2.3 Pressure Poisson's equation

As a difference to other particle methods as Smooth particle hydrodynamics (SPH), the pressure is solved implicitly by means of the pressure Poisson's equation as,

$$\nabla^2 P_i = \frac{\rho}{\Delta t^2}\frac{n_i^*-n^o}{n^o} \tag{10}$$

where ρ is the fluid density, n_i^* and n^o are the *pnd* of a particle i at intermediate and the initial position respectively. Solving the Poisson pressure's equation in the original MPS method causes negative pressures as the particle number densities near the free surface is smaller than the one for inner particles. Those negative pressure values cause instability and they must be replaced by zero [15].

3 Free surface stabilisation

Stabilization was obtained via reformulating the Poisson pressure equation and by computing the pressure gradient for non free surface particles only.

3.1 Pressure Poisson's equation

The mathematical model on the calculation of the Poisson pressure's equation has been proposed similarly as in multistep methods. It consists in a linear combination of the source term with information of previous time steps. The first term is the variation of the particle number density respect to the time from a higher order accurate time differentiation of particle number density which considers the standard Weight function [10,12]. The second term, is the instantaneous time variation of the particle number density from the previous time step and the third term, is the deviation from the initial particle number density. That is,

$$\frac{\Delta t}{\rho}\nabla^2 P_i = \begin{cases} \alpha \dfrac{1}{n^o}\left(\dfrac{Dn}{Dt}\right)_i^* \\ +\beta_1 \left|\left(\dfrac{n_i^k - n^o}{n^o}\right)\right| \left[\dfrac{1}{n^o}\left(\dfrac{n_i^k - n_i^{k-1}}{\Delta t}\right)\right] \\ +\beta_2 \left|\left(\dfrac{n_i^k - n_i^{k-1}}{n^o}\right)\right| \left[\dfrac{1}{n^o}\left(\dfrac{n_i^k - n^o}{\Delta t}\right)\right], \\ \textbf{if } n_i^k > \gamma n^o. \\ \\ \alpha \dfrac{1}{n^o}\left(\dfrac{Dn}{Dt}\right)_i^* \\ +\beta_1 \left|\left(\dfrac{n_i^k - n_s^o}{n_s^o}\right)\right| \left[\dfrac{1}{n^o}\left(\dfrac{n_i^k - n_i^{k-1}}{\Delta t}\right)\right] \\ +\beta_2 \left|\left(\dfrac{n_i^k - n_i^{k-1}}{n_s^o}\right)\right| \left[\dfrac{1}{n^o}\left(\dfrac{n_i^k - n_s^o}{\Delta t}\right)\right], \\ \textbf{if } n_i^k \le \gamma n^o. \end{cases} \tag{11}$$

where α, β_1 and β_2 are relaxation factors, n^o and n_s^o are the initial particle number density of a particle inside the domain and a particle which is located on the free surface respectively, γ can be the same free surface criterion β from Eq. (9). The upper Eq. (11) is used for inner particles and the lower Eq. (11) is used for free surface particles. When the fluid is being compressed the instantaneous variation of the particle number density is positive, on the contrary when the fluid is being expanded the instantaneous variation is negative. In the same way, when the fluid has already been compressed the deviation from the initial particle number density is positive, on the contrary when the fluid has already being expanded the deviation is negative. In this way when the fluid is being expanded and it has already been compressed the source term leads just to the first term.

3.2 Gradient condition

As the particle number density near to the free surface tends to be smaller than the one of inner particles, the source term and pressure values after solving the Poisson Pressure equation are calculated as negative ones in the traditional MPS which causes instabilities. Hence, those negative pressures must set to zero in order to avoid that issue [15]. This was clearly experienced when negative pressures were allowed causing particle disorders and the simulation crashes in early stages of the computation. The following condition has been imposed over the particles to calculate the gradient,

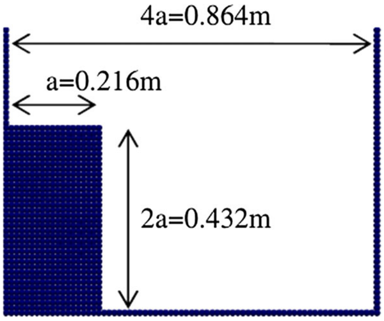


Fig. 1 Initial configuration of the dam break

$$n_i > \beta n^o \tag{12}$$

with β equals to 0.95, otherwise the gradient is zero. This means, particles near to the free surface do not have gradient pressure and it was taken only on inner particles.

4 Results

The dam break is a common benchmark used to validate computations with free surface due to its quick evolution of fluid walls in time. It consists on a fluid column initially at rest that collapses, hence the height of the column falls and the fluid spreads out. In this case, the dam break was tested for 648 fluid particles, the same number as the original MPS method proposed by Koshizuka and Oka [18]. The initial configuration is a rectangular particle array of a = 0.216 m width by twice a, 2a = 0.432 m height. Also, the domain width is 4 times a, 4a = 0.864 m, and the domain height is just important to keep particles inside it. Particles are separated by a distance from their centres by 0.012 meters and the time step used was 1×10^{-3} s. The initial configuration and the main parameters are shown in Fig. 1 and Table 1 respectively. The equations used here were the original MPS ones and the condition of zero pressure was imposed to free surface particles which reached the condition of the Eq. 9.

Figure 2 shows snapshots of particles performance at 0.5 and 1.0 seconds. Also, the advance of the right bottom end of the column of water was tracked. In order to compare with data on the literature it was convenient to present the results in units of the non-dimensional quantities. Figure 3 shows the evolution of the non-dimensional right bottom end of the column compared with experimental data from Moyce [24]. It can be observed that computational results are in good

Table 1 Calculation parameters for the dam break

Parameter	Value
Initial particle spacing	$l_o = 0.012$ m
Time step	1×10^{-3} s
Free surface criteria	$\beta = 0.97$
Fluid density	$1000\,\mathrm{kg/m^3}$
Kinematic viscosity	$1 \times 10^{-6}\,\mathrm{m^2/s}$

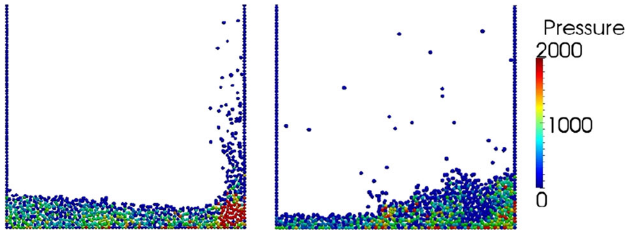


Fig. 2 Snapshots of the dam break at t = 0.5 s on the *left* and 1.0 s on the *right*

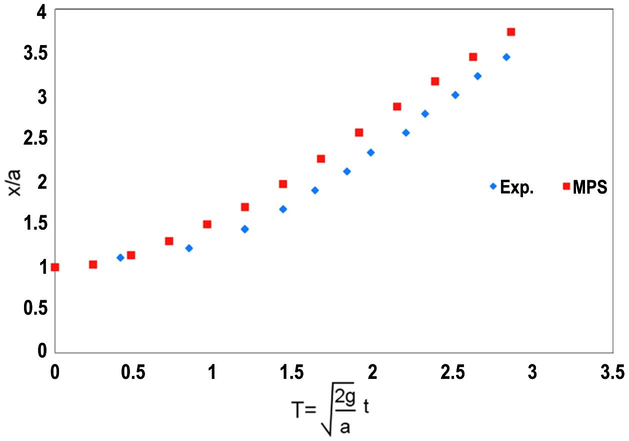


Fig. 3 Non-dimensional right edge advance versus dimensionless time for the dam break test. Comparison of MPS with experimental results. Here a is the width of the liquid column base and x is the horizontal edge of advance of the water [24]

agreement with experimental ones. Hence the MPS method was verified and validated for a closed domain.

4.1 A square cylinder in free surface flow

The MPS method with the free surface stabilisation proposed in Sect. 3 was implemented in Python 3.2. The method was tested with a 2D solid square with side length $L = 0.1$ m which was introduced into a free surface rectangular domain with dimensions 0.6 m long × 0.3 m height. A total of 2281 fluid particles were generated to represent the fluid domain, see Fig. 4. The following parameters were used in the simulation: time step = 1×10^{-4} s, initial particle distance $l_o = 8.3 \times 10^{-3}$ m, cut-off radius $r_e = 2.1 l_o$, velocity at inlet v = 1.0405 m/s, and a Reynolds number $Re \approx 100{,}000$. The solid domain was modelled as a set of rigid particles. Fluid particles were allowed to leave the domain when they reached the outlet on the left end and new fluid particles were created and introduced on the right side in order to keep a continuous fluid flow.

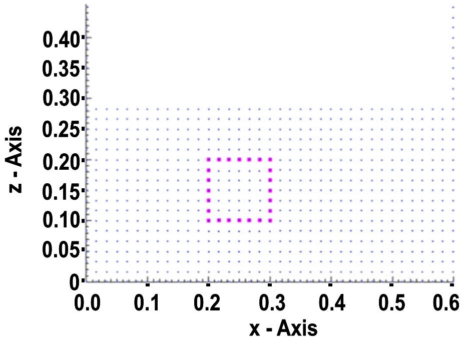


Fig. 4 Fluid particles domain. The *red particles* correspond to the boundary of the *solid square*. Dimensions are given in meters

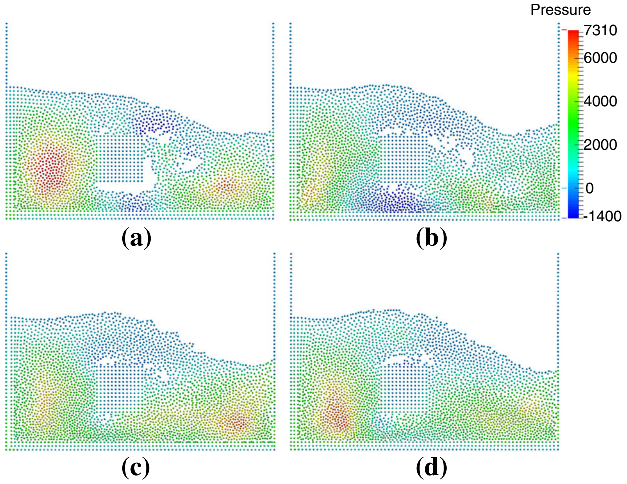


Fig. 5 Particle evolution in time of the floe around the 2d square for 2281 fluid particles. **a** 250 ms, **b** 500 ms, **c** 750 ms, **d** 1 s

Figure 5 shows the particle evolution in time of the flow over the 2D body. The colour corresponds to pressure values. The fluid particles move forwards softly from the inlet on the left hand side to the outlet on the right hand side. It can be observed well defined areas of high and low pressure. The transition between those pressure areas is done in a smooth manner. The spaces inside the fluid particles are regarded as proper interactions between the solid and the fluid. The free surface is clearly identifiable and there are not solid penetration.

In order to validate the computational data, a squared cylinder was assembled on an open channel flow. The dimension of the base of the squared cylinder was 10 cm by 10 cm, the same dimension as in the simulations. The dimension of the channel was 30 cm width by 50 cm high by 4.5 m long.

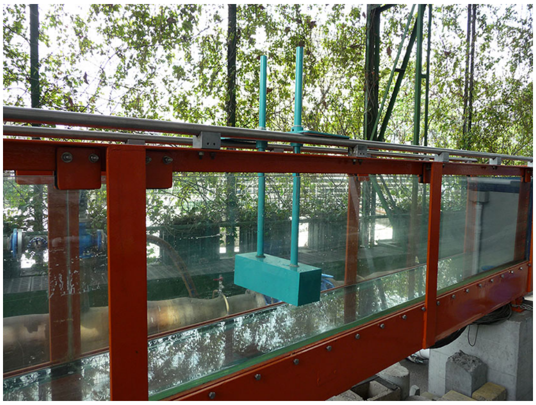


Fig. 6 Squared cylinder in the open channel used to validate the simulations

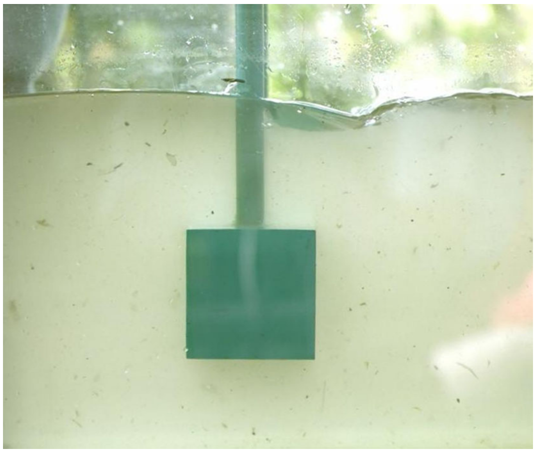
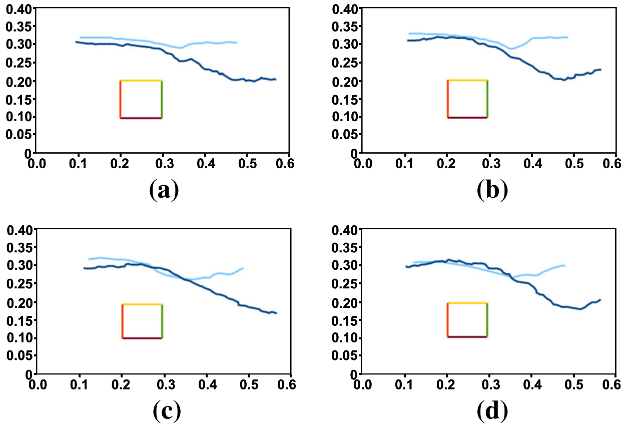


Fig. 7 Fluid flow over the squared cylinder used for experimental validation



(a) **(b)** **(c)** **(d)**

Fig. 8 Free surface comparison. The *cyan* or *light blue* is the experimental and the *navy* or *dark blue* is the computational free surface. **a** 250 ms, **b** 500 ms, **c** 750 ms, **d** 1 s

The maximum flow capacity of the channel is 28 l/s. Figure 6 shows the assembly of the squared cylinder in the open channel.

The channel was filled of water until the desired free surface level with an average velocity inlet of 1 m/s. This velocity was calculated based on the volumetric flow measurements in the pipe that provides water to the channel. Then, the squared cylinder was immersed and few seconds afterwards the evolution of the free surface was recorder with a high speed, high resolution camera. Figure 7 shows a lateral view of the fluid flow over the squared cylinder.

The experiment exhibit some differences with the simulation as the squared cylinder assembled had a couple of vertical guides to move it into place. The images obtained with the camera were processed in order to recover the free surface at different times. The results can be seen in Fig. 8. The cyan, or light blue, is the experimental free surface and the navy, or dark blue, is the computational free surface. It can be observed that the computed surface on top of the solid is in agreement with the experimental free surface. However, it can be observed that after the flow passes the square the computational free surface falls down while the experimental surface goes up again reaching the height of the initial free surface.

Differences in the results can be explained by the differences between the experimental and computational domain: the experiment was carried out in a 3D channel flow meanwhile the computational simulation was done considering just two dimensions, The computational domain was smaller than the experimental. The supporting structure of the cylinder was not taken into account in the simulation. The experimental channel was closed at the end so a drop in the level of the channel was controlled. Hence, it can be said that the second half of the free surface differ due to the outlet condition in the simulation. In the simulation, when particles reached the left hand side of the domain they were simply deleted. In the water channel, there is a static pressure of the rest of the fluid flow which causes a damming of flow downstream. The cylinder squared was restricted on the side direction so the drag resistance was not measured.

In addition, for immersed bodies with free surface, Venougopal [35] reports an experimental drag resistance coefficient of $C_d = 2.15$ for rectangular cylinders at Reynolds number in a range from 1×10^4 to 1×10^5. In this study a computational calculation of the average drag resistance coefficient was calculated according to the formula $C_d = 2F/\rho V^2 A$. For the total run simulation it gives an average value of the $C_d = 2.1$ for a Reynolds number $Re \approx 1 \times 10^5$ which is in a good agreement with the literature.

5 Conclusions

This article proposed a model for the flow around a 2D object where the pressure given by the solution of the Poisson Pressure's equation can take negative values and there is no need to modify those values to zero as it happened with the original MPS method. First, in the source term of the Poisson Pressure's equation has been conceived as a linear combination of particle number densities of previous steps and the initial setup. Also there is a different treatment if particles are on the free surface. This gives more numerical stability. The pressure gradient is only calculated for inner particles and not for particles on to the free surface. This condition improved considerably the fluid flow behaviour. Meanwhile, in the original MPS, the gradient was calculated over the whole domain and particles on the free surface tended to rise up. As fluid particles do not enter the solid and free surface particles do not rise up it can be concluded that the fluid particles flow appears to obey better a physical behaviour. The previous variants of the method reported on the literature were tested and the following issues were found: when the time step was as high as 1×10^{-3} s, solid penetration was observed and the simulation did not capture the physics observed experimentally. On the other hand, when the time step was as low as 1×10^{-4} s, some particles escape from the initial domain. This was correctly solved implementing the above-mentioned modifications and the final simulations were finished with no issues.

References

1. Caboussat, A., Picasso, M., Rappaz, J.: Numerical simulation of free surface incompressible liquid flows surrounded by compressible gas. J. Comput. Phys. **203**(2), 626–649 (2005)
2. Donea, J., Huerta, A.: Finite Element Methods for Flow Problems. Wiley, New York (2003)
3. Ferziger, J.H., Perić, M.: Computational Methods for Fluid Dynamics, 3rd edn. Springer, Berlin (2002)
4. García, M., Duque, J., Boulanger, P., Figueroa, P.: Computational steering of CFD simulations using a grid computing environment. Int. J. Interact. Des. Manuf. **9**(3), 235–245 (2015). doi:10.1007/s12008-014-0236-1
5. Gingold, R.A., Monaghan, J.J.: Smoothed particle hydrodynamics–theory and application to non-spherical stars. Mon. Not. R. Astron. Soc. **181**, 375–389 (1977)
6. Goswami, P., Schlegel, P., Solenthaler, B., Pajarola, R.: Interactive SPH simulation and rendering on the GPU. In: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 55–64. Eurographics Association (2010)
7. Hirt, C.W., Nichols, B.D.: Volume of fluid (VOF) method for the dynamics of free boundaries. J. Comput. Phys. **39**(1), 201–225 (1981)
8. Hwang, S., Khayyer, A., Gotoh, H., Park, J.: Development of a fully lagrangian MPS-base coupled method for simulation of fluid–structure interaction problems. J. Fluids Struct. **50**, 497–511 (2014)
9. Khayyer, A., Gotoh, H.: Development of CMPS method for accurate water-surface tracking in breaking waves. Coast. Eng. J. **50**(2), 179–207 (2008)
10. Khayyer, A., Gotoh, H.: Modified moving particle semi-implicit methods for the prediction of 2d wave impact pressure. Coast. Eng. **56**(4), 419–440 (2009)
11. Khayyer, A., Gotoh, H.: A high order Laplacian model for enhancement and stabilization of pressure calculation by MPS method. Appl. Ocean Res. **32**, 124–131 (2010)
12. Khayyer, A., Gotoh, H.: Enhancement of stability and accuracy of the moving particle semi-implicit method. J. Comput. Phys. **230**(8), 3093–3118 (2011)
13. Khayyer, A., Gotoh, H.: A 3D higher order Laplacian model for enhancement and stabilization of pressure calculation in 3D MPS-based simulations. Appl. Ocean Res. **37**, 120–126 (2012)
14. Khayyer, A., Gotoh, H.: Enhancement of performance and stability of MPS mesh-free particle method for multiphase flows characterized by high density ratios. J. Comput. Phys. **242**, 211–233 (2013)
15. Kondo, M., Koshizuka, S.: Improvement of stability in moving particle semi-implicit method. Int. J. Numer. Methods Fluids **65**, 638–654 (2011)
16. Koshizuka, S., Ikeda, H., Oka, Y.: Numerical analysis of fragmentation mechanisms in vapor explosions. Nucl. Eng. Des. **189**(1–3), 423–433 (1999). doi:10.1016/S0029-5493(98)00270-2. http://www.sciencedirect.com/science/article/pii/S0029549398002702
17. Koshizuka, S., Nobe, A., Oka, Y.: Numerical analysis of breaking waves using the moving particle semi-implicit method. Int. J. Numer. Methods Fluids **26**(7), 751–769 (1998)
18. Koshizuka, S., Oka, Y.: Moving particle semi-implicit method or fragmentation of incomprssible fluid. Nucl. Sci. Eng. **123**, 421–434 (1996)
19. Lee, B., Park, J., Kim, M., Hwang, S.: Step by step improvement of MPS method in simulating violent free-surface motions and impact-loads. Comput. Methods Appl. Mech. Eng. **200**(9), 1113–1125 (2011)
20. Li, S., Liu, W.: Meshfree and particle methods and their applications. Appl. Mech. Rev. **55**(1), 1–34 (2002)
21. Li, S., Liu, W.K.: Meshfree Particle Methods. Springer, Berlin (2004)
22. Lucy, L.: A numerical approach to the testing of the fission hypothesis. Astron. J. **82**(12), 1013–1024 (1977)
23. Maronnier, V., Picasso, M., Rappaz, J.: Numerical simulation of free surface flows. J. Comput. Phys. **155**(2), 439–455 (1999)
24. Martin, J.C., Moyce, W.J.: Part IV. An experimental study of the collapse of liquid columns on a rigid horizontal plane. Philos. Trans. R. Soc. Lond. A: Math. Phys. Eng. Sci. **244**(882), 312–324 (1952). doi:10.1098/rsta.1952.0006
25. Monaghan, J.J.: Simulating free surface flows with SPH. J. Comput. Phys. **110**, 399–406 (1994)
26. Müller, M., Charypar, D., Gross, M.: Particle-based fluid simulation for interactive applications. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 154–159. Eurographics Association (2003)
27. Price, D.J.: Splash: an interactive visualisation tool for smoothed particle hydrodynamics simulations. Publ. Astron. Soc. Aust. **24**(03), 159–173 (2007)
28. Shakibaeinia, A., Jin, Y.C.: A weakly compressible MPS method for modeling of open-boundary free-surface flow. Int. J. Numer. Methods Fluids **63**(10), 1208–1232 (2010)
29. Shibata, K., Koshizuka, S.: Numerical analysis of shipping water impact on a deck using a particle method. Ocean Eng. **34**, 585–593 (2007)
30. Shibata, K., Koshizuka, S., Tanizawa, K.: Three-dimensional numerical analysis of shipping water onto a moving ship using a particle method. J. Mar. Sci. Technol. **14**, 214–227 (2009)
31. Sueyoshi, M., Kashiwagi, M., Naito, S.: Numerical simulation of wave-induced nonlinear motions of a two dimensional floating body by the moving particle semi-implicit method. J. Mar. Sci. Technol. **13**, 85–94 (2008)
32. Tamai, T., Koshizuka, S.: Least squares moving particle semi-implicit method. Comput. Part. Mech. **1**(3), 277–305 (2014)

33. Tanaka, M., Masunaga, T.: Stabilization and smoothing pressure in MPS method by quasi-compressibility. J. Comput. Phys. **229**(11), 4279–4290 (2010)
34. Tsubota, K.I., Wada, S., Kamada, H., Kitagawa, Y., Lima, R., Yamaguchi, T.: A particle method for blood flow simulation: application to flowing red blood cells and platelets. J. Earth Simul. **5**, 2–7 (2006)
35. Venugopal, V., Varyanib, K.S., Barltrop, N.D.: Wave force coefficients for horizontally submerged rectangular cylinders. Ocean Eng. **30**, 1669–1704 (2006)
36. Wang, Q., Zheng, Y., Chen, C., Tadahiro, F., Chiba, N.: Efficient rendering of breaking waves using MPS method. J. Zhejiang Univ. Sci. A **7**(6), 1018–1025 (2006). doi:10.1631/jzus.2006.A1018
37. Wenisch, P., Treeck, C.V., Borrmann, A., Rank, E., Wenisch, O.: Computational steering on distributed systems: indoor comfort simulations as a case study of interactive CFD on supercomputers. Int. J. Parallel Emerg. Distrib. Syst. **22**(4), 275–291 (2007)
38. Zhang, F., Hu, L., Wu, J., Shen, X.: A SPH-based method for interactive fluids simulation on the multi-GPU. In: Proceedings of the 10th International Conference on Virtual Reality Continuum and its Applications in Industry, pp. 423–426. ACM (2011)