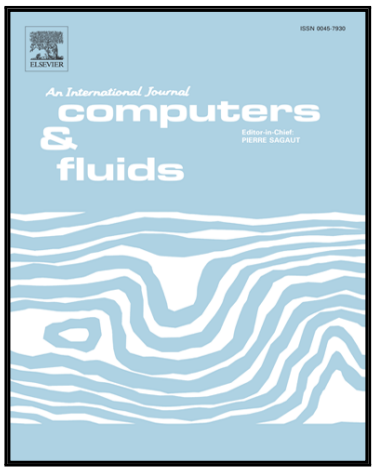

Modified mps method for the 2d fluid structure interaction problem with free surface

Zhe Sun , K. Djidjeli , Jing T. Xing , Fai cheng

PII: S0045-7930(15)00287-X
DOI: 10.1016/j.compfluid.2015.08.017
Reference: CAF 2983

To appear in: *Computers and Fluids*

Received date: 21 November 2014
Revised date: 22 July 2015
Accepted date: 14 August 2015

Please cite this article as: Zhe Sun , K. Djidjeli , Jing T. Xing , Fai cheng , Modified mps method for the 2d fluid structure interaction problem with free surface, *Computers and Fluids* (2015), doi: 10.1016/j.compfluid.2015.08.017

Highlight

- Mixed source term in the pressure Poisson equation without artificial coefficient
- New solid and free surface boundary condition
- Simple particle position shifting and collision handling
- New version of “cell-link” neighbour particle searching strategy
- Violent fluid structure interaction with rigid and flexible structures

MODIFIED MPS METHOD FOR THE 2D FLUID STRUCTURE INTERACTION PROBLEM WITH FREE SURFACE

ZHE SUN[1], K. DJIDJELI[1], JING T. XING[1] AND FAI CHENG[2]

[1] CED/FSI Group, FEE, University of Southampton
B176, Boldrewood Campus, Southampton SO16 7QF, UK
e-mail: zs2g12@soton.ac.uk, kkd@soton.ac.uk, jtxing@soton.ac.uk

[2]Strategic Research & Technology Policy Group, Lloyd's Register
B177, Boldrewood Campus, Southampton, SO16 7QF, UK
email: Fai.Cheng@lr.org

Abstract

As a Lagrangian mesh-free method, the Moving Particle Semi-implicit (MPS) 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 in the literature. In this paper, we propose the following modifications to improve the accuracy of pressure calculations and the stability of the method: i) A mixed source term in the pressure Poisson equation with no artificial term in the formulation, ii) New solid and free surface boundary handling methods, iii) Particle position shifting and collision handling, and iv) A new version of "cell-link" neighbour particle searching strategy, which reduces about 6.5/9 (~72%) of the searching area compared with traditional "cell-linked" algorithm.

The proposed modifications are verified and validated by some model free-surface flow problems, such as a two-dimensional dam break (with rigid and flexible structures on the impacting end - FSI model), liquid sloshing and ship cross section dropping problems. The numerical results obtained are found to be in good agreement with the available numerical or experimental results. With the proposed modifications, the stability and accuracy of the pressure field are improved in spatial and time domains.

Keywords: *Moving Particle Semi-implicit (MPS) Method, Free Surface Flow, Fluid Structure Interaction, Slamming, Sloshing, Dam-break Simulation.*

1. Introduction

Free surface flow problems play a vital role in the field of marine engineering. The interaction of ships and other floating structures with the waves could generate highly-deformed and complex free surface flows. And this will in turn impose impact force to the structure, which needs to be evaluated to ensure structural safety.

In the traditional mesh-based CFD approaches, the computational domain is first discretized by a set of grid, i.e. mesh; the governing equations are then solved by various numerical schemes (e.g. Finite Volume, Finite Difference or Finite Element). For problems with moving boundaries such as free surface flow, the mesh is not necessary fixed in the space. According to the movability of the mesh, the numerical methods could be classified into different categories:

First, if the mesh is fixed in space, it is an Eulerian method, in which the moving boundaries need to be captured by some additional techniques such as Volume of Fluid (VOF) [1], Level Set (LS) [2], and Constrained Interpolation Profile (CIP) [3], etc. These schemes are quite popular for problems with complex geometry fluid interfaces such as free surface flow, where moving mesh to track the interface is not affordable or not desirable or just not manageable[4]. Although, the aforementioned methods have all been successfully applied to various free surface problems, these schemes usually require solving additional set of equations to capture the interface on the non-moving mesh and consequently increase the numerical complexity. Another issue with these kinds of methods is that the theoretical sharp free surface might grow thicker gradually due to the numerical diffusion. This problem could be mitigated to some extent by introducing some special treatment such as the function

transformation mentioned in Ref [5], but the complexity is again further increased. Furthermore, as pointed in Ref [4], for the fluid-solid interface, the accuracy of the boundary layer will be limited by the resolution of the fluid mesh at the interface no matter how accurately the interface geometry is represented.

This issue could be tackled using another way of handling mesh, i.e. moving the mesh to track the boundary shape changing. Arbitrary Lagrangian Eulerian (ALE) and Space-Time[4] methods are both examples which fall into this category. In ALE, the mesh moves independently (not following the fluid motion exactly everywhere) based on the need of tracking the interface deformation or some other criteria. This mesh motion introduces an additional mesh velocity in the convective term of the governing equations. Alternatively, the Space-Time method adopts a different philosophy which is treating the time variable as an additional spatial coordinate over a generalized space-time domain[4, 6]. The governing equations are then solved over a sequence of space-time "slabs"[4]. And each slab is the slice of the space-time domain between the time level n and level n+1. The spatial mesh of the slabs would also change to accommodate the deformation of the computational domain at different time steps. Due to this special way of handling time dimension, there would be no mesh velocity in the formulation as in ALE method.

Generally, as mentioned above, the interface-capturing methods (e.g. VOF) are more suitable for complex geometry deformation such as free surface flow. On the other hand, the interface-tracking approaches (e.g. ALE or Space-Time) make it easier to maintain a higher accuracy on the moving interface when its deformation is not too dramatic such as fluid-solid interface. This is the reason that the Mixed Interface-Tracking/Interface-Capturing Technique (MITICT)[7] was introduced for the problem that involves both kinds of interfaces.

Finally, if the computational nodes follow exactly the physical fluid motion everywhere in the computational domain, i.e. the Lagrangian frame is employed, it leads to so-called

particle methods such as SPH (Smoothed Particle Hydrodynamics) [8], MPS (Moving Particle Semi-implicit) [9] and PFEM (Particle Finite Element Method) [10]. The PFEM still needs to update the finite element mesh at each time step to adjust the changing of the particle distribution, whereas in SPH and MPS, the use of fully 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 equation, and prevent numerical diffusion.

The original MPS method was proposed by Koshizuka *et al.* [9] to calculate the incompressible flow. It has been successfully applied to various problems [11, 12]. However, there are also 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 improvements done by other researchers [11-14], the present study illustrates some new modifications to remedy some of the shortcomings of the standard MPS method, especially in suppressing the pressure fluctuations in both time and spatial domain.

2. Governing equations

The problems investigated here are all marine related problems with violent and rapid changing physical processes, which means that the viscosity effect is quite small. As such, the Lagrangian form of incompressible and inviscid Navier-Stokes equations are employed here as the governing equations of the flow.

$$\frac{D\boldsymbol{u}}{Dt} = \frac{\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}}{\Delta t} = \boldsymbol{g} - \frac{\nabla p}{\rho_0} \qquad (2.1)$$
$$\nabla \cdot \boldsymbol{u} = 0$$

where $\boldsymbol{u}$, p and ρ_0 are the fluid velocity, pressure and density respectively. The superscripts k and $k+1$ refer to the k^{th} and $(k+1)^{th}$ time steps, respectively. This superscript

notation will be used in the subsequent sections. $\boldsymbol{g}$ is the vector pointing to the gravity direction, i.e. $\boldsymbol{g} = [0, -g]$, where g is the value of gravity acceleration. In Eqn. (2.1), the time derivative $\frac{D\boldsymbol{u}}{Dt}$ is discretized by a first order Lagrangian finite difference as is shown.

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

The solid boundary condition proposed here is different to the one used in the standard MPS methods, and will be illustrated in Section 4.2.

3. The standard MPS methodology

In this section, the original MPS method [9] is briefly described, including 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 [15], 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 according to this velocity:

$$\begin{cases} \boldsymbol{u}^{(*)} = \boldsymbol{u}^{(k)} + \Delta t \boldsymbol{g} \\ \boldsymbol{r}^{(*)} = \boldsymbol{r}^{(k)} + \Delta t \boldsymbol{u}^{(*)} \end{cases} \tag{3.1}$$

where $\boldsymbol{r}$ represents the location vector of particles. The superscript * indicates the value of intermediate status of a particular time step. A pressure Poisson equation is then derived as follows to solve the pressure field:

$$\nabla^2 p^{(k+1)} = \rho_0 \frac{n_0 - n^{(*)}}{n_0 \Delta t^2} \tag{3.2}$$

Here, the term n_0 and $n^{(*)}$ are called "particle density", with n_0 the desired value

corresponding to uniform particle distribution, and $n^{(*)}$ the actual value at the intermediate status. They are proportional to the physical density and the definition is provided in Section 3.2

The solid particles which lie in the support domain of the adjacent fluid particles are also included in the pressure calculation. As a consequence, its pressure will push away the fluid particles which are too close to the solid, and thus avoiding the penetrating of fluid particles into solid boundary. To compensate the deficiency of neighbour particles for the solid and "near- solid" fluid particles when calculating$n^{(*)}$, two additional layers of dummy particles are placed just outside the inner solid particle layer. These particles are only involved in the particle density calculation in standard MPS method. The Laplacian and gradient discretization do not take these dummy particles into account.

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

$$n^{(*)} < \beta n_0 \tag{3.3}$$

where β is a parameter slightly smaller than 1 (e.g. 0.97). Since within the neighbor domain of a free surface particle, there would be no fluid particles in the area which is outside the fluid domain, particle density of these free surface particles will drop dramatically. This difference of the particle density value between free surface and inner fluid particles makes the identification of the free surface particle possible.

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

$$\begin{cases} \boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(*)} - \Delta t \cdot \dfrac{\nabla p^{(k+1)}}{\rho_0} & (a) \\ \boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} + \Delta t \cdot \boldsymbol{u}^{(k+1)} & (b) \end{cases} \tag{3.4}$$

3.2 Particle interaction model

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

$$\nabla u(\boldsymbol{r}_i) = \frac{d}{n_0} \sum_{j \neq i}^{M} \frac{u(\boldsymbol{r}_j) - u(\boldsymbol{r}_i)}{{r_{ij}}^2} (\boldsymbol{r}_j - \boldsymbol{r}_i) w(r_{ij}) \tag{3.5}$$

$$\nabla^2 u(\boldsymbol{r}_i) = \frac{2d}{n_0 \lambda} \sum_{j \neq i}^{M} [u(\boldsymbol{r}_j) - u(\boldsymbol{r}_i)] w(r_{ij}) \tag{3.6}$$

where d is the number of space dimensions, and M is the particles number in the support domain. The length of the support domain is different for Laplacian and gradient operators. As suggested by [9], they are chosen to be 4 and 2.1 times of the initial particle distance r_0, respectively. It should be mentioned that the divergence operator is just replacing the scalar function in Eqn. (3.5) to a vector using dot product accordingly. $w(r_{ij})$ is the weight function and λ is a parameter related to $w(r_{ij})$, which are given by:

$$w(r_{ij}) = \begin{cases} \dfrac{r_e}{r_{ij}} - 1 & 0 \leq r_{ij} \leq r_e \\ 0 & r_{ij} \geq r_e \end{cases} \tag{3.7}$$

$$\lambda = \frac{\sum_{j \neq i}^{M} w(r_{ij}) {r_{ij}}^2}{\sum_{j \neq i}^{M} w(r_{ij})} \tag{3.8}$$

where r_e is the radius of local support domain. The particle density is defined as follows:

$$n = \sum_{j \neq i}^{M} w(r_{ij}) \tag{3.9}$$

4. Proposed modifications for MPS

4.1 Density error compensation in source term of Poisson equation

Basically, there are two forms [16] of source term in the Poisson equation, namely the Density Invariant (DI) type and Divergence-Free velocity (DF) type. The one used in standard MPS (Eqn (3.2)) is the DI type. A heuristic explanation is given below to show the difference. The momentum is reformulated and then split into two successive equations as:

$$\boldsymbol{u}^{(*)} = \boldsymbol{u}^{(k)} + \Delta t \boldsymbol{g} \tag{4.1}$$

$$\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(*)} = -\Delta t \frac{\nabla p^{(k+1)}}{\rho_0} \tag{4.2}$$

Similarly, the Lagrangian form of the continuity equation is also reformulated as:

$$\frac{\rho^{(*)} - \rho^{(k)}}{\Delta t} = -\rho_0 \nabla \cdot \boldsymbol{u}^{(*)} \tag{4.3}$$

$$\frac{\rho^{(k+1)} - \rho^{(*)}}{\Delta t} = -\rho_0 \nabla \cdot (\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(*)}) \tag{4.4}$$

Different types of Poisson equations could be obtained using different combinations of Eqns. (4.1)-(4.4).

The DI type Poisson equation could be obtained by first taking the divergence of Eqn. (4.2) and then substituting the resultant equation into Eqn.(4.4). The incompressibility condition is realized by enforcing $\rho^{(k+1)}$ to be the initial density ρ_0. The final result is Eqn. (3.2) (replacing the physical density with particle density).

On the other hand, if we take the divergence operation of Eqn. (4.2) and apply the incompressibility condition by enforcing the divergence of k+1 time step $\nabla \cdot \boldsymbol{u}^{(k+1)}$ to be zero, the DF type Poisson equation is obtained as:

$$\nabla^2 p^{(k+1)} = \rho_0 \frac{\boldsymbol{\nabla} \cdot \boldsymbol{u}^{(*)}}{\Delta t} \tag{4.5}$$

The relation between DI type (Eqn. (3.2)) and DF type (Eqn. (4.5)) could be obtained by substituting Eqn. (4.3) into Eqn. (4.5), which leads to:

$$\nabla^2 p^{(k+1)} = \frac{\rho_0 \nabla \cdot \boldsymbol{u}^{(*)}}{\Delta t} = \rho_0 \frac{n^{(k)} - n^{(*)}}{n_0 \Delta t^2} \tag{4.6}$$

The comparison between Eqn. (3.2) and Eqn. (4.6) shows that the DI form is the DF form plus the accumulated density error ($n_0 - n^{(k)}$) from the last time step. Due to the inevitable error introduced by any numerical scheme, the accumulated density error will always exist. As a consequence, the DF approach is reported to suffer from the particle clustering and void, which will result in density error accumulation and bad pressure distribution [17, 18]. On the other hand, the DI type source will lead to large density variation (probably caused by the full inclusion of accumulated density error), and consequently large pressure fluctuations in both spatial and temporal domain [19]; which reduce the stability of the pressure computation.

To tackle this problem, Hu et al. [15] proposed to use the DI and DF conditions in succession in ISPH. However this approach means to solve the Poisson equation two times, which is more time-consuming than both DI and DF methods. Xu et al. [20] have shown that the calculation time is 4-5 times larger than any of DF or DI scheme. Another strategy is to combine DF and DI in the source term [12, 14, 21] as:

$$\nabla^2 p^{(k+1)} = \frac{\rho_0 \nabla \cdot \boldsymbol{u}^{(*)}}{\Delta t} + \alpha \rho_0 \frac{n_0 - n^{(k)}}{n_0 \Delta t^2}, \tag{4.7}$$

where α is a coefficient which is normally far smaller than 1. Unlike DI scheme, this strategy attempts to take a certain amount (not all of them like in DI) of accumulated density

error into account. In most of these proposed modifications, the coefficient α needs to be calibrated based on different cases. One exception is shown in Ref [12], where the authors coupled the absolute density variation accumulated and the rate of density variation at the last time step to formulate a density error compensation term. Similarly, a new density error compensation term is proposed and added in the formulation with no artificial coefficient (which requires calibration).

The coefficient α in Poisson equation (Eqn. (4.7)) is chosen in the following way:

$$\alpha = \begin{cases} \left|\frac{n_0 - n^{(k)}}{n_0}\right| + \Delta t\left|\nabla \cdot \boldsymbol{u}^{(k)}\right| & if \quad (n_0 - n^{(k)})\nabla \cdot \boldsymbol{u}^{(k)} \geq 0 \\ \left|\frac{n_0 - n^{(k)}}{n_0}\right| & if \quad (n_0 - n^{(k)})\nabla \cdot \boldsymbol{u}^{(k)} \leq 0 \end{cases} \tag{4.8}$$

which means no artificial term appears in the process of the determination of α.

The condition $(n_0 - n^{(k)})\nabla \cdot \boldsymbol{u}_{\mathrm{n}} \geq 0$ means that the fluid is compressed, i.e. $(n_0 - n^{(k)}) \leq 0$ (or expanded, i.e. $(n_0 - n^{(k)}) \geq 0$) in the last time step ($t = k\Delta t$), and will be further compressed according to the motion trend of particles, that is, $\nabla \cdot \boldsymbol{u}^{(k)} \leq 0$ (or expanded, i.e. $\nabla \cdot \boldsymbol{u}^{(k)} \geq 0$). Under this situation, an additional term ($\Delta t|\nabla \cdot \boldsymbol{u}^{(k)}|$) is added into the coefficient to help to control further the compression (or expansion).

4.2 Boundary conditions

a) Pressure Neumann condition on solid boundaries

In most of the cases, the solid boundary in particle method will normally be handled by ghost (dummy) particle[16, 18] method , repulsive force method[22] or the combination of these two[23]. These traditional techniques require careful handling of the ghost (dummy) particle arrangement or repulsive force parameter selection according to various cases. In this study, as proposed by another particle method, MLPG_R[24] (Meshless Local Petrov Galerkin method with Rankine source solution) and also commonly used in mesh-based method, instead of Poisson equation (Eqn. (4.7)) as in the standard MPS, the following

Neumann condition (Eqn. (4.9)) is applied on the inner most layer of solid boundary. The gradient of Pressure is calculated between the current boundary particle and the nearest fluid particle (or the linear interpolation between nearest fluid particles), which will avoid the deficiency of particles within its support domain.

$$\boldsymbol{n} \cdot \nabla p^{(k+1)} = \rho_0(\boldsymbol{n} \cdot \boldsymbol{g} - \boldsymbol{n} \cdot \dot{\boldsymbol{U}}^{(k+1)}) \tag{4.9}$$

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

b) Laplacian operator compensation near solid boundary

For the fluid particles which are close to the solid boundary, the Laplacian operator needs to be modified to be consistent with the Neumann condition on solid boundary and compensate for the insufficiency of neighbour particles.

More specially, as shown in Fig. 4.1, if the virtual particle, which is along the local normal direction and towards the outside of solid boundary with a distance of r_0, is within the support domain of the fluid particle, this virtual particle will also be included in the calculation of Laplacian operator. The pressure value is derived according to Eqn. (4.9), i.e.

$$p_V = p_S + \rho_0(\boldsymbol{n} \cdot \boldsymbol{g} - \boldsymbol{n} \cdot \dot{\boldsymbol{U}})r_0 \tag{4.10}$$

where, p_V is the pressure of virtual particle, and p_S is the pressure of the corresponding

solid particle.

c) Intermediate velocity of boundary particles

The choice of intermediate velocity $\boldsymbol{u}_*$ on solid boundary will affect the accuracy of pressure which is computed from the Poisson equation (Eqn. (4.7)), since the divergence of intermediate velocity is the source term of this equation [25]. According to the idea of Ref[25], for the case of viscous flow, the non-slip condition should be applied. This means the choice of $\boldsymbol{u}^{(*)}$ should guarantee that, at k+1 time step, the fluid velocity on boundary (i.e. $\boldsymbol{u}^{(k+1)}\big|_{\partial\Omega}$) is equal to the solid body velocity on the fluid-solid interface ($\boldsymbol{u_b}^{(k+1)}$). Thus, if $\boldsymbol{u_{n+1}}$ is replaced by boundary velocity $\boldsymbol{u_b}^{(n+1)}$ in Eqn. (4.1), and using $\nabla p^{(k)}$ to approximate $\nabla p^{(k+1)}$ (since it is not known at this stage), the intermediate velocity of boundary particles is:

$$\boldsymbol{u_b}^{(*)}\big|_{\partial\Omega} = \boldsymbol{u}^{(k+1)}\big|_{\partial\Omega} = \boldsymbol{u_b}^{(k+1)} + \Delta t\frac{\nabla p^{(k)}}{\rho_0} \tag{4.11}$$

Furthermore, the above equation could be split into two components after projecting to the tangent (i.e., $\boldsymbol{\tau}$) and normal (i.e., $\boldsymbol{n}$) directions, respectively. And Eqn. (4.9) could be used to calculate the normal component of the pressure gradient. The tangent part is calculated by simple finite difference approach between its neighbour solid particles. Finally, the intermediate velocity of the boundary particles is chosen as:

$$\begin{aligned}
&\frac{\partial \boldsymbol{u_b}^{(*)}}{\partial n} = \boldsymbol{n}\cdot\boldsymbol{u}^{(k+1)}\big|_{\partial\Omega} + \frac{\Delta t}{\rho_0}\frac{\partial p^{(k)}}{\partial n} = \boldsymbol{n}\cdot\boldsymbol{u_b}^{(k+1)} + \Delta t\left(\boldsymbol{n}\cdot\boldsymbol{g} - \boldsymbol{n}\cdot\dot{\boldsymbol{U}}^{(k+1)}\right)\\
&\frac{\partial \boldsymbol{u_b}^{(*)}}{\partial \tau} = \boldsymbol{\tau}\cdot\boldsymbol{u}^{(k+1)}\big|_{\partial\Omega} + \frac{\Delta t}{\rho_0}\frac{\partial p^{(k)}}{\partial \tau} = \boldsymbol{\tau}\cdot\boldsymbol{u_b}^{(k+1)} + \frac{\Delta t}{\rho_0}\frac{\partial p^{(k)}}{\partial \tau}
\end{aligned} \tag{4.12}$$

In this study, since the viscosity is not taken into account, the free-slip boundary condition

should be applied. The only difference that should be made is the tangential part of the fluid velocity on boundary (i.e., $\boldsymbol{\tau} \cdot \boldsymbol{u}^{(k+1)}|_{\partial\Omega}$) is different to $\boldsymbol{\tau} \cdot \boldsymbol{u_b}^{(k+1)}$. So in this case the term $\boldsymbol{\tau} \cdot \boldsymbol{u_b}^{(k+1)}$ is then replaced by the velocity projection of the nearest fluid particle on the direction of $\boldsymbol{\tau}$.

d) Free surface identification

A simplified version of the method used by Koh *et al* [26] is adopted. 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 the circle. If all these points are covered, the circle is then regarded as being covered. For example, in Fig. 4.2, particle A is recognized as free surface, because the yellow points on its "circle" are not covered by its neighbors. In contrast, particle B is identified as inner fluid particle.

Finally, the use of this Neumann boundary could avoid the discretization of Laplacian operator and the calculation of particle density for the boundary particles. Therefore, the two layers of dummy particles in the standard MPS are not required. The one layer of virtual particles outside of solid boundary particles does not require any sophisticated generation procedure (simply along the normal direction of inner solid particles). It is helpful for constructing complex geometry boundary shapes. Moreover, it also provides a cover for the solid boundary particle to prevent them to be falsely recognized as free surface particles.

4.3 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 some techniques to handle this problem [13, 14, 16, 17, 27, 28]. Among these improvements, rearranging the particle

positions after each time step is considered to be a very effective approach. It could stabilise the pressure calculation in both spatial and temporal domain [13, 27]. For example, in Ref [13], a dynamic force is introduced during the evolution of the particle movement. The force is the summation of all the contributions from the neighbour particles. For each force component between the pair of the concerned particle and a neighbour particle, the principle of calculating this force is to guarantee that they will be separated and verged on each other by a distance, which is at least r_0, for the next particle configuration. However for an arbitrary particle distribution, it is obvious that each force component will affect each other, and consequently makes the final particle configuration not exactly evenly distributed. But the overall distribution is much more improved and consequently makes the computation more stable and accurate. Following similar idea, in this study, a simple particle shifting method, which also intends to avoid the improper short distance between each pair of particles, is proposed to improve the stability of computation. The difference is that in this scheme the position is manipulated directly instead of by applying an artificial force.

After each time step, the positions of particles are slightly shifted to regularize their distribution. This technique could also be regarded as a re-meshing procedure. Moreover, since the Poisson equation is derived based on the incompressibility condition (i.e. the second equation in Eqn. (2.1)), the resultant pressure would roughly keep the distance between neighboring particles to be around the same value (i.e. the initial particle distance r_0). As a consequence, the space left for this further shifting would be very small. Therefore not mapping the value onto the new positions will not corrupt the result.

The amount of shifting is chosen as:

$$\delta \boldsymbol{r}_i = \sum_{j \neq i} \frac{\bar{r}_0 - |r_{ij}|}{2} \cdot \frac{\boldsymbol{r}_i - \boldsymbol{r}_j}{|r_{ij}|} \tag{4.13}$$

$when\ |r_{ij}| \leq \bar{r}_0$

where $\bar{r}_0$ 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 motions 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 be completely 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 velocities which mean they will get very close after prediction step. As a consequence, similar to [26], 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:

$$\delta \boldsymbol{v}_i = \sum_{j \neq i} -\epsilon(r_{ij})\, \boldsymbol{v}_{\tau_{ij}}, \tag{4.14}$$

$$for\ \ (r_{ij} - \boldsymbol{v}_{\tau_{ij}} \Delta t) \leq r_{min}$$

where $\boldsymbol{v}_{\tau_{\mathrm{ij}}}$ 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, if it is a solid boundary particle, ϵ is equal to 1.0. This kind of setting is chosen to make sure that the solid particles velocity involved will not be affected

while the relative velocity between its neighbor fluid particles will still be set to be zero.

4.4 Neighbor particle searching strategy

The discretizing of gradient /divergence and Laplacian operators on the position of each particle requires the information of its neighbour particles. Because of the Lagrangian nature of particle method, all the particles are constantly moving during the computation, and thus the neighbour particle lists need to be updated after every advance of the particle distribution. This neighbour particle searching could be very time-consuming if the primitive "all-pair" searching strategy is used. In terms of computational efficiency, the state-of-the-art neighbour particle searching acceleration strategy could basically be classified into three types: (1). Verlet list method, whose algorithm complexity is O(N^2)[22]; (2). Cell-linked method and the Verlet list enhanced by cell-linked approach. The cell-linked and also the Verlet list enhanced by cell-linked methods have a linear complexity [29] i.e. O(N). This is proved by the results in Section 6.1 as well. (3). The so-called "tree algorithm", which uses hierarchically tree-structure to partition the computational domain into a sequence of squares (in 2D, for 3D, it becomes cubes) until each square contains only one particle or nothing [30]. It generally has a complexity with the order of O(Nlog(N))[30]. Furthermore, this kind of algorithm is designed to handle the situation where the particles are highly unevenly distributed, which is not the case of this study.

As a consequence, only the second type of methods i.e. the linear type is first reviewed in the following context. And then a new strategy based on the cell-linked method is proposed.

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

($4r_0$). As a consequence, the neighbour searching for a particular particle could be conducted just within the surrounding cells (nine cells in 2D, i.e., the rectangular area constrained by the yellow and green lines in Fig. 4.3).

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 (e.g., $5r_0$ or $6r_0$). Because the radius is chosen to be larger than $4r_0$, the neighbour particles will not exceed the scope of this list for the next several particle distributions, consequently this list could be used as the base pool of refined searching for several time steps without the need of updating.

The generation of the Verlet list could be accelerated by cell-linked method with the radius as the cell length instead of using "all-pair" searching. The tricky problem of Verlet list method is the choice of its radius. If it is too large, the candidates in the list might be more than those covered by the nine adjacent cells with exact $4r_0$cell length (in 2D case), which means it would be meaningless to generate Verlet list since it is more time-consuming than using the cell-linked approach directly. Actually, the circle with the radius of $6r_0$ already makes the area covered by Verlet list circle (shown in Fig. 4.3) almost the same as the nine cells with $4r_0$cell length (rectangular area limited by the yellow and green line in Fig. 4.3). The radius of $5r_0$will make the Verlet list circle smaller than the nine cells with exact $4r_0$cell length. However, if we add the time used on establishing Verlet list with $5r_0$ length cells (the green line covered area in Fig. 4.3) which contains more particles than the $4r_0$ length cells (rectangular area limited by the yellow and green line in Fig. 4.3), the total time consumed would be similar or even larger. These facts about these two approaches will be further illustrated in Section 6.1.

There is only one issue remaining, which is in cell-linked approach, after each changing of the particle distribution, all the particles need to be "re-registered" to the cells again. While

this is not required in Verlet list approach at every time when the particle configuration changes (as aforementioned, the Verlet list is valid for several time steps depending on the size of the list). This seems to be an advantage for Verlet list, however, as will be shown in Section 6.1, the time spent on particle registering is almost negligible compared to the time spent on other processes. As a consequence, the cell-linked method is overall better than Verlet list method or the combination of these two methods.

In this study, in order to further reduce the computation burden, this cell-linked principle is further explored by making the cell smaller than the traditional one, i.e. to be the initial particle distance (r_0), as shown in Fig. 4.3. This change means the searching could be performed just within the red line covered area instead of the area contained by yellow and green lines in Fig. 4.3. This reduces about 4/9 of the searching area compared with the traditional cell (with $4r_0$ cell length). It is worth to mention that the time spent on cell establishing is basically the same for the smaller and traditional cell, because in both of these situations each of the particles is only required to be checked once for registering them to a particular cell (no matter larger or smaller cells). This means the cell establishing time is only proportional to the total particle number regardless of the cell length.

Another strategy was developed to avoid repetitive checking of particle pair by Crespo [31] for the traditional cell-linked approach. The core idea is that if particle *j* is in the neighbour list of particle *i*, particle *i* is obviously also in the neighbour list of particle *j*. Hence, the repeating of pair interaction could be avoided if the neighbour list is updated simultaneously for both of the particles in the pair when one of them is currently regarded as a centre particle. And then this centre particle will be excluded during the following neighbour list generation process for the rest of particles. This means if the checking is conducted cell by cell (i.e. after the establishment of neighbour list is finished for all the particles in one cell, and then moving to the next cell), only the particles in the cells with

higher indexes in the related neighbour cells are needed to be checked (as 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 neighbour list could be conducted just in the area covered by blue color in Fig. 4.3. This means that the computation burden is further reduced by half.

To summarise, the new proposed neighbour searching strategy is essentially a further improvement of the traditional cell-linked approach. It consists of two parts, i.e. smaller (i.e. r_0) and more economical cell length and the non-repeating particle pair checking. Generally speaking, this new strategy is applicable for the improvement of traditional cell-linked model with any cell length (e.g. $4r_0$, $5r_0$or $6r_0$), and the searching area required by this new strategy is always about 2.5/9 of the corresponding traditional cell-linked model (the comparison of the blue color with the yellow and green line covered area is an example for the case of $4r_0$ length criteria).

Finally, it is worth to discuss the computational complexity of the methods mentioned in this section. For the incompressible flow studied in this paper, the particles are roughly evenly distributed (although not regular, that is the reason of introducing the particle shifting in Section 4.3), hence the number of particles that is required to be checked for neighboring particle searching of a particular particle would be proportional to the area of the searching.

As discussed above, the searching area of each particle is always the same for a particular method (e.g. the blue area for the proposed strategy and the yellow line covered area for the traditional cell-linked method in Fig. 4.3). Moreover, the cell establishing time is only proportional to the total particle number regardless of the cell length. Based on these discussions, the number of particles that needs to be searched is a constant for each particle for any method mentioned in this section (although this constant is different for different method), so the overall searching time is proportional to the overall number of particles,

which means the complexity is linear, i.e. O(*N*).

5. Fluid Structure coupling strategy

In order to verify the model, some fluid structure interaction cases with both prescribed boundary motion (sloshing) and fluid field dependent boundary motion (ship cross section dropping, breaking dam interacting with spring supported rigid wall and fixed end elastic wall) are computed. The strategy used for the fluid structure coupling algorithm is as follow:

The interaction between the structure and fluid is computed in an iterative way. The Gauss-Seidel method with Aitken relaxation approach is adopted in this study.

We suppose that all the fluid and structure variables are known at $t=t_{n-1}$. The detailed process of interaction is:

1). Predict the position, velocity and acceleration of all the points on fluid structure interface $\boldsymbol{\Gamma}_{\Sigma,0}^{n}$, $\dot{\boldsymbol{\Gamma}}_{\Sigma,0}^{n}$and $\ddot{\boldsymbol{\Gamma}}_{\Sigma,0}^{n}$ at $t=t_n$ based on the value at time step $t=t_{n-1}$.

2). Based on the updated kinetic information of interface, calculate the fluid motion at $t=t_n$, by the modified MPS method. Then, obtain the new pressure $p_{\Sigma,i}^{n}$applied on the interface for i^{th} iteration at $t=t_n$.

3). Use the new fluid pressure $p_{\Sigma,i}^{n}$ to update the structure kinetic values $\tilde{\boldsymbol{\Gamma}}_{\Sigma,i+1}^{n}$, $\dot{\tilde{\boldsymbol{\Gamma}}}_{\Sigma,i+1}^{n}$ and $\ddot{\tilde{\boldsymbol{\Gamma}}}_{\Sigma,i+1}^{n}$ of all the points on the interface.

4). Check the difference between $\tilde{\boldsymbol{\Gamma}}_{\Sigma,i+1}^{n}$, and the value from the last iteration, i.e. $\boldsymbol{\Gamma}_{\Sigma,i}^{n}$. If the convergence condition

$$\left|\tilde{\boldsymbol{\Gamma}}_{\Sigma,i+1}^{n}-\boldsymbol{\Gamma}_{\Sigma,i}^{n}\right|\leq\epsilon, \qquad (5.1)$$

is satisfied, then go to step (1) to continue the computation for the next time step ($t=t_{n+1}$).

Otherwise, correct the structure position $\boldsymbol{\Gamma}_{i+1}^{n}$ for $(i+1)^{th}$ iteration using Eqn. (5.2),

$$\boldsymbol{\Gamma}^{n}_{\Sigma,i+1} = \chi_i \widetilde{\boldsymbol{\Gamma}}^{n}_{\Sigma,i+1} + (1-\chi_i)\boldsymbol{\Gamma}^{n}_{\Sigma,i} \qquad (5.2)$$

and update the velocity $\dot{\boldsymbol{\Gamma}}^{n}_{\Sigma,i+1}$and acceleration $\ddot{\boldsymbol{\Gamma}}^{n}_{\Sigma,i+1}$ by HHT [32] method:

$$\ddot{\boldsymbol{\Gamma}}^{n}_{\Sigma,i+1} = \frac{1}{\beta\Delta t^2}\boldsymbol{\Gamma}^{n}_{\Sigma,i+1} - \left[\frac{\boldsymbol{\Gamma}^{n-1}_{\Sigma}}{\beta\Delta t^2} + \frac{\dot{\boldsymbol{\Gamma}}^{n-1}_{\Sigma}}{\beta\Delta t} + \left(\frac{1}{2\beta}-1\right)\ddot{\boldsymbol{\Gamma}}^{n-1}_{\Sigma}\right] \qquad (5.3)$$

$$\dot{\boldsymbol{\Gamma}}^{n}_{\Sigma,i+1} = \frac{\gamma}{\beta\Delta t}\boldsymbol{\Gamma}^{n}_{\Sigma,i+1} + \left(1-\frac{\gamma}{\beta}\right)\dot{\boldsymbol{\Gamma}}^{n-1}_{\Sigma} + \Delta t\left[(1-\gamma)-\gamma\left(\frac{1}{2\beta}-1\right)\right]\ddot{\boldsymbol{\Gamma}}^{n-1}_{\Sigma} - \frac{\gamma}{\beta\Delta t}\boldsymbol{\Gamma}^{n-1}_{\Sigma} \qquad (5.4)$$

where $\gamma = \frac{1-2\alpha}{2}$, $\beta = \frac{(1-\alpha)^2}{4}$, and α is chosen to be 0.05 in this study.

Using these corrected interface information, conduct $(i+1)^{th}$ iteration by going back to step (2).

In Eqn. (5.2), χ_i is the Aitken relaxation factor. Its value is calculated by the following equation[33]:

$$\chi_i = -\chi_{i-1}\frac{\Delta\boldsymbol{\Gamma}^{n}_{\Sigma,i+1}{}^{T}\left(\Delta\boldsymbol{\Gamma}^{n}_{\Sigma,i+1}-\Delta\boldsymbol{\Gamma}^{n}_{\Sigma,i}\right)}{\left(\Delta\boldsymbol{\Gamma}^{n}_{\Sigma,i+1}-\Delta\boldsymbol{\Gamma}^{n}_{\Sigma,i}\right)^{T}\left(\Delta\boldsymbol{\Gamma}^{n}_{\Sigma,i+1}-\Delta\boldsymbol{\Gamma}^{n}_{\Sigma,i}\right)} \qquad (5.5)$$

where $\Delta\boldsymbol{\Gamma}^{n}_{\Sigma,j} = \widetilde{\boldsymbol{\Gamma}}^{n}_{\Sigma,j} - \boldsymbol{\Gamma}^{n}_{\Sigma,j-1}$.

This procedure is summarized in the diagram Fig. 5.1.

6. Numerical results

6.1 Efficiency test of neighbour searching strategy

The efficiency of the new proposed and traditional neighbour particle searching strategies are tested and compared, using the 2D dam-break problem with different particle numbers.

Basically, the neighbour search requires two times neighbour searching in each time step

for the particle method, as the particle distribution will change twice, i.e. advancing particle position to the intermediate stage (Eqn. (3.1)) without pressure, and correcting them with the pressure from Poisson equation (Eqn. (3.2), (4.5) or (4.7)). The following strategies are tested for this two times neighbour searching in each single time step:

(a). Pure traditional cell-linked

Conducting the two times neighbour searching within the area contained by the yellow and green line in Fig. 4.3.

(b). New strategy

As illustrated in Section 4.4, by using the new strategy, the two times neighbour searching is only needed within the area covered by blue color in Fig. 4.3.

(c). Verlet list combined with traditional cell-linked

This strategy means we first generate a Verlet list using traditional cell-linked approach, and then conduct the refined searching within the Verlet list twice. As discussed in Section 4.4, the refined searching within the Verlet list with $6r_0$ radius will not be more efficient than directly searching within the nine $4r_0$ length cells in 2D (as is shown in Fig. 4.3). Moreover, if we add the time consumed on establishing the Verlet list using the $6r_0$ length cells, the total time would be absolutely larger than the traditional cell-linked approach with $4r_0$ length cell (this will require twice the time used in comparison to strategy (a)).

If we repeat the above analysis for the case of $5r_0$ radius, the comparison of overall time

cost with (a) is not so obvious. Hence, this situation will be tested here.

(d). Verlet list enhanced by new strategy

The radius is taken to be the same as (c), i.e. $5r_0$. The difference of the approach used in (d) as compared to the one in (c) is that the establishing of Verlet list is accelerated by the new strategy; since as mentioned in Section 4.4, the new strategy is applicable for the acceleration of cells with any length.

Fig. 6.1 shows comparisons of the time consumed in a single time step by these four neighbour searching strategies. Fig. 6.2 shows the proportion of the time cost used by each part of the four searching strategies. It is worth mentioning here that the cell generation time is the same for different size cells, as has been explained in Section 4.4. Moreover, as shown in Fig. 6.2, the time spent on cell generation is indeed almost negligible compared with other parts (as mentioned in Section 4.4). All the simulations in this paper were run on computer with Intel(R) Core(TM) i5-2400 (duo 3.1GHz) CPU, RAM 4.0 GB. The compiler used is Microsoft Visual Studio.

From Fig. 6.1, it can be seen that by using the new searching strategy, the computation time in one single time step has been reduced at least by half compared with all the other three approaches. The new searching time (red line) is only about one third of the traditional cell linked model (blue line), which is consistent with the prediction in Section 4.4. The efficiency of the neighbour particle searching has been improved remarkably.

The Verlet list combined with the traditional cell-linked (i.e. (c)) is more time-consuming than the pure traditional cell-linked approach (i.e. (d)). From Fig. 6.2, it can be seen that the average time of the two parts in (c) i.e. Verlet generation (yellow bar) and refined searching (brown bar) with this list is basically equal to that of traditional cell-linked model (blue bar).

However, strategy (c) requires two times of refined searching within the Verlet list, which makes it slower.

Also in Fig. 6.2, the comparison of time in traditional cell-linked (blue and yellow bar) and that after being accelerated by the new strategy (light blue and orange bar) proves that the new strategy will reduce the searching time to about 2.5/9, as illustrated in Section 4.4.

6.2 Dam-break simulation

Dam-break problem is a common benchmark 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, several 2D Dam-break cases are simulated, as shown in Figs. 6.3-6.6. It includes pure dam-break within a rigid rectangular tank, dam-break with an obstacle in the middle of the tank and two fluid structure interaction cases, which are dam-break with a spring supported wall and a flexible wall at the impacting end of the tank, respectively. The effectiveness of the proposed modifications (Section 4) is tested by the dam-break within a rigid rectangular tank case.

a). With rigid tank

The model set-up is shown in Fig. 6.3. The initial particle distance is 0.005m, which corresponds to the number of fluid particles of 14400 (16164 particles in total). 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. Four points on the impacting wall are selected to monitor the pressure time history.

Fig. 6.7 shows the comparison of free surface profiles and pressure contours between

experiment results by Lobovsky *et al*. 2013 [34], for the original MPS and the Improved MPS. Although both the original MPS and the Improved MPS can give consistent flow configuration at various typical time instants compared with experiment results, the smoothness of the pressure field produced by the Improved MPS is much better than the one generated by the original MPS.

In order to verify the effect of the modifications, different combinations of the modifications are tested, as shown in Table 1. The pressure time history results at P2 are depicted in Fig. 6.8. The pressure fluctuation of the original MPS is quite large and finally triggers the termination of the simulation, as shown in Fig. 6.8 (a). Fig. 6.8 (b) shows the effect of the boundary conditions. The use of Neumann type solid boundary condition and the free surface recognition method adopted by [26] could reduce the fluctuation after the initial impact period. But the pressure is still too noisy because of the DI type source. Actually the comparison between Fig. 6.8 (b) and Fig. 6.8 (c) shows clearly that the DI type tend to generate larger fluctuation compared with DF type, as discussed in Section 4.1. The density error compensation scheme could further smooth out the high frequency part in the pressure time history, as shown in Fig. 6.8 (d). Finally, the use of particle position shifting successfully eliminates the singular pressure impulse by improving the regularity of the particle distribution as shown in Fig. 6.8(e). All the modifications proposed here make the numerical pressure history to be consistent with the experiment results in a very high level.

Fig. 6.9 shows the pressure time histories monitored at the other three points, i.e. P1, P3 and P4. All the curves at the four monitor points match well with the experimental data.

In order to test the convergence property of the Improved MPS method, three different initial particle distances, which are 0.01m, 0.0075m and 0.005m, are adopted to compute the same dam-break problem aforementioned. The pressure time history at the P2 monitor point and the pressure contour of the three cases are shown in Fig. 6.10 and Fig. 6.11. The

consistency of the pressure time history for different particle distances is well maintained, and they all agree well with the experiment results. The characteristic features of the free surface profile and the pressure contour at the same instant (t=1.25s) for the three cases are also very consistent, as shown in Fig. 6.11.

The mass conservation performance of the MPS with proposed modifications is checked using the same dam-break problem with different initial particle distances i.e. 0.01m and 0.0075m. The mass conservation is measured by the total volume (area in 2D) of the fluid domain[35]. More specially, the Delaunay triangulation is first established for the particle configuration at each time step, and then the area of the fluid domain is calculated by the summation of these triangles. The mass loss percentage is shown in Fig. 6.12. As is shown in this figure, the mass loss is within 5% and the refinement of particle distribution improves the mass conservation performance. The relatively large fluctuation after t(g/h)0.5=1.5 is caused by the splashing and re-entering of the water into the main body.

b). With obstacle in the middle

The model of breaking-dam impacting with a beam in the middle of tank is simulated, as shown in Fig. 6.4. The initial particle distance is also 0.005m. The fluid particle number is 1682 (2440 particles in total).

Fig. 6.13 compares the improved MPS results (third and last column from left) with experiment (first and fourth column from left) by Koshizuka *et al*. [36] and Particle Finite Element Method (middle column) by Larese et al. [10]. The results of the improved MPS are matching well with the experimental results and the Particle Finite Element Method, except at the time instant around 0.5 s. At about this time, the experiment results reveal a phenomenon in the air phase, which is completely covered by the water jet, is trying to break out from the coating water. In the experiment this causes the water jet to be higher than that in both of the

numerical results (PFEM and Improved MPS). This disagreement is reasonable since the air phase is not considered in both of the numerical computations. When the air is compressed out from the water, e.g. in the time instant of 1.0s, the Improved MPS results agree better with experiment than the PFEM results.

c). With spring supported rigid wall

The computational model for a spring with supported rigid wall is illustrated in Fig. 6.5. The initial particle distance is chosen as 0.01m, i.e., 7200 fluid particles are involved (8652 particles in total). The motion of the rotating beam is governed by the following equation:

$$I_b\ddot{\theta}_b - \frac{1}{2}M_bL_bgsin\theta_b + K_b\theta_b = T_b \qquad (6.1)$$

where θ_b is the rotation angle of the beam, I_b is the moment of inertia with respect to rotating axis, and T_b is the torque generated by the fluid pressure. The other related parameters are chosen as: mass of the beam $M_b = 1\text{kg}$, stiffness coefficient $K_b = 1500\text{N/m}$, and length of the beam $L_b = 2\text{m}$. Fig. 6.14 shows the pressure contour at several typical time instants. The initial space between particles is 0.01m, which corresponds to a total of 7200 fluid particles. In order to compare the effect of rigid and this rotational beam, the two cases are both computed in this study. For the rigid beam case, in which iteration is not required, the computation time is about 8 CPU hours for 10s physical time under the same computer condition aforementioned.

From Fig. 6.14, it can be seen that the distribution of the pressure is quite smooth in space domain. Two major impacts are found to occur in two durations, i.e. t=0.75-2s and t=5.1-5.9s. During these two periods, the large fluid pressure which is generated from the falling of the water column pushes the beam to relatively large angles.

Fig. 6.15(a) shows the time history of the rotational angle of the beam. The two significant angle impulses are consistent with the time of the two major impacts. Except from these

violent interactions, the water pressure applied on the beam is relatively much smaller. As a consequence, the beam oscillates with a period (≈0.2s) which is very close to the natural frequency ω=33.4312rad/s (T=0.1879s), as expected.

In order to investigate the effect of the beam rotation to pressure field, the time history of the pressure monitored at the point, which is 0.16m above the right corner of the beam, is compared in Fig. 6.15(b).

As can be seen in Fig. 6.15(b), the time history during the first major impact of rotational beam case is basically the same as the one in rigid beam case. However, the peak pressure value during the second major impact is larger than the one in first major impact for the rotational case. This is probably because that at the beginning of the second impact, the beam is rebounding back (which can be seen in Fig. 6.14 at about t=5s) which means that the water front and the beam are moving towards each other. And this consequently makes this impact more vigorous than the first one. The second difference is that there is a regular oscillation after t=6s for the rotational case, this is the reflection of the beam oscillation with its near-natural frequency. The pressure fluctuations which occur at about t=3s and t=6s for both the rigid and rotational cases are caused by the isolated water particles re-entering the main fluid field near right corner, which shows that this is not due to the stability issue of the MPS solver.

d). With fixed end elastic wall

As illustrated in Fig. 6.6, the elastic wall impacted by a breaking-dam is simulated in this section. The related parameters are chosen as: Young's modulus $E = 0.2\text{GPa}$, thickness $\delta = 0.006\text{m}$, line density $m = 47.16\text{kg/m}$, and the moment of inertia of the beam cross section $I = 1.8 \times 10^{-8}\text{m}^3$. The other geometric parameters are all provided on the sketch picture in Fig. 6.6. The initial particle distance is 0.004m. Consequently 1250 fluid particles

are used in the simulation (1736 particles in total).

The beam is fixed on the bottom, and its motion is solved by normal linear Finite Element (FE) method, as in Ref [37].

Similar to the spring supported beam case, both rigid and elastic beam cases are computed and compared. The physical simulation time is 10s, which requires roughly 1 CPU hours on the same computer, as illustrated above for the rigid case.

The pressure contours and the beam deformation at some typical time instants are shown in Fig. 6.16. Smooth pressure field is successfully generated.

The trajectory of the top end of the beam, which is represented by the global X coordinate of this point, is shown in Fig. 6.17 (a). The frequency of the oscillation (≈ 0.6s) is very close to its first order natural frequency $\omega = 10.5235$ ($T = 0.5971$s). This is also consistent with the fact that the first modal shape is the dominant one as is shown in Fig. 6.16.

The pressure history monitored at $Y = 0.02$m at the right corner for both rigid and elastic wall cases are shown in Fig. 6.17 (b). For both cases, the fluctuation of pressure at the time domain is very small. And the difference between them is also negligible, which means the small elastic deformation will not greatly change the fluid motion and consequently the pressure field.

6.3 Sloshing simulation

A 2D sloshing phenomenon in partially filled tank is simulated in this section. The physical dimensions are shown in Fig. 6.18. The initial particle distance is still 0.005m, which means that 2880 fluid particles are involved (3592 particles in total).

The tank moves sinusoidally in horizontal direction as: $X = A\sin(\omega t)$, where A is the amplitude of motion and ω is the circular frequency of the excitation. In this simulation, the frequency $\omega = 4.8332$ rad/s (period T is 1.3s) and the amplitude $A = 0.05$m. 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. (2.1)). And the benefit is that all the boundaries remain stationary.

Fig. 6.19 shows the free surface profiles of both the experiment results and the Improved MPS at three representative time instants, i.e. 0.1T, 0.2T and 0.3T, where T is the period of the sloshing (1.3s). The contour is the pressure distribution. Again, since no falsely recognised free surface particles exist in the main fluid body, the pressure is very smoothly distributed. The Improved MPS shows a good agreement with experiment for the free surface profiles as well.

Fig. 6.20 shows the comparison between the original MPS, experiment results and the Improved MPS. The original MPS results are scanned from Ref [14], and the experiment data are extracted from the paper of Kishev et al. [38]. From this Figure, 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 characteristics. The period of the results also match well, although a shifting manipulation (also found in Ref [14, 38]) is made to align the first impulse. This may be due to the starting of the measuring time in the experiment, is not exactly the same as 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.

6.4 Ship cross section dropping

The ship bow section dropping has also been studied by Aarsnes [39] experimentally and Sun [40] numerically. The computational model is depicted in Fig. 6.21.

Two scenarios are investigated here: the dropping heights of 0.02m and 0.118m. The corresponding entry velocities are 0.61m/s and 1.48m/s, respectively. Except from the shape of the section, other experimental set-up is the same as the one described in wedge dropping case. The total weight of the falling rig is 261kg.

The simulation is conducted in a tank with the length of 2.4m and water depth of 1m. As in the above sections, the CFL condition is applied to determine the time step. The initial distance between particles is 0.005m, which corresponds to 96000 fluid particles involved in the simulation. Here, the same setting as in the wedge dropping case is used. At the beginning of the simulation, the wedge (ship section) is placed just above the water with a distance (from the apex of the wedge to the calm water surface) of the initial particle distance i.e. 0.005m with the aforementioned initial velocity. Because of the symmetrical shape of the wedge and for programming convenience, the horizontal motion of the wedge is constrained. The physical simulation time is 0.12s to 0.18s, which takes roughly 8 CPU hours.

The velocity and vertical forces are compared in Figs. 6.22-6.23. The results of the Improved MPS are generally in good agreement with experimental results[39]; and very close to the BEM results[40]. However, it can be noticed that after about 0.13s in scenario 1 (entry speed 0.61m/s) or 0.085s in scenario 2 (entry speed 1.48m/s), the Improved MPS and BEM results start to deviate from the experimental one, this may also be explained by the same reason as with the wedge dropping case.

In order to illustrate the space distribution of pressure and velocity, the pressure and velocity contours of the fluid fields for scenario 1 (entry speed 0.61m/s) are shown in Fig. 6.24. The three states selected here are representative time instants when the water reaches the characteristic positions. The pressure and velocity fields obtained are all quite smooth.

7. Conclusion

The particle method MPS is suitable for problems with rapidly changing boundaries/flow interfaces, such as free surface flow. The Lagrangian way of moving the computational points (i.e. particles) makes the numerical domain automatically match the deformation of the physical domain. However, this feature also brings problems such as the disordered particles distribution, which can lead to inaccurate results. In order to remedy this problem, several

modifications have been proposed in this paper to improve the performance of standard MPS method, including density error compensation in source term of Poisson equation, new type of solid and free surface boundary conditions, particle shifting technique and a more efficient neighbour particle searching method.

To test these modifications, some 2D numerical examples such as Dam-break simulation (interacted with various structures), sloshing and ship cross section dropping are simulated using the MPS method with the proposed modification. The numerical results are compared with the existing numerical or experimental results, and are found to be in good agreement. The test results show that the proposed modifications are capable of producing smooth and stable velocity and pressure field for various free surface flow problems.

Acknowledgement

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

References

[1] 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.
[2] S. OSHER and R. FEDKIW, "Level Set Methods: An Overview and Some Recent Results," *Journal of Computational Physics,* vol. 169, pp. 463-502, 2001.
[3] 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.
[4] Y. Bazilevs, K. Takizawa, and T. E. Tezduyar, *Computational Fluid-Structure Interaction: Methods and Applications*: John Wiley & Sons, Ltd, 2013.
[5] 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.
[6] M. Neumüller, *Sapce-time method: Fast Solvers and Applications* vol. 20, 2013.
[7] T. E. Tezduyar, "Finite Element methods for flow problems with moving boundaries and interfaces," *Archives of Computational Methods in Engineering,* vol. 8, pp. 83-130, 2001.

[8] 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.

[9] 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.

[10] A. Larese, R. Rossi, E. Oñate, and S. R. Idelsohn, "Validation of the particle finite element method (PFEM) for simulation of free surface flows," *Engineering Computations*, vol. 25, pp. 385-425, 2008.

[11] 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.

[12] 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.

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

[14] 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.

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

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

[17] 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.

[18] S. Shao and E. Y. M. Lo, "Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface," *Advances in Water Resources*, vol. 26, pp. 787-800, 2003.

[19] X. Y. Hu and N. A. Adams, "An incompressible multi-phase SPH method," *Journal of Computational Physics*, vol. 227, pp. 264-278, 2007.

[20] R. Xu, P. Stansby, and D. Laurence, "Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach," *Journal of Computational Physics*, vol. 228, pp. 6703-6725, 2009.

[21] 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, 2011.

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

[23] J. R. Shao, H. Q. Li, G. R. Liu, and M. B. Liu, "An improved SPH method for modeling liquid sloshing dynamics," *Computers & Structures*, vol. 100-101, pp. 18-26, 2012.

[24] Q.W.Ma, "MLPG Method Based on Rankine Source Solution for Simulating Nonlinear Water Waves," *CMES*, vol. 9, pp. 193-209, 2005.

[25] D. L. Brown, R. Cortez, and M. L. Minion, "Accurate Projection Methods for the Incompressible Navier–Stokes Equations," *Journal of Computational Physics*, vol. 168, pp. 464-499, 2001.

[26] 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.

[27] 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.

[28] 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.

[29] S. d. Chowdhury, "SPH Simulation of nonlinear water waves," PhD, DEPARTMENT OF OCEAN ENGINEERING, INDIAN INSTITUTE OF TECHNOLOGY MADRAS, 2014.

[30] V. Springel, N. Yoshida, and S. D. M. White, "GADGET: a code for collisionless and gasdynamical cosmological simulations," *New Astronomy*, vol. 6, pp. 79-117, 2001.

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

[32] H.Hilber, T.Hughes, and R.Taylor, "Improved numerical dissipation for time integration algorithms in structural dynamics," *Earthquake Engineering & Structural Dynamics*, vol. 5, pp. 283-292, 1977.

[33] U. Küttler and W. A. Wall, "Fixed-point fluid–structure interaction solvers with dynamic relaxation," *Computational Mechanics*, vol. 43, pp. 61-72, 2008.

[34] 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.

[35] P. Ryzhakov, E. Oñate, R. Rossi, and S. Idelsohn, "Improving mass conservation in simulation of incompressible flows," *International Journal for Numerical Methods in Engineering*, vol. 90, pp. 1435-1451, 2012.

[36] S. Koshizuka, H. Tamako, and Y. Oka, "A particle method for incompressible viscous flow with fluid fragmentation," *Computational Fluid Dynamic Journal*, vol. 4, pp. 29-46, 1995.

[37] V. Sriram and Ma.Q.W., "Improved MLPG_R method for simulating 2D interaction between violent waves and elastic structures," *Journal of Computational Physics*, vol. 231, pp. 7650-7670, 2012.

[38] 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.

[39] J. V. Aarsnes, "Drop test with ship sections – effect of roll angle," Norwegian Marine Technology Research Institute1996.

[40] H. Sun, "A Boundary Element Method Applied to Strongly Nonlinear Wave-Body Interaction Problems," PhD, Norwegian University of Science and Technology, 2007.

Table 1. The models tested in the Dam-break simulation

	Source term	Boundary condition	Particle shifting
Original MPS	Density invariant type (DI)	Traditional MPS type	No
Model A	Density invariant type (DI)	New proposed type in section 4.2	No
Model B	Divergence free type (DF)	New proposed type in section 4.2	No
Model C	Density error compensation scheme in section 4.1	New proposed type in section 4.2	No
Improved MPS	Density error compensation scheme in section 4.1	New proposed type in section 4.2	Yes

g. 4.1 laplace compensate

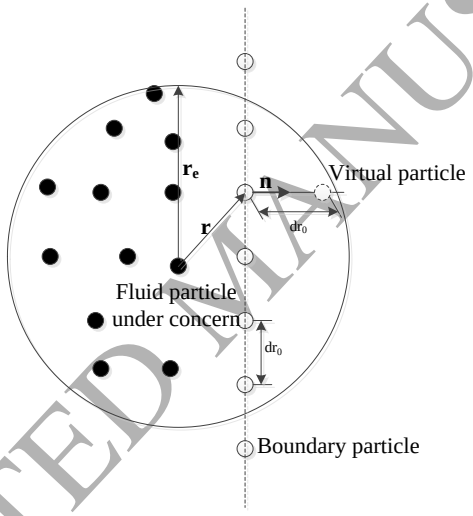

Fig. 4.1: Demonstration of new pressure boundary condition

g. 4.2 fs identification

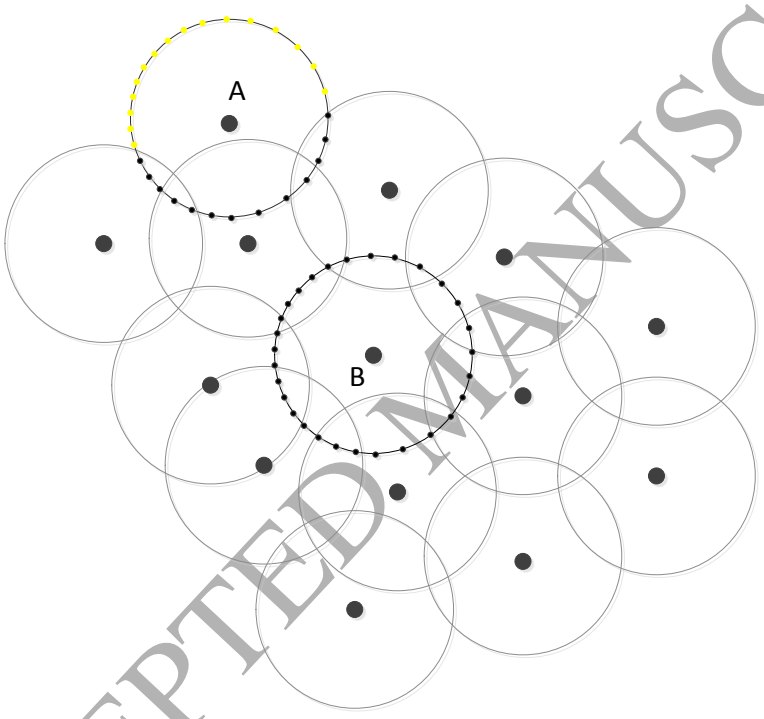

Fig. 4.2: Demonstration of new free surface particle recognition method

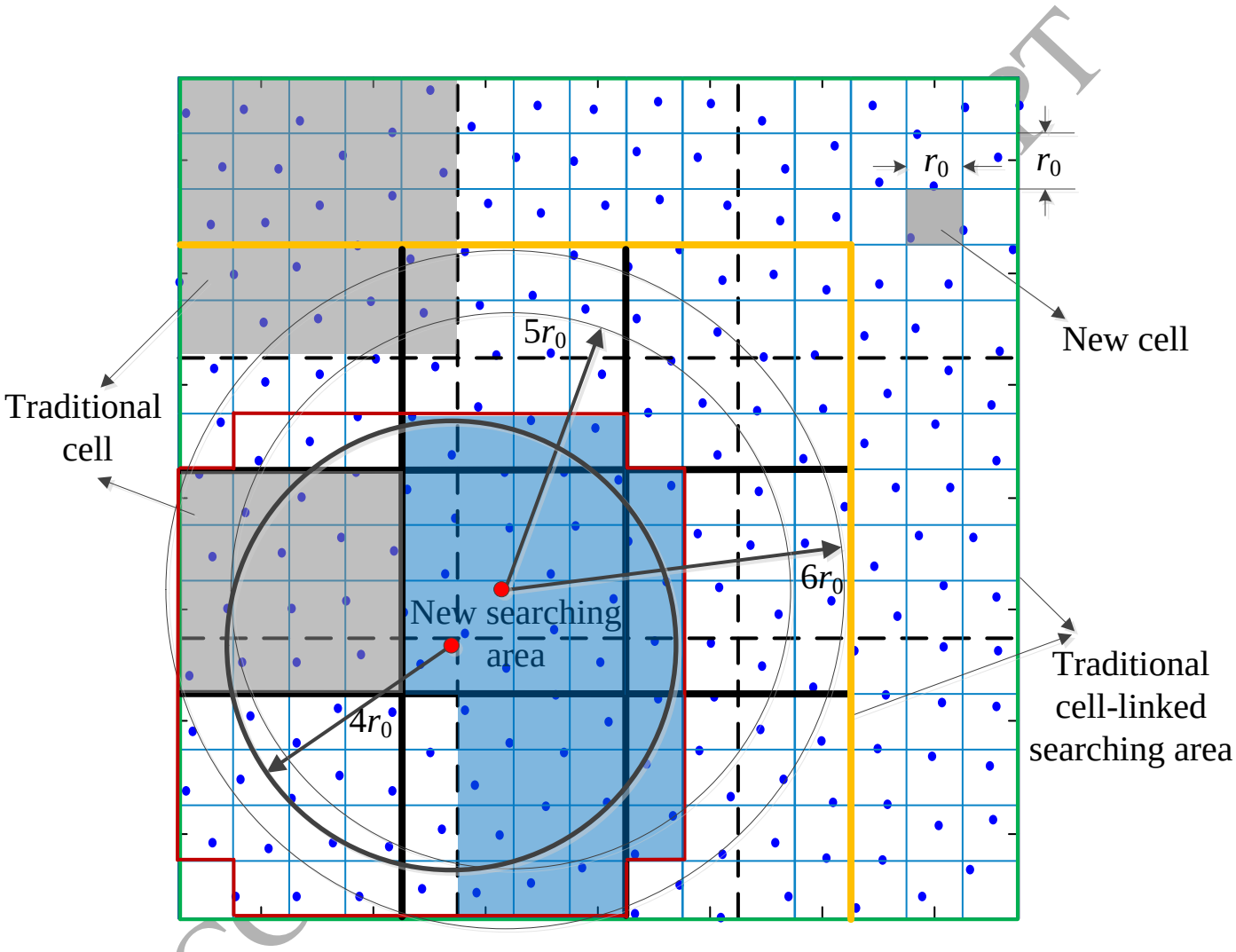

Fig 4.3: Demonstration of the neighbour particle searching strategy

g. 5.1 Flow chart of fsi

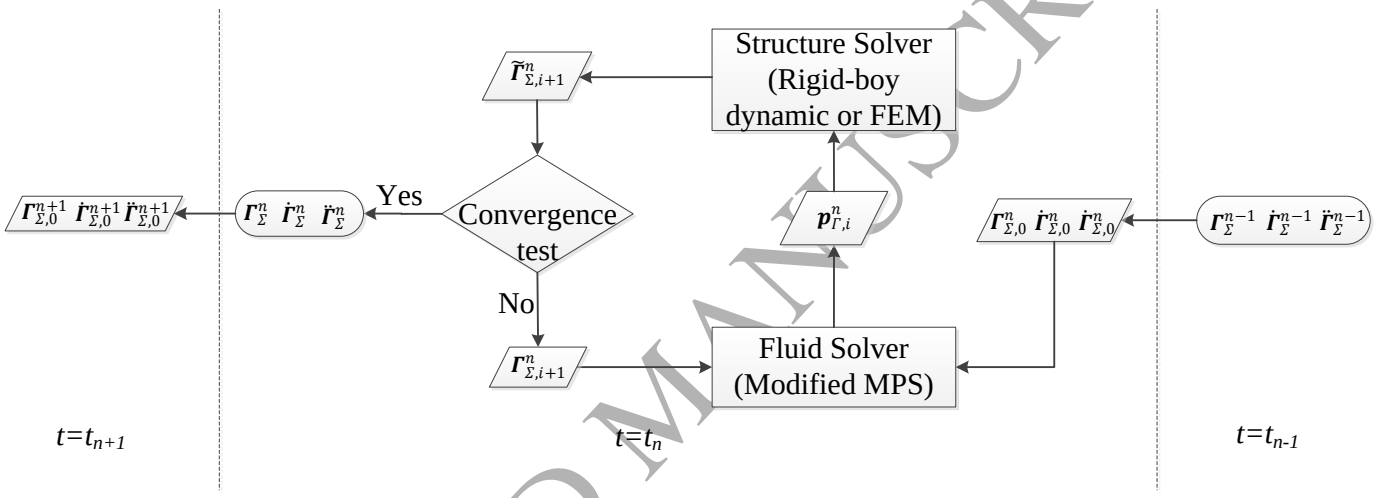

Fig. 5.1: Flow chart of the iterative process of fluid structure interaction

t neigh_search_comparison_4cases_more_ptcl

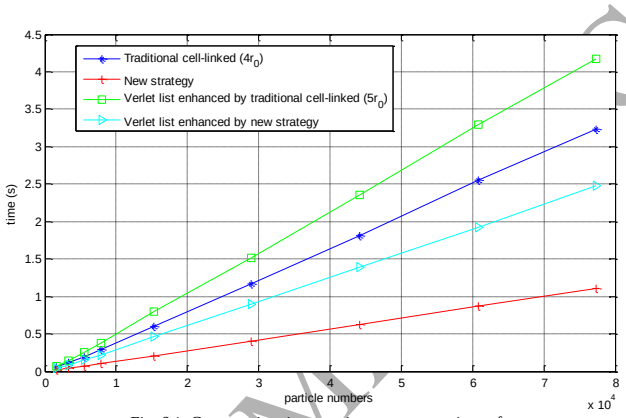

Fig. 6.1: Computation time per time step comparison of new and traditional neighbour search strategies

2 neigh_proportion_compare

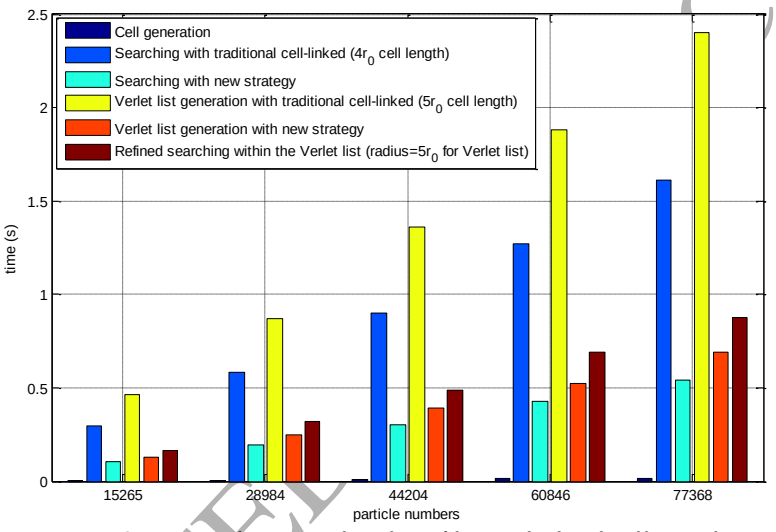

Fig. 6.2: Proportion of time consumed in each part of the new and traditional neighbour search strategies

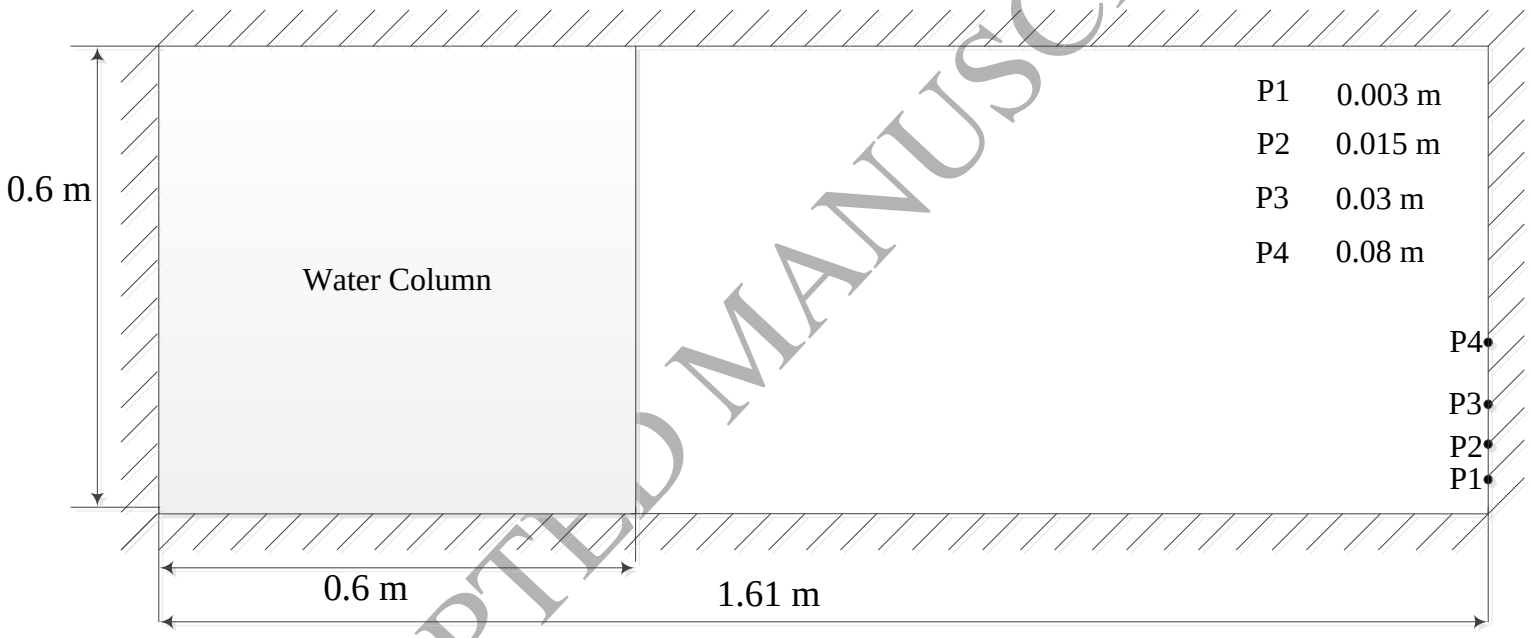

Fig. 6.3: Sketch of the Dam-break calculation model

g. 6.4 dam with obstacle sketch

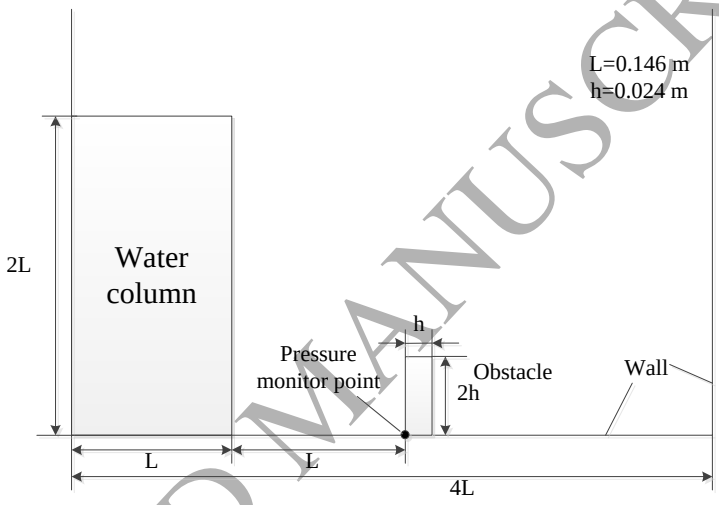

Fig.6.4: Sketch of the Dam break with obstacle model

g. 6.5 spring supported wall sketch

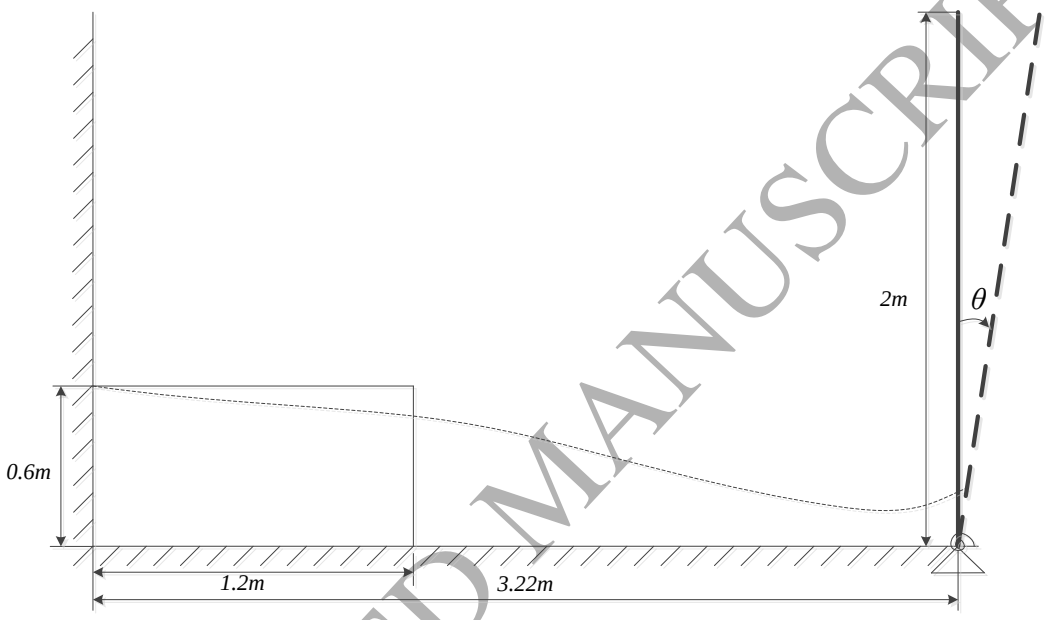

Fig.6.5: Sketch of the Dam-break calculation model

g. 6.6 Elastic beam sketch

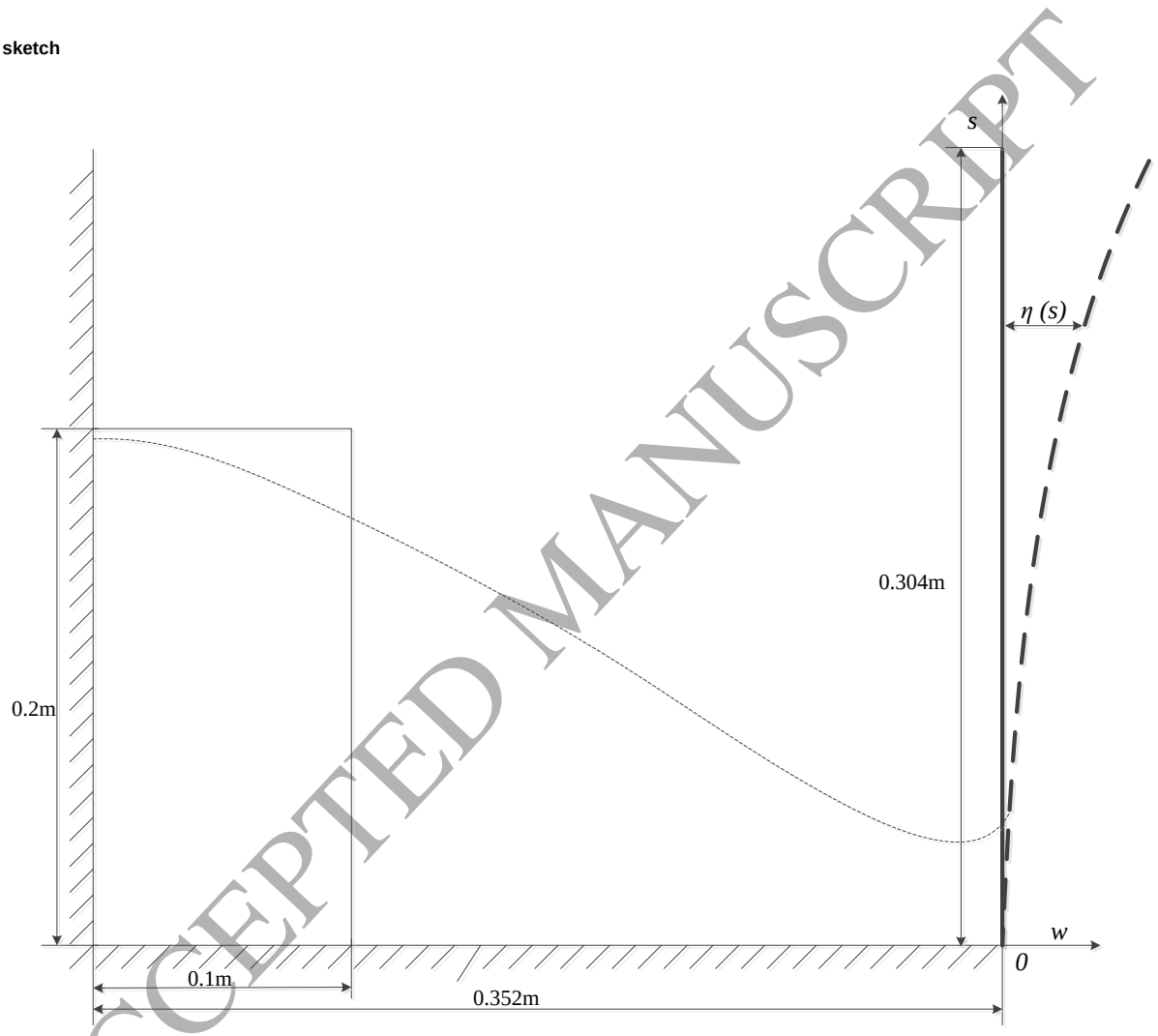

Fig. 6.6: Sketch of the Dam break with obstacle model

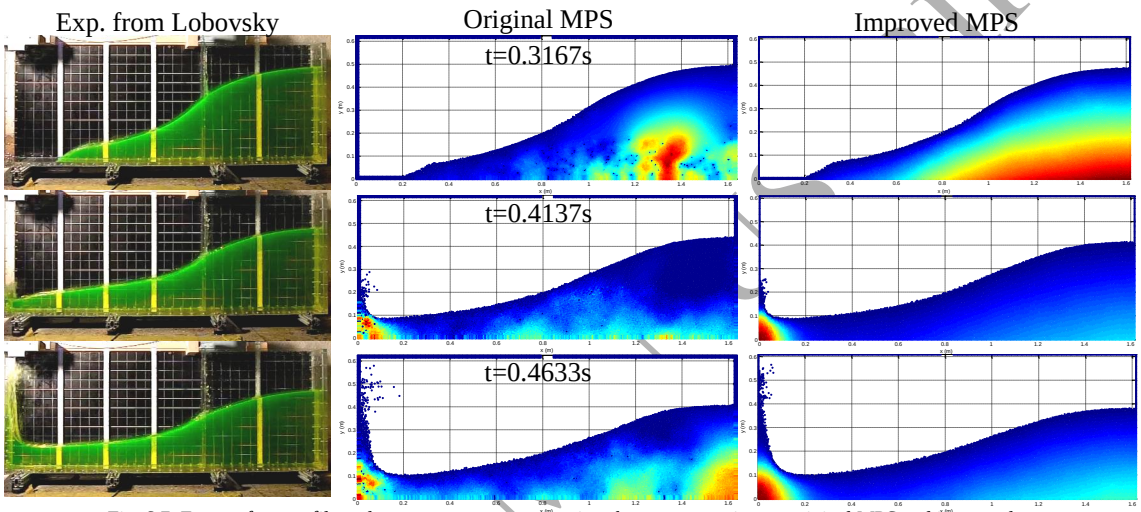

Fig. 6.7: Free surface profile and pressure contour comparison between experiment, original MPS and improved MPS

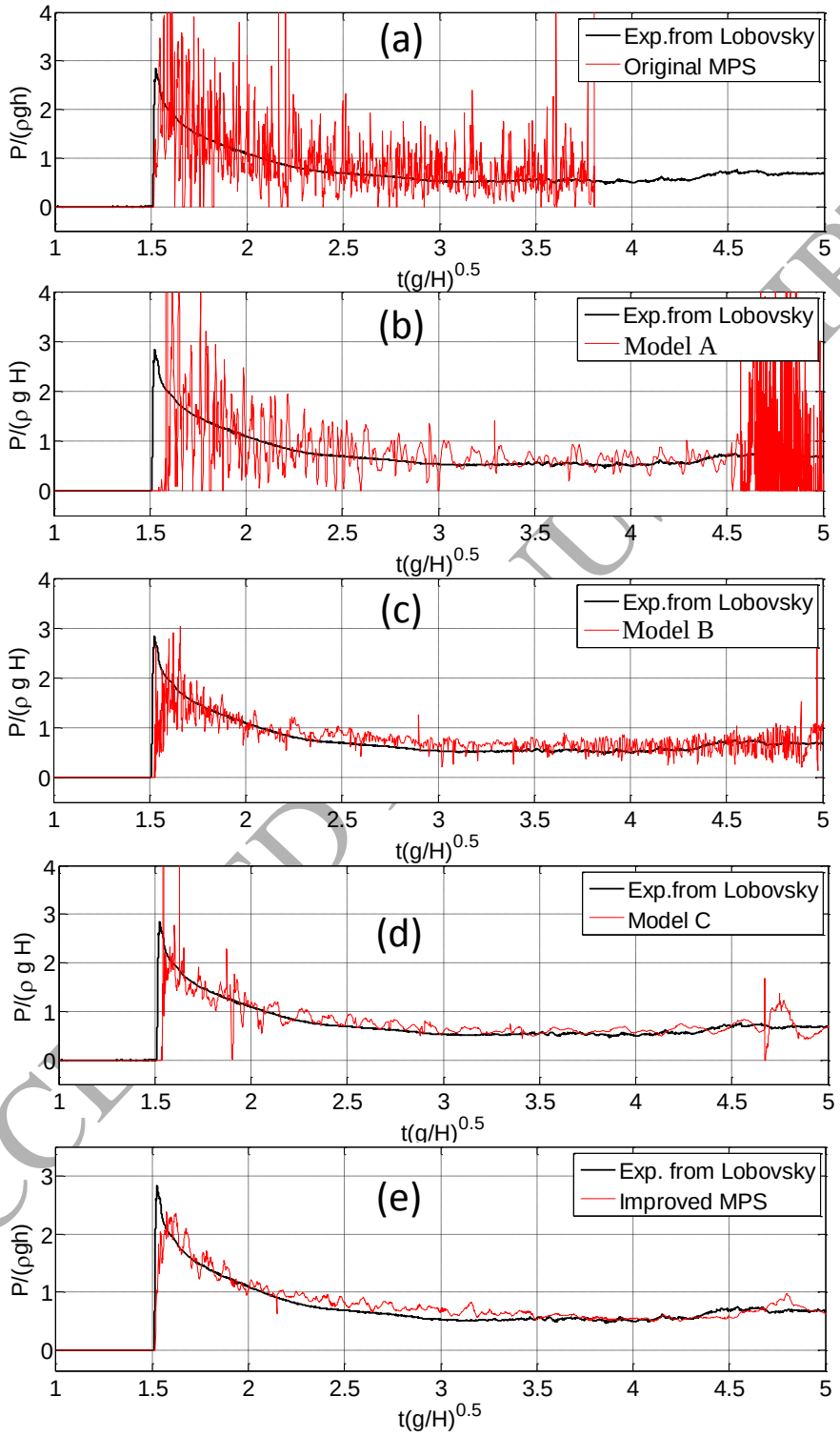

Fig. 6.8: Pressure history of different models monitored at P2 ((a) Original MPS; (b) DI source term, no particle shifting, proposed boundary condition; (c) DF source term, no particle shifting, proposed boundary condition; (d) Density error compensation source term, no particle shifting, proposed boundary condition; (e) MPS with all the proposed modifications

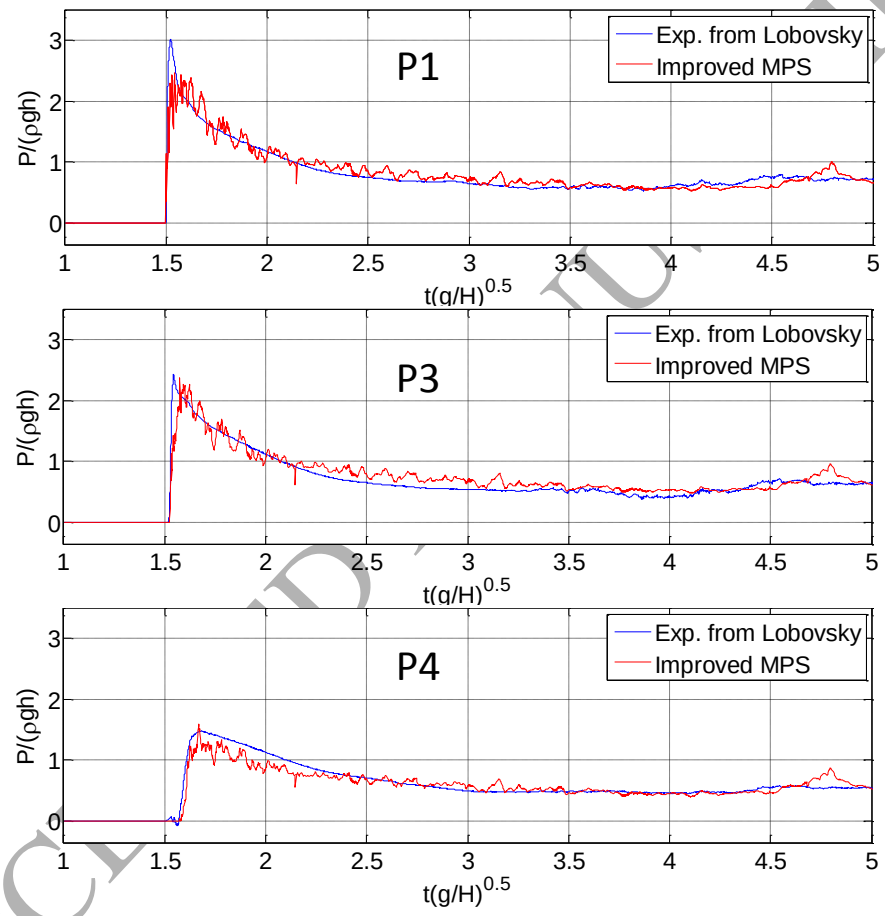

Fig. 6.9: Pressure history monitored at P1, P3 and P4

10 Comparison of p history for different particle distanc

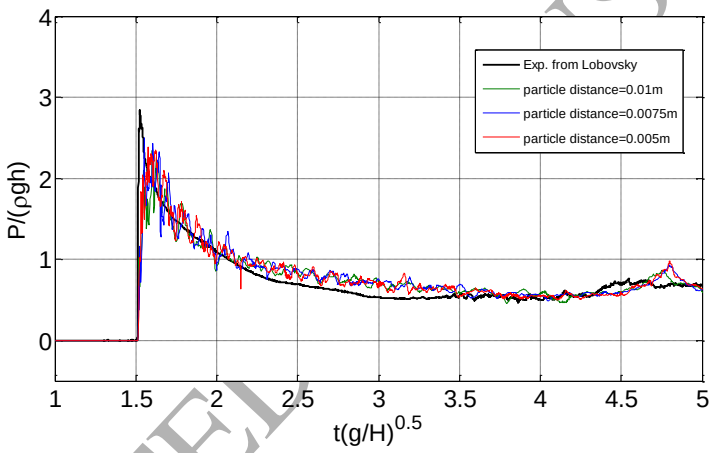

Fig. 6.10: Comparison of pressure time history for different particle distances

11 convergence test

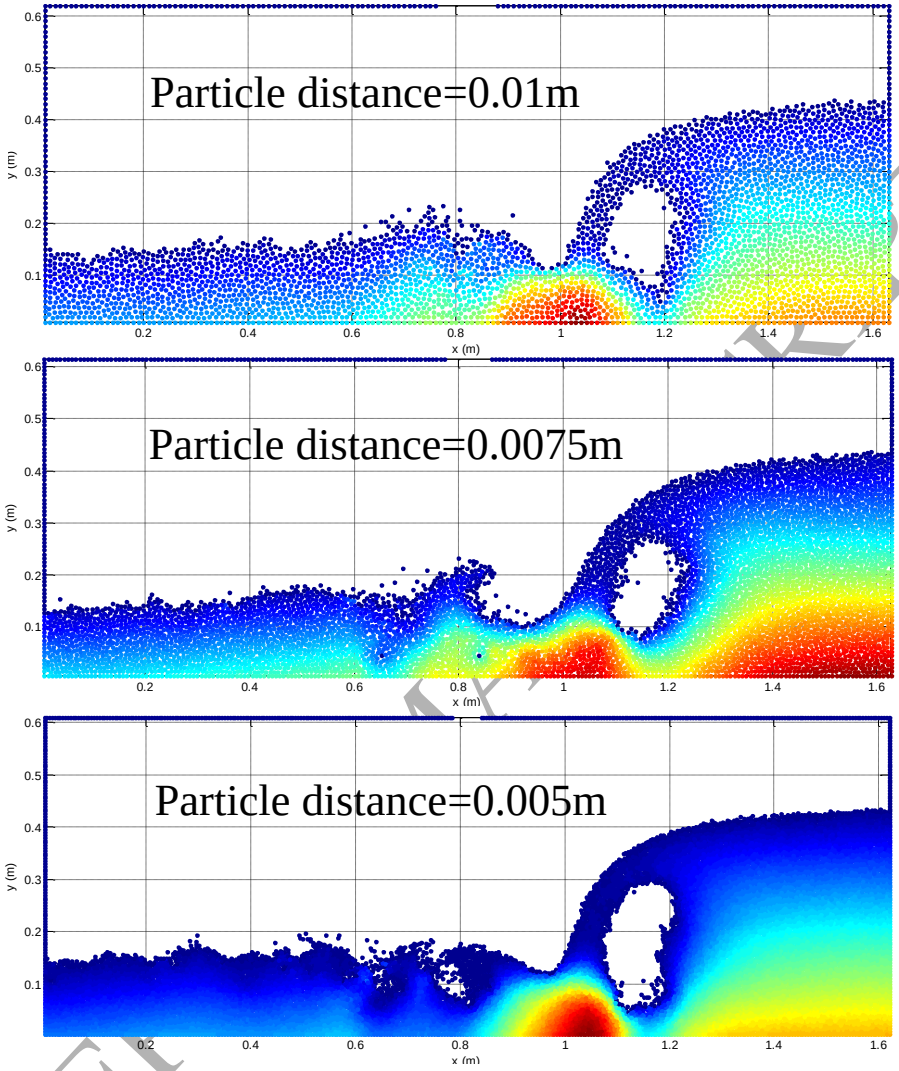

Fig. 6.11: Comparison of pressure contour at t=1.25s for different particle distances

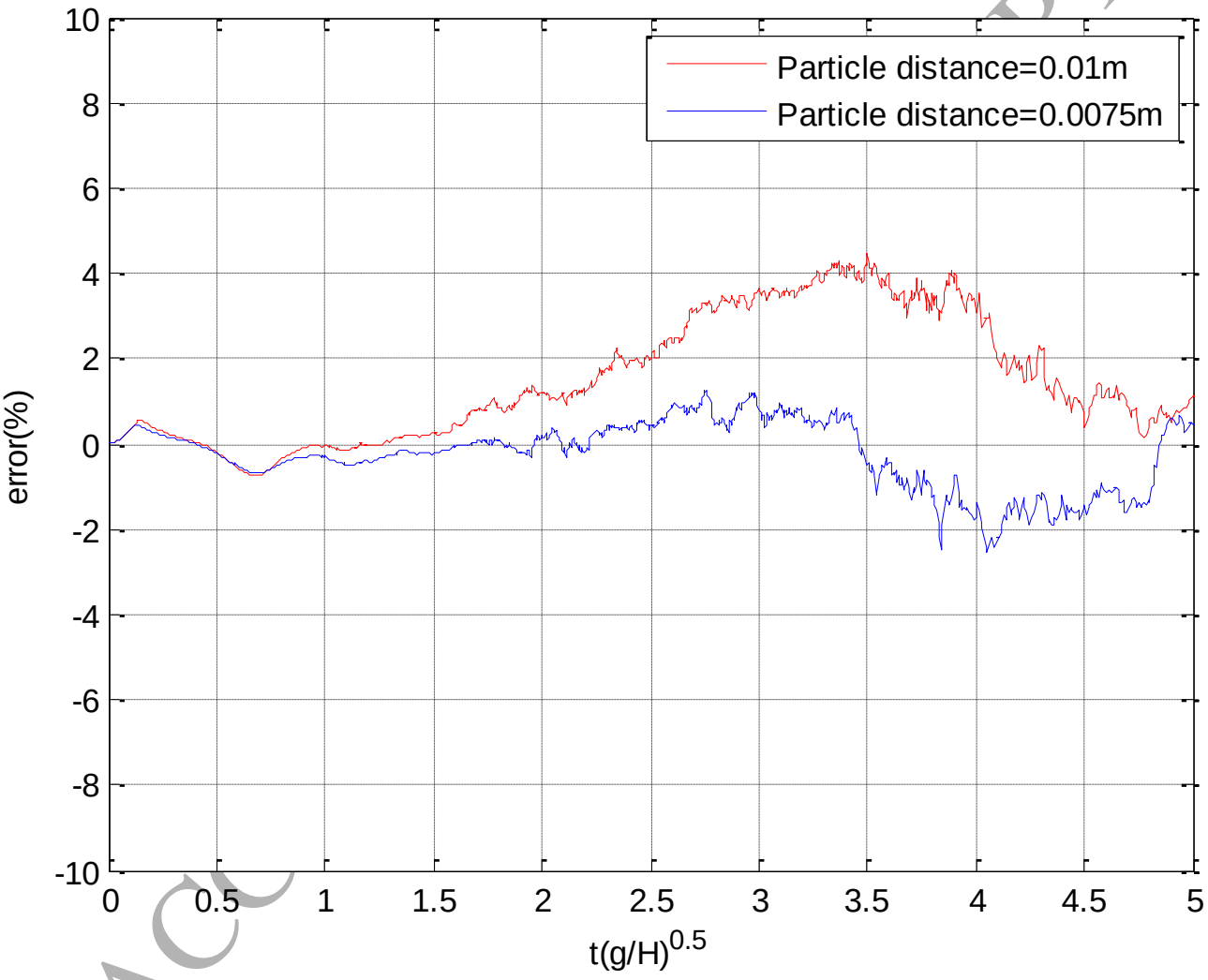

Fig. 6.12 Mass conservation comparison between different particle resolutions

13 Dam with obstacle-pressure contour

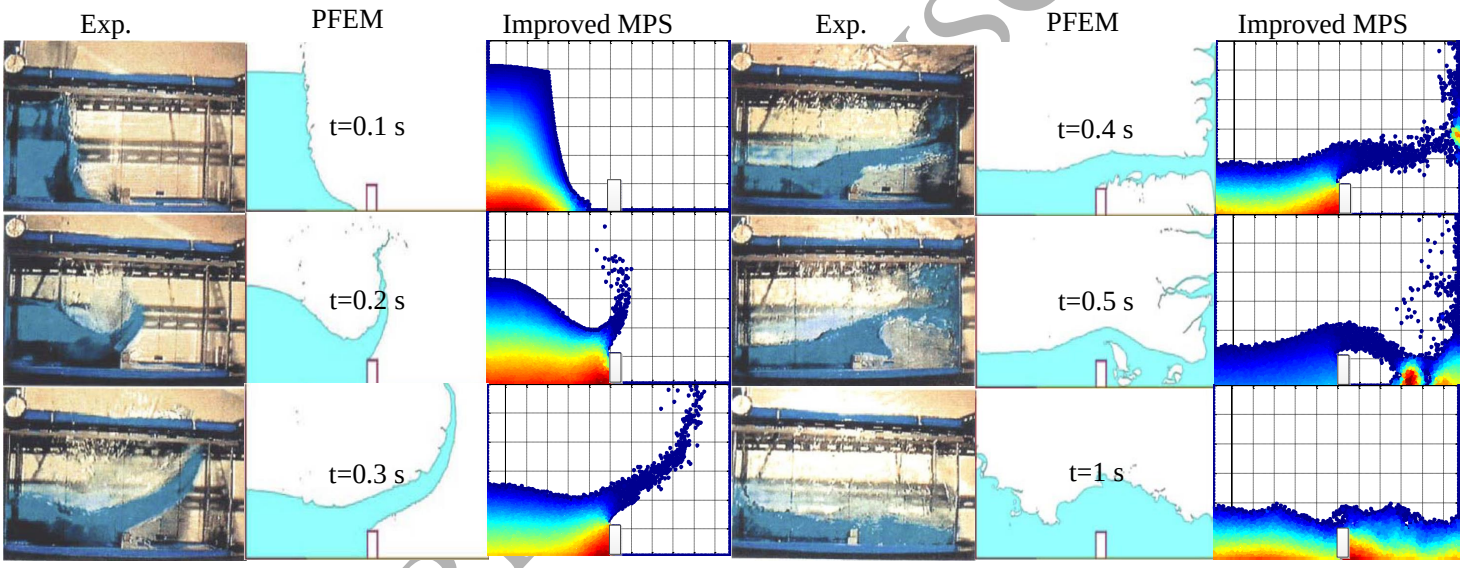

Fig. 6.13: Dam break with obstacle: free surface profiles (comparison between experiment[36],PFEM[10] and Improved MPS)

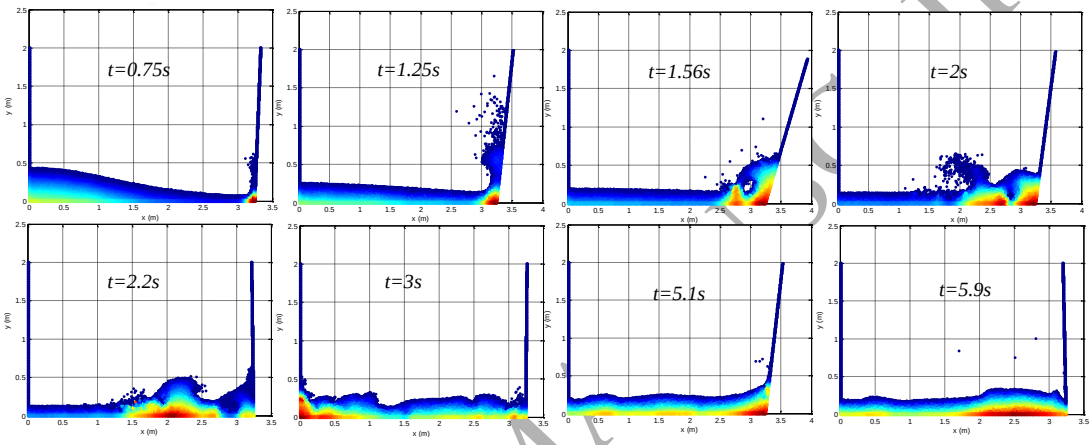

Fig. 6.14: Pressure contour and free surface profiles at several typical instants

15 Spring supported wall pressure and tip positon time hi

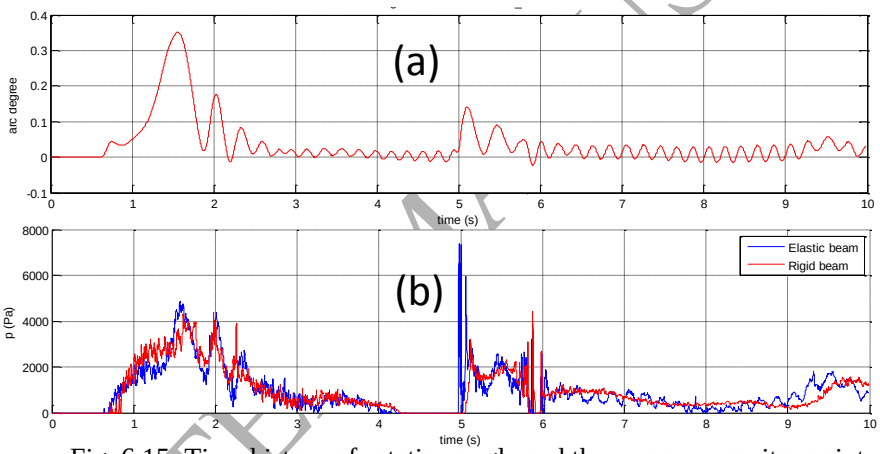

Fig. 6.15: Time history of rotation angle and the pressure monitor point

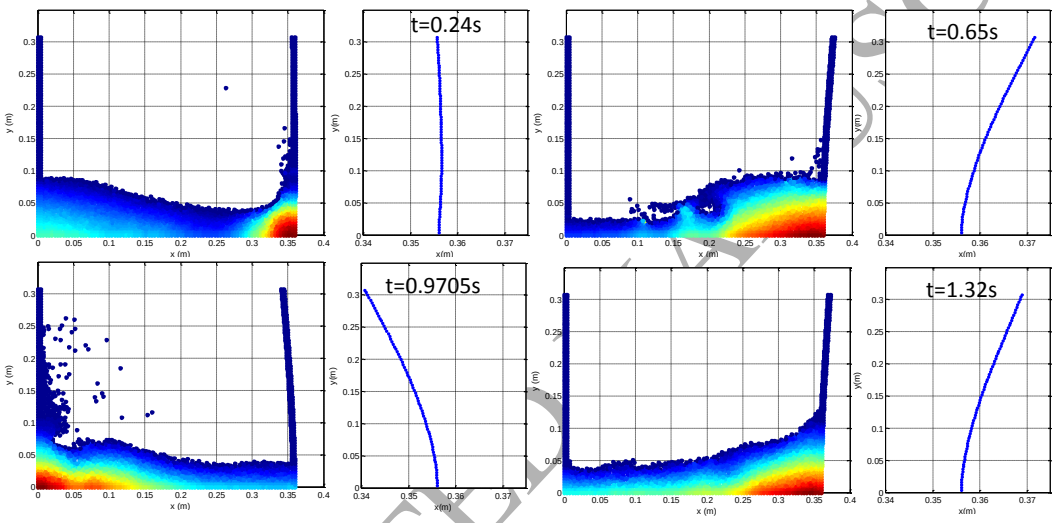

Fig. 6.16: Pressure contour, free surface profiles and beam deformations at several typical instants

17 flex_beam p monitor and top tip monitor

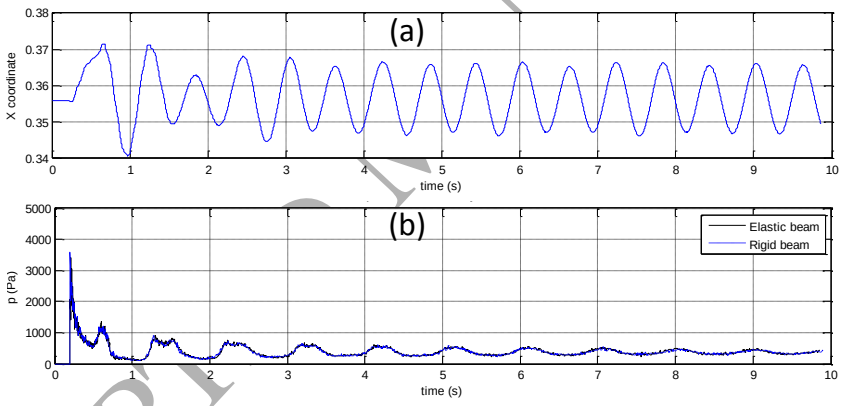

Fig. 6.17: Trajectory at the beam top and pressure time history at the impacting corner

g. 6.18 sloshing sketch

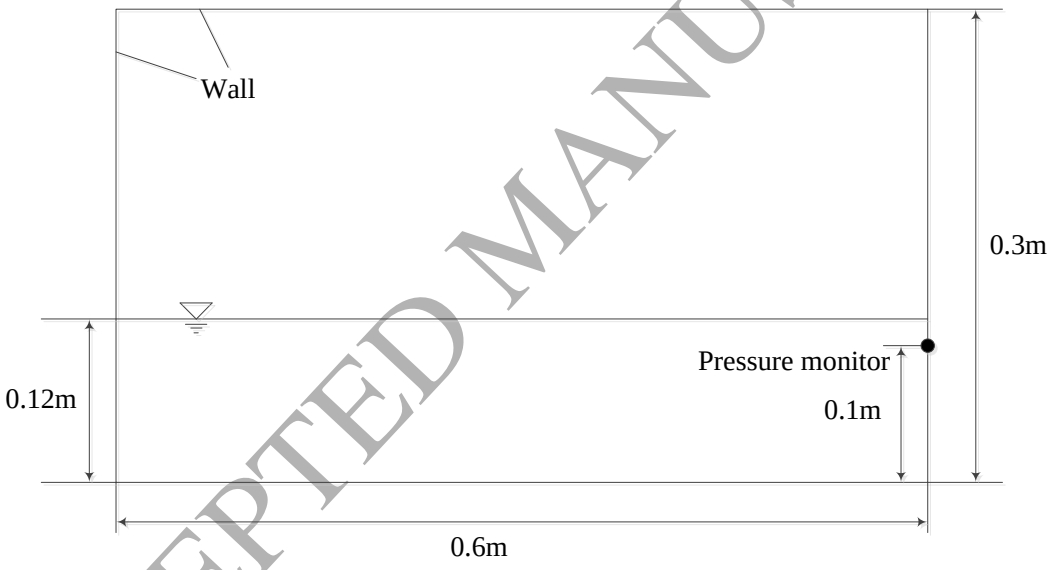

Fig. 6.18: Sketch of the sloshing model

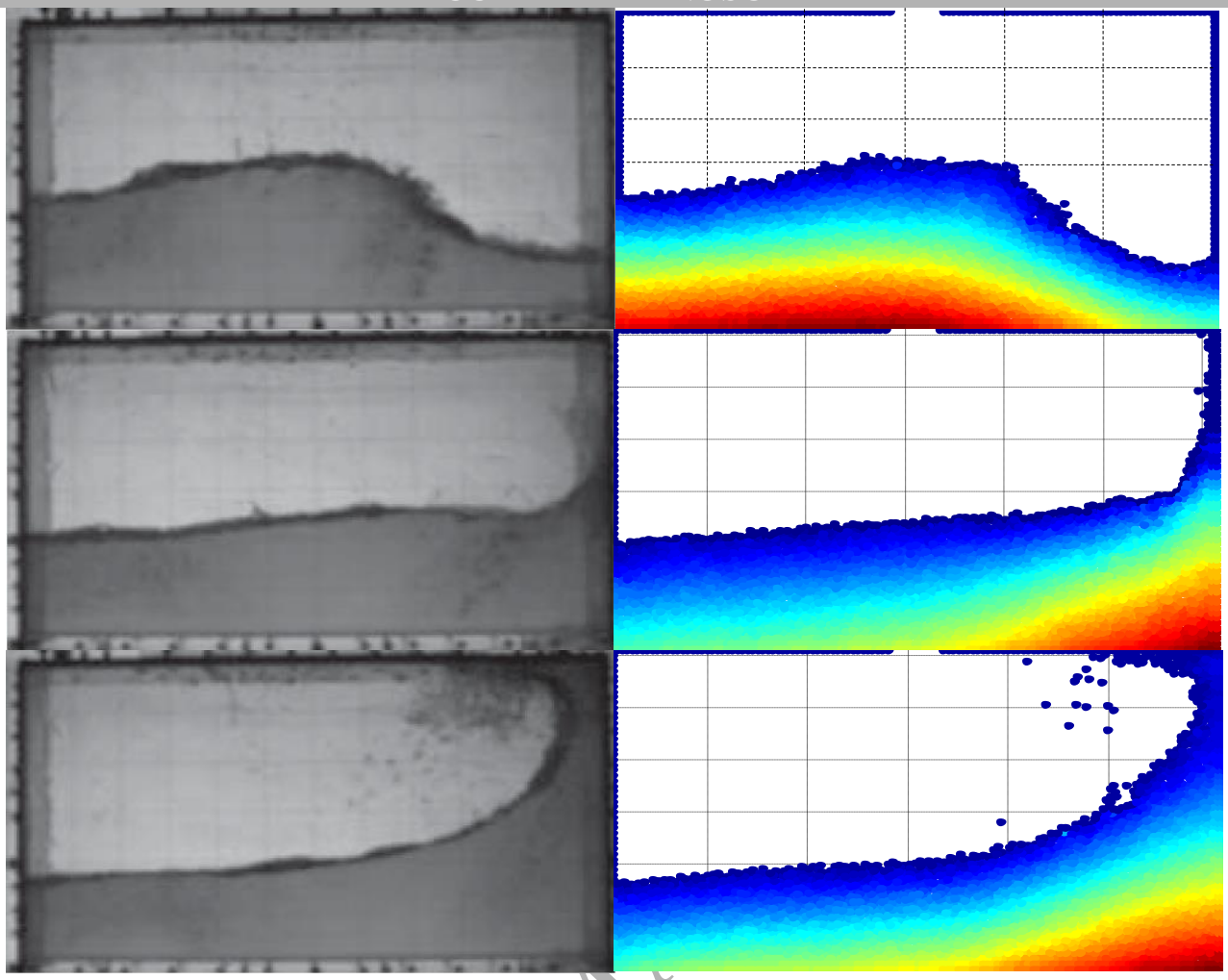

Fig. 6.19: Comparison of free surface profiles between experiment and Improved MPS

g. 6.20 sloshing pressure compare

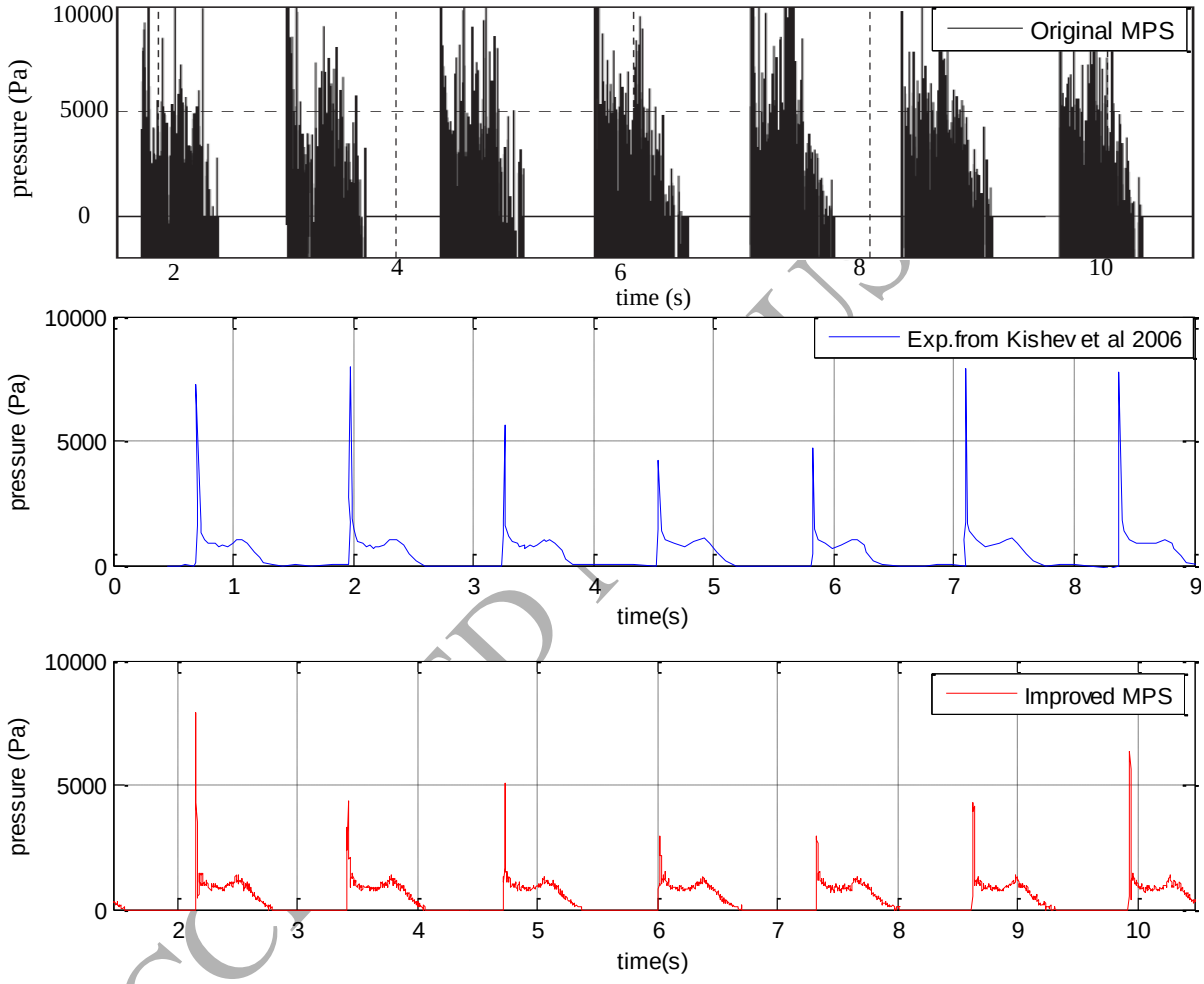

Fig. 6.20: Comparison of pressure with experiment of Kishev et al[38] and original MPS from B. H. Lee et al[14]

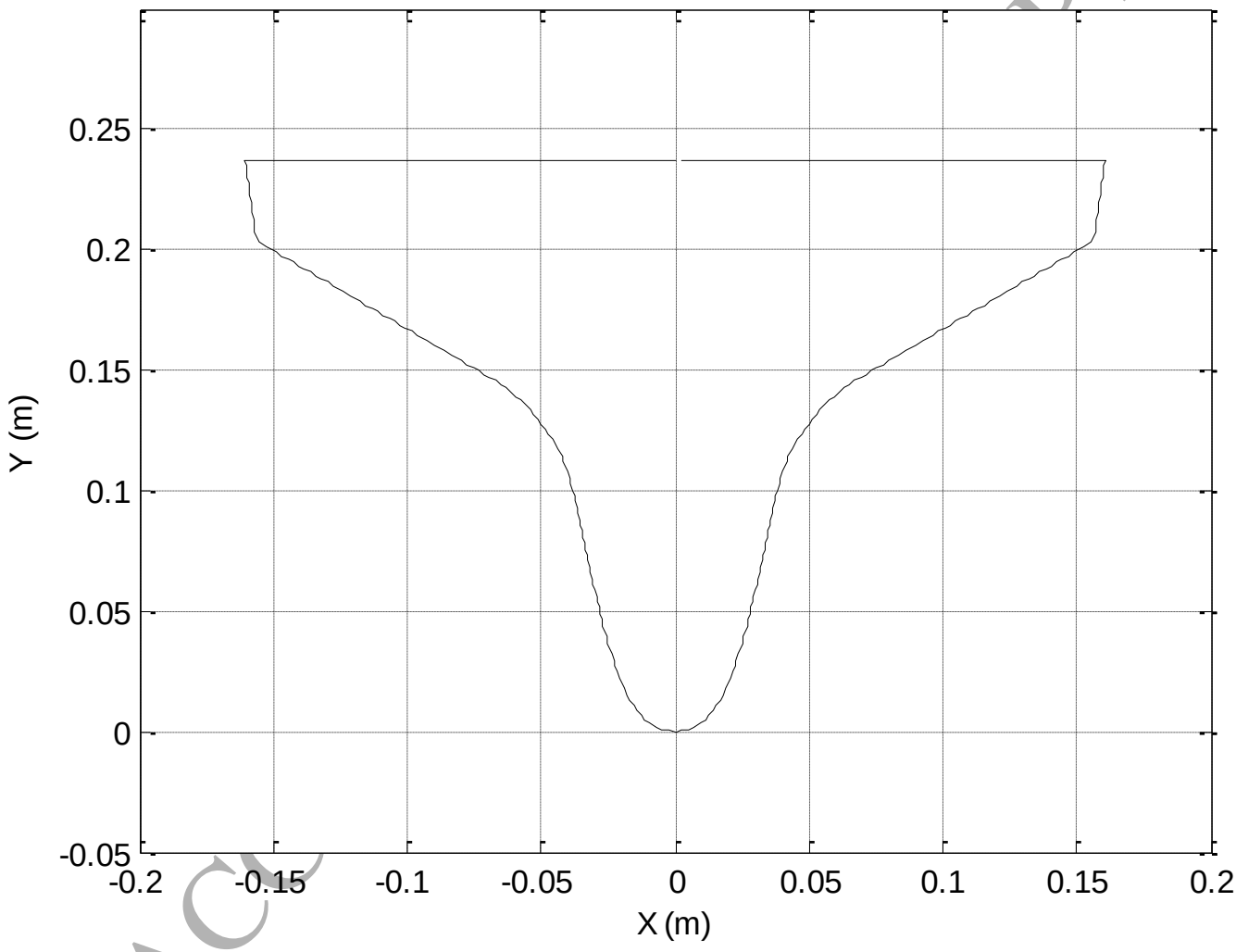

Fig. 6.21: Shape of the ship section

22 Ship section force and vel comparison(v0_61)

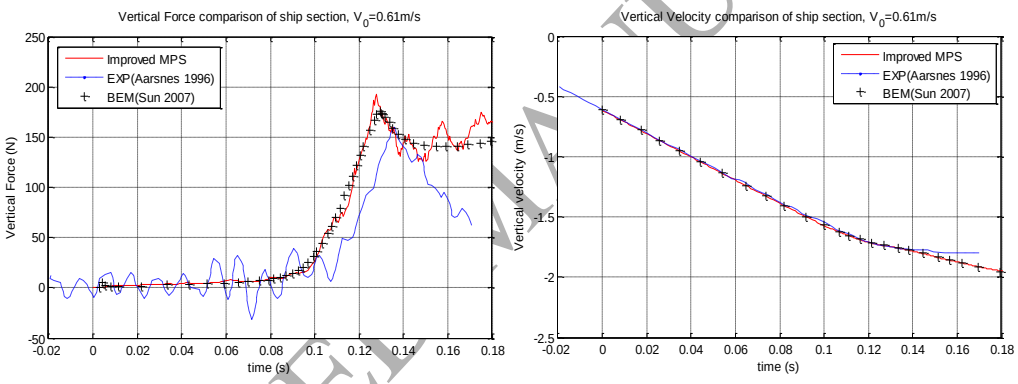

Fig. 6.22: Vertical force and velocity comparison with experiment and BEM (entry speed =0.61m/s)

23 Ship section force and vel comparison(v1_48)

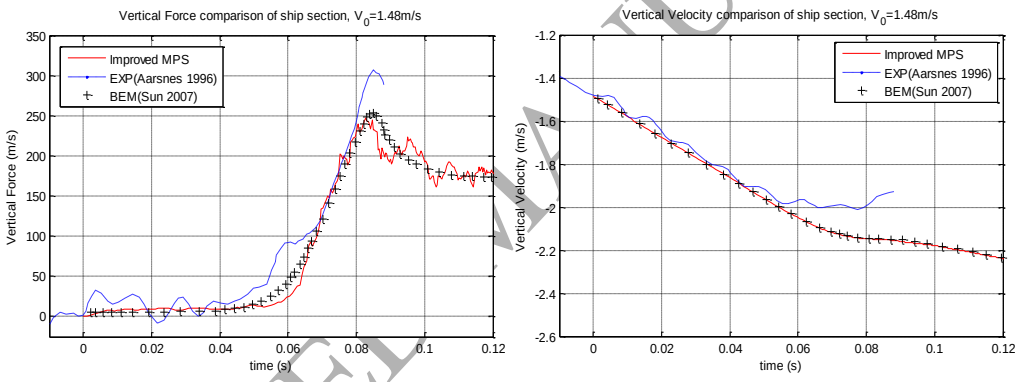

Fig. 6.23: Vertical force and velocity comparison with experiment and BEM (entry speed =1.48m/s)

24 Ship section dropping pressure contour

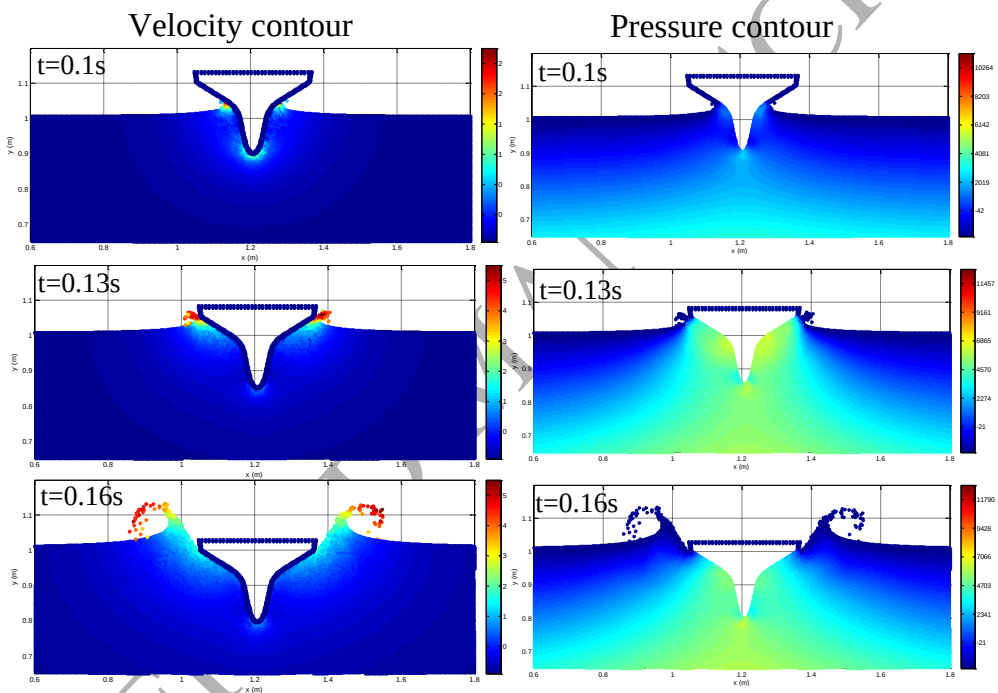

Fig. 6.24: Velocity and pressure contour of scenario 1(entry speed 0.61m/s)