

MLPG_R Method for Numerical Simulation of 2D Breaking Waves

Q.W. Ma^{1,2} and J.T. Zhou¹

Abstract: Following our previous work, the Meshless Local Petrov-Galerin method based on Rankine source solution (MLPG_R) will be extended in this paper to deal with breaking waves. For this purpose, the governing equation for pressure is improved and a new technique called Mixed Particle Number Density and Auxiliary Function Method (MPAM) is suggested for identifying the free surface particles. Due to complexity of the problem, two dimensional (2D) breaking waves are only concerned here. Various cases are investigated and some numerical results are compared with experimental data available in literature to show the newly developed method is robust.

Keywords: MLPG_R method; Mixed Particle Number Density and Auxiliary Function Method (MPAM); breaking waves; solitary wave.

1 Introduction

Wave breaking plays a significant role in air-sea interactions, such as energy and momentum transfer from wind to water and from waves to currents, and in the generation of turbulence in the upper ocean. It also plays an important role in bed-sea interactions, such as sediment transport or sand bar formation. In addition, wave breaking is the most concerned phenomenon associated with wave-structure interaction, for instance, interactions of breaking waves with breakwaters, offshore wind energy structures, offshore oil and gas exploitation platforms and transport vehicles. The force produced by these interactions is an important factor which must be considered in engineering design to secure the safety of the structures. As a result, wave breaking has been a topic of prime importance to the coastal/offshore engineering and environmental communities for many years. Nevertheless, it still remains a great challenge due to its high nonlinearity and complexities.

¹ School of Engineering and Mathematical Sciences, City University London, Northampton Square, London, UK, EC1V 0HB

² Communication: q.ma@city.ac.uk

Some investigations have been carried out by the laboratory experiments or field observations [Bonmarin, (1989); Rapp and Melville, (1990); Li and Raichlen, (2003)]. These investigations produced very useful results for some cases but are generally very expensive. In addition, many studies are based on numerical analysis, for which various numerical methods have been developed. The numerical methods can be grouped into mesh-based methods and meshless methods. The mesh-based methods for steep and/or breaking waves mainly include the finite difference method [e.g, Miyata, (1986); Lin and Liu, (1998)]; finite element method [e.g, Ma and Yan, (2006)], boundary element method [e.g, Grilli, Guyenne and Dias, (2001)] and finite volume method [e.g, Greaves (2009); Klessfsman, Fekken, Veldman, Iwanowski, Buchner (2005); Devrard, Marcer, Grilli, Fraunie and Rey, (2005)]. They all produced many impressive results. However, a limitation of mesh-based methods is that a computational mesh/grid is required and needs to be managed. Depending on whether using Lagrangian or Eulerian formulation, the mesh/grid may need to be updated repeatedly or to be refined to follow the motion of the free surface and need to be maintained to have good quality. This is often a difficult or time-consuming task, particularly in the cases with breaking waves.

Alternatives are meshless methods. In these methods, the fluid domain is discretised as nodes or particles, instead of a computational mesh/grid. Therefore, the limitation associated with mesh does not exist. So far, many meshless methods have been developed and reported in literatures, such as the Moving Particle Semi-implicit method (MPS) [e.g, Koshizuka and Oka, (1996); Gotoh and Sakai (2006); Khayyer and Gotoh (2008)], the Smooth Particle Hydrodynamic method (SPH) [e.g, Monaghan (1994); Khayyer, Gotoh and Shao (2008)], the finite point method [e.g, Onate, Idelsohn, Zienkiewicz, Taylor and Sacco (1996)], the element free Galerkin method [e.g, Belytschko, Lu and Gu (1994)], the diffusion element method [e.g, Nayroles, Touzot and Vilon (1992)], the method of fundamental solution [e.g, Wu, Tsay, Young (2006)] and so on. Among them, the MPS and SPH have been used to simulate breaking wave problems. For example, Monaghan (1994) modelled waves propagating over beaches; Lo and Shao (2002) and Koshizuka and Oka (1996) simulated the collapse of a water column; and Gotoh and Sakai (2006) investigated the breaking waves and the wave-body interaction problems. Khayyer, Gotoh and Shao (2008) simulated the breaking and post-breaking of solitary waves on a slope using the SPH method. Issa, Violeau, Lee and Flament (2009) made a review on how to deal with turbulent models in the SPH method.

More recently, another meshless method, called Meshless Local Petrove-Galerkin (MLPG) method, has been invented by Atluri and Zhu (1998) and Atluri and Shen (2002) and has been developed into many forms as summarized in Atluri, Liu and Han (2006) and reviewed by Ma (2009). This is a true meshless method without

need of any background mesh. It is based on a local weak form over simple sub-domains (circles for two dimensional problems and spheres for three dimensional ones). The success of the MLPG method has been reported in solving fracture mechanics problems [Batra and Ching (2002), Gao, Liu and Liu (2006)], beam and plate bending problems [Atluri and Zhu (2000)], shell and thermal elastic problems [Sladek, Sladek, Wen and Aliabadi (2006), Sladek, Sladek, Zhang and Tan (2006) and Jarak, Soric and Hoster (2007)], three dimensional elasto-static and -dynamic problems [Han and Atluri (2004a,b)], and some fluid dynamic problems, such as steady flow around a cylinder [Atluri and Zhu (1998)], steady convection and diffusion flow [Lin and Atluri (2000)] in one and two dimensions and lid-driven cavity flow in a two dimensional box [Lin and Atluri (2001), Mohammadi (2008)]. Over the past couple of years, the method has been extended to deal with more advanced problems. Han, Liu, Rajendran and Atluri (2006) made an advance in analyzing very complicated high-speed impact, penetration and perforation problems and indicated that the MLPG can be much faster than mesh-based methods for the problems of this kind. Li and Atluri (2008a,b) applied the MLPG method to analyze material orientation and to perform topology optimization of anisotropic solids and structures. Sladek, Sladek, Tan and Atluri (2008) studied transient heat conduction in 3D anisotropic solids. Ching and Chen (2006), Sladek, Sladek, Zhang and Solek (2007), Sladek, Sladek, Solek and Wen (2008), Sladek, Sladek, Solek, Wen and Atluri (2008) dealt with various thermo-structure dynamic problems. Vavourakis, Sellountos and Polyzos, (2006) and Sellountos, Sequeira and Polyzos (2009) performed elastic transient analysis using the MLPG(LBIE) method. Dang and Sankar (2008) applied the MLPG method to micromechanical analysis of periodic composites. Arefmanesh, Najafi and Abdi (2008) considered non-isothermal lid-driven cavity flow and non-isothermal flow over an obstacle. Developments are also made in refining the MLPG method, such as Yuan, Chen and Liu (2007), Pini, Mazzia and Sartoretto (2008), Cai and Zhu (2008), Chen, Liu and Cen (2008), Wang, Zhou, Nie and Kong (2008).

In addition to its various applications to solid and general fluid problems, Ma (2005a) extended the MLPG method to simulating nonlinear water waves and produced some encouraging results. In that paper, the simple Heaviside step function was adopted as the test function to formulate the weak form over local sub-domains, resulting in one in terms of pressure gradient.

In Ma (2005b), the MLPG method was further developed into a new form called the MLPG_R method, better suitable for modelling nonlinear water waves. In the MLPG_R method, the solution for Rankine sources rather than the Heaviside step function was taken as the test function. Based on this test function, a weak form of governing equations was derived, which did not contain any gradients of unknown

functions and therefore made numerical discretisation of the governing equation relatively easier and more efficient. A semi-analytical technique was also developed to evaluate the domain integral involved in this method, which dramatically reduce the CPU time spent on the numerical evaluation of the integral. Numerical tests showed that the MLPG_R method could be twice as fast as the MLPG method for modelling nonlinear water wave problems.

Ma (2008) made another step forward in the development of the MLPG_R method for water waves. In that paper, a new meshless interpolation was suggested, which is as accurate as the moving least square method but much more efficient, particularly for computation of gradient of unknown functions.

The MLPG_R method has been applied to model various nonlinear water waves, including the waves generated by a wavemaker, sloshing waves and freak waves [Ma, (2007)]. In this paper, it will be extended to deal with breaking waves. Because breaking waves are more complicated than non-breaking waves, new techniques will be developed to achieve the purpose. Further to them, different configurations under different wave conditions will be considered and some results will be compared with experimental data available in the public domain to show its accuracy.

2 Mathematical formulation

The flow of incompressible fluid is considered, which is confined in a two dimensional (2D) domain, as shown in Fig. 1. The figure also shows the coordinate system, in which the x -axis is on the mean free surface and $x=0$ corresponds to the left wall unless mentioned otherwise. In some cases, the sloped beach may be considered. For these cases, the left wall will move like a wavemaker to generate waves. d is the mean depth of fluid in the range from the wall to the sloped beach.

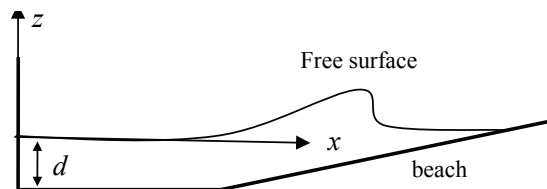


Figure 1: Sketch of the computational domain

The continuity equation and the Navier-Stokes equation (referred as NS equation hereafter) together with proper boundary conditions are given as follows:

$$\nabla \cdot \vec{u} = 0, \quad (1)$$

$$\frac{D\vec{u}}{Dt} = -\frac{1}{\rho}\nabla p + \vec{g} + \nu\nabla^2\vec{u}, \quad (2)$$

where \vec{g} is the gravitational acceleration; \vec{u} is the fluid velocity; ρ and ν are the density and the kinematical viscosity of fluid, respectively; and p is the pressure. On a rigid boundary, the velocity satisfies,

$$\vec{n} \cdot \vec{u} = \vec{n} \cdot \vec{U}, \quad (3)$$

where \vec{n} is the unit normal vector of the rigid boundary and \vec{U} is its velocity. On the free surface, the condition is specified by

$$p = 0. \quad (4)$$

The mathematical model given by Eqs. (1) to (4) is solved using a time marching procedure, which has been described by our previous papers [e.g, Ma, (2005b)] and will be summarised below.

Suppose the velocity, pressure and the location at n^{th} time step ($t=t_n$) are known, one can use the following procedure to find them at $(n+1)^{th}$ time step.

(1) Calculate the intermediate velocity (\vec{u}^*) and position (\vec{r}^*) of particles using

$$\vec{u}^* = \vec{u}^n + \vec{g}\Delta t + \nu\nabla^2\vec{u}^n\Delta t, \quad (5)$$

$$\vec{r}^* = \vec{r}^n + \vec{u}^*\Delta t, \quad (6)$$

where \vec{r} is the position vector of particles, the superscript n represents n^{th} time step; Δt is the increment of time step.

(2) Evaluate the pressure p^{n+1} using the following semi-implicit equation

$$\nabla^2 p^{n+1} = \alpha \frac{\rho^{n+1} - \rho^*}{\Delta t^2} + (1 - \alpha) \frac{\rho}{\Delta t} \nabla \cdot \vec{u}^*, \quad (7)$$

where α is an artificial coefficient between 0 and 1, and ρ^{n+1} and ρ^* are the fluid density at $(n+1)^{th}$ time step and at the intermediate time step, respectively. For the incompressible flow, the density should be a constant and so $\rho^{n+1} = \rho$, where ρ is the fluid density specified initially. The density ρ^* at the intermediate step may not be the same as ρ because the velocity and position calculated in Eqs. (5) and (6) do not necessarily satisfy the continuity equation given in Eq. (1).

(3) Calculate the fluid velocity and update the positions of the particles using

$$\vec{u}^{**} = -\frac{\Delta t}{\rho} \nabla p^{n+1} \quad (8)$$

$$\vec{u}^{n+1} = \vec{u}^* + \vec{u}^{**} = \vec{u}^* - \frac{\Delta t}{\rho} \nabla p^{n+1} \quad (9)$$

$$\vec{r}^{n+1} = \vec{r}^n + \vec{u}^{n+1} \Delta t \quad (10)$$

(4) Go to the next time step.

The above formulation is different from our previous work by Ma (2005a, b and 2008) in two aspects. (1) The viscous term is considered here but not considered there. (2) There are two terms on the right hand side of Eq. (7) rather than one term in the previous publications. Nevertheless, if $\alpha = 0$, the formulation here become the formulation in previous papers automatically. As pointed out in our cited papers, viscous effect is neglectable if waves are not breaking. That is why the associated term is ignored in those papers that considered only non-breaking steep waves. This paper aims to study breaking waves and thus viscosity is likely important. As a result, we must take it into account in Eq. (2). Regarding the change in Eq. (7), it can be shown that the governing equation for the pressure have two forms: one corresponding to $\alpha = 0$ and the other to $\alpha = 1$ [Koshizuka & Oka (1996); Idelsohn Storti and Onate (2001)], and both are derived by applying the continuity equation for incompressible fluid. Without wave breaking, the formulation with $\alpha = 0$ can yield good results as demonstrated in our previous papers. However, when wave breaking occurs, splash and re-entry take place frequently. In such cases, none of the two forms on their own works very well. Assigning a nonzero small value to α can improve the results dramatically, which will be demonstrated in the later section of the paper. Such an approach has also been adopted by Zhang, Morita, Kenji and Shirakawa (2006) for the MPS method.

3 MLPG_R formulation

Eq. (7) is solved by using the MLPG_R method. The details of the MLPG_R formulation and other techniques can be found in Ma (2005b). Here, we focus only on the aspects necessary to accommodate the change in Eq. (7). This method is based on a number of nodes or particles, which discretize the computational domain. Some nodes are located on the boundaries and others lie inside the fluid domain. At each of the inner nodes, a sub-domain is specified, which is a circle for 2D cases and Eq. (7), after multiplying by an arbitrary test function φ , is integrated

over the sub-domain, which becomes:

$$\int_{\Omega_I} \varphi \nabla^2 p^{n+1} d\Omega_I = \int_{\Omega_I} \left[\alpha \frac{\rho - \rho^*}{\Delta t^2} + (1 - \alpha) \frac{\rho}{\Delta t} \nabla \cdot \vec{u}^* \right] \varphi d\Omega_I \quad (11)$$

where Ω_I is the area of the sub-domain (also called integration domain) centered at Particle I and the density ρ^{n+1} has been replaced by ρ based on the discussion after Eq. (7). In the MLPG_R method, the test function is taken as the solution of Rankine source, i.e., the function φ satisfies $\nabla^2 \varphi = 0$, in Ω_I except for the center and $\varphi = 0$, on $\partial\Omega_I$ which is boundary of Ω_I . The expression of the solution for Rankine source is

$$\varphi = \frac{1}{2\pi} \ln(r/R_I) \text{ for 2D cases} \quad (12)$$

where r is the distance between a concerned point and the center of Ω_I and R_I is the radius of Ω_I .

In Eq. (11), the second derivative of unknown pressure and the gradient of the intermediate velocity are included. Numerical calculation of the derivative and gradient does not only requires much computational time but also degrades the accuracy. In order to obtain a better form, Eq. (11) is changed, by adding a zero term $p\nabla^2 \varphi$ and applying the Gauss's theorem, into

$$\int_{\partial\Omega_I + \partial\mathcal{E}} [\vec{n} \cdot (\varphi \nabla p) - \vec{n} \cdot (p \nabla \varphi)] dS = \alpha \int_{\Omega_I} \frac{\rho - \rho^*}{\Delta t^2} \varphi d\Omega + (1 - \alpha) \left[\int_{\partial\Omega_I + \partial\mathcal{E}} \frac{\rho}{\Delta t} \vec{n} \cdot (\varphi \vec{u}^*) dS - \int_{\Omega_I} \frac{\rho}{\Delta t} \vec{u}^* \cdot \nabla \varphi d\Omega \right] \quad (13)$$

where $\partial\mathcal{E}$ is a small surface surrounding the center of Ω_I , which is a circle in 2D cases. The reason for adding $\partial\mathcal{E}$ is that the test function φ in Eq. (11) becomes infinite at $r=0$ and so the Gauss's theorem can not be used otherwise. In the same way as in Ma (2005b), Eq. (13) can be reduced to:

$$\int_{\partial\Omega_I} \vec{n} \cdot (p \nabla \varphi) dS - p = \alpha \frac{\rho_I - \rho_I^*}{\Delta t^2} \frac{R_I^2}{4} + (1 - \alpha) \int_{\Omega_I} \frac{\rho}{\Delta t} \vec{u}^* \cdot \nabla \varphi d\Omega, \quad (14)$$

where it has been assumed that the increment of the density ($\rho - \rho^*$) is a constant within the sub-domain and so equal to its value at Particle I , which is acceptable not only because the density should not change much due to the change in the intermediate position of the particle but also because the small error caused due to

the assumption is further reduced by multiplying the coefficient α of a small value. The term may be evaluated in a more accurate way, for example by interpolation as done for the second term but such a way will not improve the accuracy significantly due to the reason.

It is obvious that Eq. (14) requires the intermediate density that is not computed in the MLPG_R method. Actually, the density can be replaced by a particle number density (PND) defined by Koshizuka and Oka (1996) in their MPS method as follows:

$$n_I = \sum_{j=1, j \neq I}^M W(|r_j - r_I|), \tag{15}$$

where W is a weight function in terms of the distance between Particle I and Particle j , which becomes zero when the distance is larger than a certain value. The domain, within which the weight function is not zero, is called support domain. In the above equation, M is the total number of particles in the support domain of Particle I . As indicated by Koshizuka and Oka (1996), the PND is related to the density by:

$$\rho_I = \frac{m_I n_I}{\int_{V_I} W(r) dV_I}, \tag{16}$$

where m_I is the mass of Particle I . After applying Eq. (16), Eq. (14) becomes

$$\int_{\partial\Omega_I} \vec{n} \cdot (p \nabla \phi) dS - p = \alpha \frac{\rho}{\Delta t^2} \frac{n^0 - n_I^*}{n^0} \frac{R_I^2}{4} + (1 - \alpha) \int_{\Omega_I} \frac{\rho}{\Delta t} \vec{u}^* \cdot \nabla \phi d\Omega \tag{17}$$

Ma (2005b) has detailed the method to discretise Eq. (17), in which the pressure on the left hand side is interpolated by a moving least square method (MLS) and the integration on the right hand side is evaluated by a semi-analytical technique. So the similar discussion will not be repeated here.

4 Boundary Conditions on a rigid boundary

Eq. (3) gives the condition on a rigid boundary in terms of velocity. To solve the problem about the pressure, the condition in terms of pressure is required on the boundary. For cases without considering the viscosity, the condition [Ma, (2005a, b)] is given as:

$$\vec{n} \cdot \nabla p = \rho \left(\vec{n} \cdot \vec{g} - \vec{n} \cdot \vec{U} \right) \tag{18a}$$

where \vec{U} is the acceleration of the boundary. This expression can be derived directly from Eqs. (2) and (3) by taking $v = 0$. When $v \neq 0$ as in the cases considered in

this paper, one may also derive the condition by using Eqs. (2) and (3) and obtain:

$$\vec{n} \cdot \nabla p = \rho \left(\vec{n} \cdot \vec{g} - \vec{n} \cdot \vec{U} + \nu \vec{n} \cdot \nabla^2 \vec{u} \right). \quad (18b)$$

It is obvious that one must compute the term $\nabla^2 \vec{u}$ when applying this condition, which needs to estimate the second order derivative at the rigid boundary. Although there is no much difficulty to do so theoretically, the error associated with it is not easy to be suppressed in computational practice as the fluid particles located only on one side of the boundary. Therefore, it is better to avoid the computation of the second order derivative when possible. For this reason, Eqs. (3) and (9) are combined to give an alternative equation for the boundary condition as follows:

$$\vec{n} \cdot \nabla p^{n+1} = \frac{\rho}{\Delta t} \vec{n} \cdot (\vec{u}^* - \vec{U}^{n+1}), \quad (19)$$

where \vec{u}^* is computed by Eq. (5) with \vec{u}^n taken as \vec{U}^n . If the solid boundary is fixed, $\vec{n} \cdot \vec{U}^{n+1} = 0$ and so Eq. (19) can be rewritten as

$$\vec{n} \cdot \nabla p^{n+1} = \frac{\rho}{\Delta t} \vec{n} \cdot \vec{u}^*. \quad (20)$$

Although one still needs to calculate $\nabla^2 \vec{u}$, which is inevitable, to estimate \vec{u}^* , the second order derivative term is not explicitly involved in Eqs. (19) and (20) and does not need to be calculated again after \vec{u}^* is available. This formulation will be used in this paper.

In numerical discretisation, Eq. (19) is directly applied to the nodes or particles on the rigid boundary, as what we did in Ma (2005b). However, in our previous papers, the gradient of the pressure is approximated by using the MLS. In this paper, it will be evaluated by the simplified finite difference interpolation (SFDI) scheme, recently proposed by Ma (2008). The SFDI scheme is as accurate as the MLS method but need less computational time. The reader is referred to Ma (2008) for more details about the scheme.

5 Technique for identifying free surface particles

In order to find solution for pressure by using the governing equation in previous section, all the particles need to be sorted into three groups: those on rigid boundaries (referred as wall particles), those on the free surface (referred as free surface particles) and others (referred as inner particles). The wall particles are always attached to the rigid boundary in our modeling. Eq. (19) or (20) is applied to wall particles. Eq. (17) is applied to the inner particles. The pressure at the free surface particles is specified by the condition in Eq. (4).

For non-breaking waves, one can assume that the particles initially on the free surface remain on the free surface. Therefore, the free surface particles need only to be specified at the first step and there is no need to identify them during the simulation.

However, for the cases with breaking waves as considered in this paper, the fluid particles initially on the free surface may not remain on the free surface during calculation. Actually, the particles initially within the fluid can emerge on the free surface and the particles initially on the free surface can immerge into the inner fluid domain due to wave breaking and splashing. Consequently, it is necessary to identify which particle is on the free surface when modeling breaking-wave cases.

There is a similar requirement on identifying the free surface in mesh-based numerical models for breaking waves. Several methods have been developed in the community using those models, such as level set method and volume-of-fluid technique. A brief review can be found in Greaves (2009).

Nevertheless, the identification of free surface particles remains to be a big challenge in meshless particle modeling, particularly for those of true meshless models without any background mesh, like the MLPG_R method. In the SPH method, the free surface particles are identified by the density; i.e., if the ratio of the density of a particle to the fluid density is less than a specified value, such as 1%, it is then identified as a free surface particle [e.g., Lo and Shao, (2002)]. The computation of the density in the SPH requires a background mesh, which is not available in the MLPG_R method. Another technique is based on the particle number density suggested for the MPS method by Koshizuka and Oka (1996) and also used by Gotoh and Sakai (2006). This technique uses the following parameter:

$$\beta_I = n_I^*/n^0, \quad (21)$$

where n_I^* is the particle number density at Particle I , as defined in Eq. (15), which is computed by using the particle configuration after the motion corresponding to the intermediate velocity. If $\beta_I < \beta$, Particle I is considered as a free surface particle. This approach is named as Particle Number Density Method, abbreviated to PNDM in this paper. Currently, there is no common agreement about how to specify the value of β . It is problem-dependent. Different researchers use different values. For example, it was 0.97 in Koshizuka and Oka (1996) and Gotoh and Sakai (2006) while it was 0.99 in Shao and Lo (2003). Results look to be promise in all the papers even with different values for β .

According to our numerical tests, the simple technique is not very robust. There are always many particles that are incorrectly identified (i.e., free surface particles are identified as inner particles or vice versa). The incorrect identification could

not be rectified by simply choosing a different value for β . Similar observation has also been described by Lee and Park (2007). To shed some light on the reason for this, let us look at Particle I and J in Fig. 2. Particle I is near the crest of a steep wave. In this area, the particles on the free surface are generally much closer to each other than in other areas and so the particle number density associated with them can be very higher. As a result, Particle I may be incorrectly identified as an inner particle rather than a free surface particle by the above criterion. On the other hand, the neighbor particle of Particle J is quite far from it. The value of the particle number density can be relatively smaller and so it may be incorrectly classified as a free surface particle even though it should be an inner particle. In addition to these situations, two or three splashed particles may be very close to each other and so may also be incorrectly identified as inner particles though they should be considered as free surface particles for the sake of solving pressure.

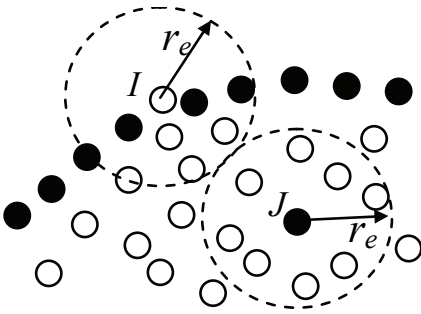


Figure 2: Two typical examples of incorrect identification of free surface particles. (Solid circle represents free surface particle identified by the PNDM; hollow ones represent inner particles identified by the PNDM).

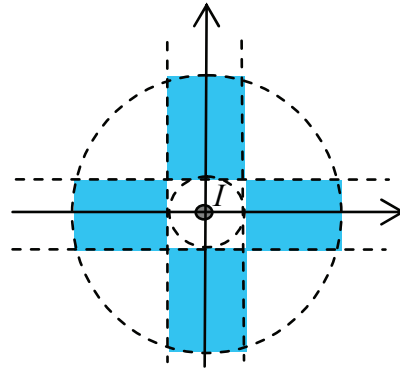


Figure 3: Local coordinate system at Particle I (inner circle denotes its integration domain; outer circle denotes its influence domain)

To improve the robustness of identification of free surface particles, a new approach will be suggested here. The approach is based on three auxiliary functions. The first auxiliary function is defined by

$$fsp_a(I) = \begin{cases} 1, & NumA \geq 1 \\ 0, & NumA = 0 \end{cases} \quad (22)$$

where $NumA$ represents the number of free surface particles existing in the influ-

ence domain (D_f) of Particle I in previous time step. The second auxiliary function is

$$fsp_b(I) = \begin{cases} 1, & NumB = 4 \\ 0, & NumB \leq 3 \end{cases}, \quad (23)$$

where $NumB$ represents the number of quadrants occupied by the neighbor particles of Particle I in a local coordinate system originating at Particle I , as shown in Fig. 3. The third auxiliary function is given by

$$fsp_c(I) = \begin{cases} 1, & NumC = 4 \\ 0, & NumC \leq 3 \end{cases}, \quad (24)$$

where $NumC$ represents the number of colored (or shaded) rectangles in Fig. 3 in which there is at least one fluid particle.

If one of following conditions is met when they are checked sequentially, Particle I is identified as a free surface particle.

- (a) no inner particle in D_f except for I ;
- (b) $\beta_I < 0.97$ and $fsp_a(I) = 1$;
- (c) $\beta_I \geq 0.97$, $fsp_a(I) = 1$ and $fsp_b(I) = 0$;
- (d) $\beta_I \geq 0.97$, $fsp_a(I) = 1$ and $fsp_c(I) = 0$.

Satisfying the condition (a) indicates that the particle concerned is in the group of particles which belong to the part of splashing fluid. The condition (b) identifies those free surface particles with at least one neighbour particle on the free surface. The condition (c) and (d) picks up those free surface particles with sufficient large number of neighbour particles but with no particle in a sufficient large part of its influence domain. This approach will be called as Mixed Particle Number Density and Auxiliary Function Method, abbreviated to MPAM.

Using this new approach, the incorrect identification of the free surface particles may be avoided. Let us look again at Particle I in Fig. 2. There are more than one free surface particle in its influence domain, and a large part of its influence domain is not occupied by any particle. As a result, $fsp_a(I) = 1$, $fsp_b(I) = 0$ and it will be identified, by applying Condition (b) or (c), as a free surface particle no matter what is the value for β_I . For Particle J shown in Fig. 2, there are no free surface particles existing in its influence domain and so $fsp_a(I) = 0$. Therefore, even though $\beta_I < 0.97$ for the particle, no one of four conditions will be met and thus Particle J is not identified as a free-surface particle.

6 Parameter study and validation

In the above formulation, we introduced the parameter α and the new technique to identify the free surface particles. In this section, some numerical results will be first discussed to show the effectiveness of the techniques. After that, validation against experimental data and the investigation on convergent properties will be presented.

In the following cases, water will be used as the fluid and the standard value of water density and viscosity are chosen. In addition, the parameters with a length scale are nondimensionalised by the water depth d and the time is nondimensionalised by $\sqrt{d/g}$, i.e. $t = \tau\sqrt{d/g}$. In all the cases, the particles are uniformly distributed at the beginning with the same distance between the particles in both directions. The number of particles along z-direction is represented by N_z .

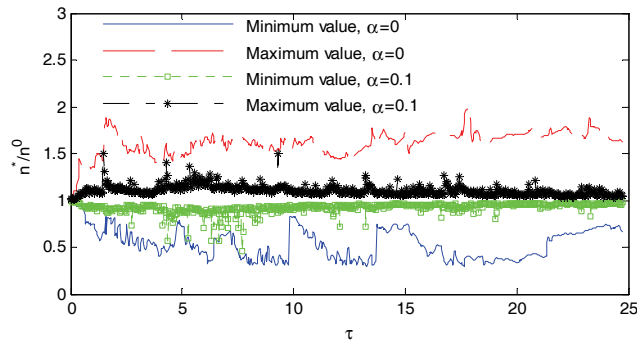
6.1 Effectiveness of Parameter α

As discussed in the paragraph just after Eq. (10), either of the form of Eq. (7) with $\alpha = 0$ or the one with $\alpha = 1$ does not work well in the cases for breaking waves. The root cause of the problem, based on the observation of numerical results, is that the distribution of the particles becomes over-distorted. This sub-section will present some results indicating that the problem may be overcome by choosing appropriate value for α .

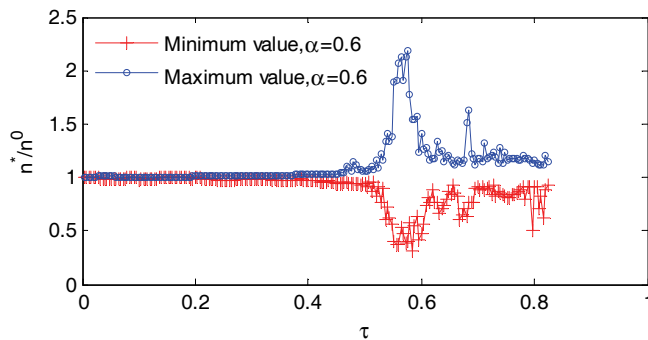
For this purpose, one needs to define an indicator that reflects the level of distortion of particle distribution. We will choose the PND as the indicator. That is because the change in the PND at inner particles defined in Eq. (15) can reflect the feature of particle distributions. At the beginning, the difference between the largest and the smallest PND of inner particles is a given value depending on the initial distribution of particles. It is very small if the initial distribution is almost uniform. With increase in the level of distortion of particle distribution, the difference between the largest and the smallest PND will grow. The larger the difference, the severer the distortion is.

To show how the PDN changes, a dam breaking problem is considered. This is a classical case for studying violent free surface flow, which has been numerically simulated by many researchers using various methods, such as Monaghan (1994) and Koshizuka and Oka, (1996). The geometry we use here is similar to that in Koshizuka and Oka (1996), i.e., the ratio of the length to the height of the water column confined by a vertical plate is 0.5 with the total length of the tank being 2. The plate is lifted off at $\tau=0$ instantaneously. To model this case, we take $N_z=50$ and the time step $\Delta\tau$ as 0.004. Because our aim here is to show the effectiveness of α , the change in particle number (N_z) and time step is not considered, though we

believe that the values chosen here are sufficient according to our numerical tests given below.



(a)



(b)

Figure 4: Effects of different α on the PND

The time histories of the ratios of the maximum and minimum values of the PND to the initial PND for inner particles excluding the free surface particle are recorded at each intermediate time step. They are plotted in Fig. 4(a) for the cases with $\alpha = 0$ and $\alpha=0.1$. One can see from this figure that the maximum ratio fluctuates at about 1.7 and the minimum ratio changes around 0.5 for the case with $\alpha = 0$. The difference between them is about 1.2. On the other hand, the difference between the maximum and minimum ratios for the case with $\alpha = 0.1$ is about 0.4 and tends to be consistently smaller with increase of time. The effect of α is further illustrated by using Fig. 5, where the configurations of particles at the same instant for $\alpha = 0$ and $\alpha = 0.1$ are plotted and two small areas are enlarged in the figure. The distribution

of the particles in the enlarged areas for $\alpha = 0.1$ is more uniform than that for $\alpha = 0$. These observations show that the particle distribution in the case with $\alpha = 0.1$ is better than in the case with $\alpha = 0$ and the former is likely to lead to better results.

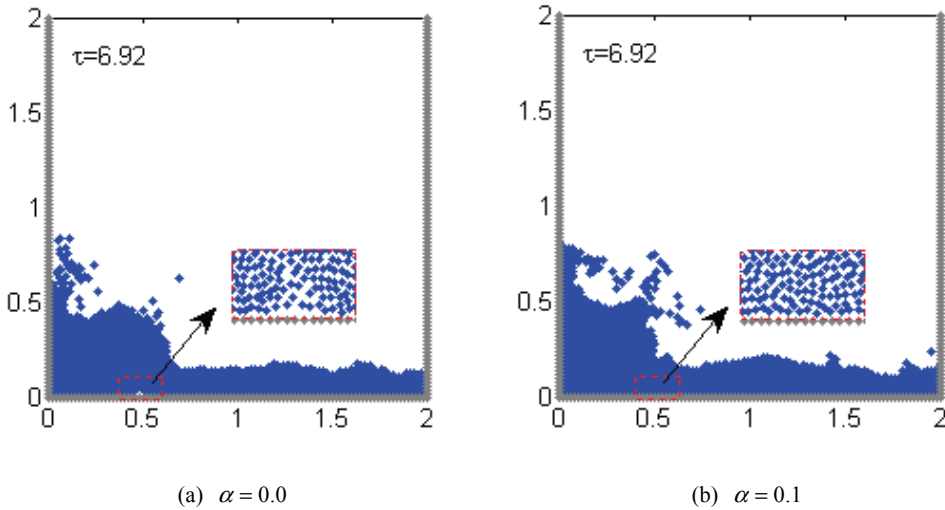


Figure 5: Configurations of particles for the cases with different value of α

The cases with other values of $\alpha > 0.1$ are also investigated. The similar results can be obtained with $\alpha=0.1$, $\alpha=0.2$ and $\alpha=0.3$. However, when $\alpha > 0.4$, the difference between the maximum and minimum PND is found to be large after a period of simulation even though it is small at the beginning. This feature is illustrated in Fig. (4b) for the case with $\alpha = 0.6$. It can be seen from this figure that the maximum and minimum ratios are almost the same from $\tau = 0$ to about $\tau = 0.5$ but in the area near $\tau = 0.57$ the difference reaches 1.5 with the maximum PND being 2.2 and the minimum PND being 0.7. Although the difference becomes smaller again after $\tau = 0.65$, the results for the flow is not right any more due to the error produced when it is large in the earlier steps. Based on these numerical investigations, the acceptable value of α is in the range from 0.1 and 0.3. In the rest part of this paper, α is chosen as 0.1. Nevertheless, the appropriate value may differ for other cases that are not considered in this paper.

6.2 Effectiveness of MPAM for identifying the free surface particles

To identify the free surface particles, a new approach is developed in the paper, called MPAM. In this sub-section, some numerical results will be presented to show

that it works better than the PNDM.

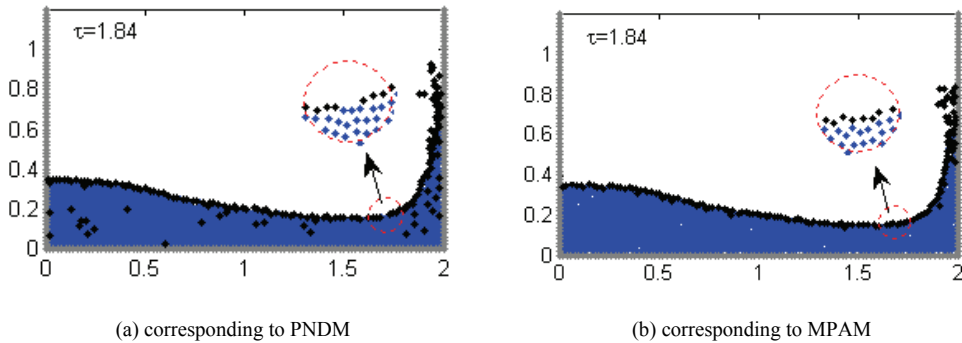


Figure 6: Comparisons of particle configurations obtained by using different free surface identification techniques (black color indicates free surface particles)

For this purpose, we will again consider the case in the above sub-section with $\alpha=0.1$. The case is simulated by using the MPAM and the PNDM, respectively. Particle configurations obtained by using the two different techniques at an instant are shown in Fig. 6. Fig. 6 (a) is the results corresponding to the PNDM while Fig. 6 (b) is these corresponding to the MPAM. This figure clearly demonstrates that many inner particles are wrongly identified as free surface particles by the PNDM but are correctly judged as inner particles by the MPAM. It also shows that the PNDM assigns some free surface particles to inner particles in the enlarged area but the MPAM does not make such a mistake.

6.3 Validation and convergence study

So far, we have not shown any comparison of our numerical results with experimental data nor given any discussion about the convergence property of the proposed method when it is used to simulate breaking waves. These will be addressed in this sub-section. In mesh-based methods, the convergence property is related to the element (or cell size) size in meshes or grids and time step ($\Delta\tau$). In meshless methods, like MLPG_R method, there is no mesh at all. Therefore, element (or cell) size is not relevant. A similar quantity is the distance between particles. Unfortunately, the distance continuously varies when modeling water waves. To bypass the problem, the initial distance of two adjoining particles is chosen as a representative distance. As the initial distribution of particles is uniform in the paper, the representative distance is equal to the difference in the x- or z- coordinates of two adjoining particles, which is denoted by Δx in the following discussion.

The two parameters (Δx and $\Delta \tau$) can be considered directly for studying the convergence property. They can also be replaced by other two parameters, such as $\Delta x/\Delta \tau$ and Δx . We prefer the second way because the later is related to the well-known Courant number that has a form of $\Delta \tau = c_t \Delta x$ ($\Delta t = c_t \Delta x/\sqrt{gd}$ in dimensional form) with c_t being a constant, as used by a number of researchers studying waves, for instance, Grilli, Guyenne and Dias (2001) and Yan and Ma (2009).

The case considered here is about the propagation of solitary waves over a beach with a slope of 1:15. The solitary wave is generated by a piston-type wavemaker according to the theory given by Goring (1978), in which the motion of the wave-maker is defined in dimensionless form by $x_p(\tau) = \frac{h}{k} [\tanh \chi(\tau) + \tanh k\lambda]$, where h is the wave height, $k = \sqrt{3h/4}$, $\chi(\tau) = k[c\tau - x_p(\tau) - \lambda]$ and the dimensionless celerity $c = \sqrt{(1+h)}$. The experimental data for this case are available in Li and Raichlen (1998). Its experimental set-up is similar to that illustrated in Fig. 7, in which the distance between the wavemaker and the toe of beach is 10 and the height of the solitary wave is $h=0.45$. In order to focus on the overturning and wave breaking process, which takes place over the beach, we assume the instant corresponding to the wave crest being at the toe of the beach to be $\tau = 0$ and the position of the toe to be $x=0$ for this case.

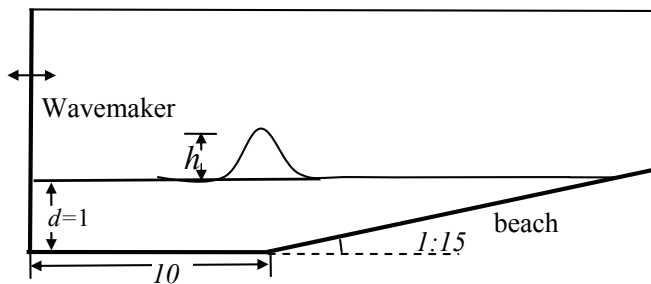


Figure 7: Illustration of model set-up for the solitary wave

Firstly, we look at how the ratio, $\Delta x/\Delta \tau$, affects numerical results. For this purpose, the representative distance, Δx , is taken as 0.05, corresponding to $Nz=20$, which yields the total particle number of 5893. The values of $\Delta x/\Delta \tau$ are chosen as 2.5, 5.0, 7.0 and 9.0, respectively. The solitary wave is generated by the wavemaker, which propagates to the beach and then evolves to overturning and breaking. The part of the process is illustrated in Fig. 8 by snapshots at several instants, which are obtained by using different values for $\Delta x/\Delta \tau$. For the purpose of comparison and validation, the experimental data from Li and Raichlen (1998) are also plotted in Fig. 8. It can be seen that at about $\tau = 9.29$, the wave crest becomes very steep;

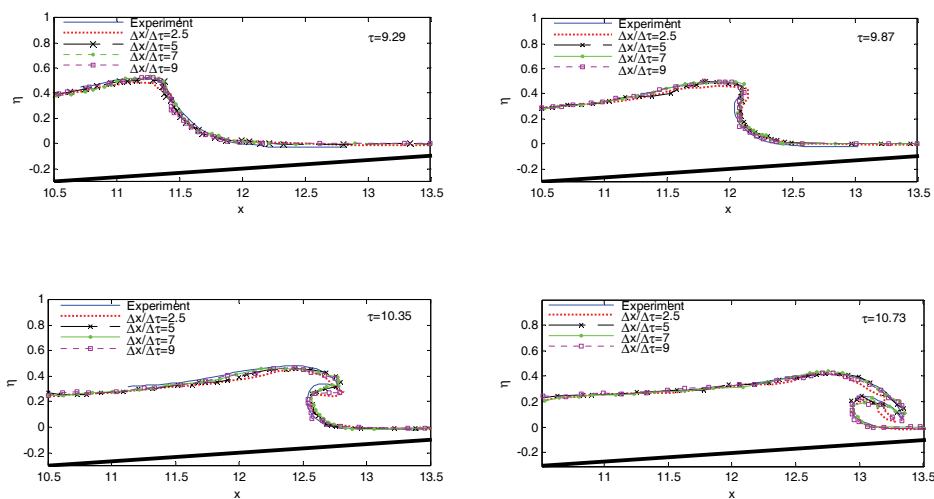


Figure 8: Comparison between experimental wave profiles (Li and Raichlen, 1998) and numerical results for different values of $\Delta x/\Delta \tau$

then at $\tau = 9.87$, a plunging jet starts to be formed; and after this, the jet is moving forward and nearly impacts on the other part of the water surface in its front at time $\tau = 10.73$. There is no experimental data after this instant. For validation purpose, we only present the results until this instant here.

It can also be seen that the numerical wave profiles generally agree well with the experimental ones except for those corresponding to $\Delta x/\Delta \tau = 2.5$. In addition, the numerical results for $\Delta x/\Delta \tau = 5$ and $\Delta x/\Delta \tau = 7$ are a bit closer to the experimental data than that for $\Delta x/\Delta \tau = 9$. This seems to imply that for a certain value of Δx , the time step should not be too large or too small. It is easy to understand why the time step should not go beyond the upper limit but the reason why it is also subjected to the lower limit from the point of view of accuracy needs to be further studied. Nevertheless, one should not choose too small time step in practice in order to save computational time if possible and so the lower limit may not be of a huge concern. It is worth noting that Yan and Ma (2009) and Grilli, Guyenne and Dias (2001) both have concluded that one may take $c_t = 0.4$ in $\Delta \tau = c_t \Delta x$ to simulate 2D overturning solitary waves using fully nonlinear potential theory for the finite element and boundary element simulations, respectively, which corresponds to $\Delta x/\Delta \tau = 2.5$. This investigation seems to suggest that the time step for the MLPG_R method based on the NS equation should be smaller (about half in this case) than that in the methods for potential theory to achieve similar accuracy. The information is useful because it helps researchers to guess the suitable time step a prior from experience

in using potential theory.

Secondly, we look at how the results vary by changing the value of Δx with $\Delta x/\Delta\tau = 5$. For this purpose, three different values for Δx are chosen. They are 0.05, 0.04 and 0.033, respectively. The corresponding numerical results are plotted in Fig. 9 together with the experimental data of Li and Raichlen (1998). This figure indicates that all these results up to $\tau = 10.73$ have an acceptable agreement with the experimental one. It also indicates that $\Delta x = 0.05$ is sufficient for the results in Fig. 8.

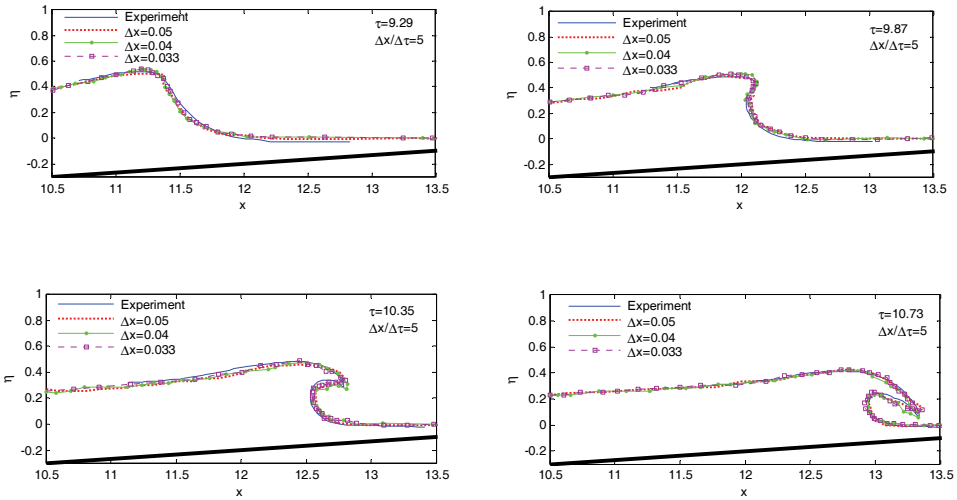


Figure 9: Comparison between experimental wave profiles [Li and Raichlen (1998)] and numerical results obtained by using different Δx values when $\Delta x/\Delta\tau = 5$

Nevertheless, it does not mean all the values are equally good beyond that instant. In order to show this, some snapshots for later instants are plotted in Fig 10, which illustrates the wave profiles after the jet impacts the front part of the water surface, called as post-breaking stage. As can be observed, another jet is formed and a cavity behind the jet appears after the impact. It is also observed that the profiles for $\Delta x = 0.04$ ($N_z=25$) and $\Delta x = 0.033$ ($N_z=30$) are quite similar but there is a significant discrepancy between the one for $\Delta x = 0.05$ and others. Although the experiential data is not available for the profiles in these instants, one may deduce that $\Delta x = 0.05$ may not be sufficiently small to model the post-breaking waves. In other words, more particles are required to model the waves in this stage.

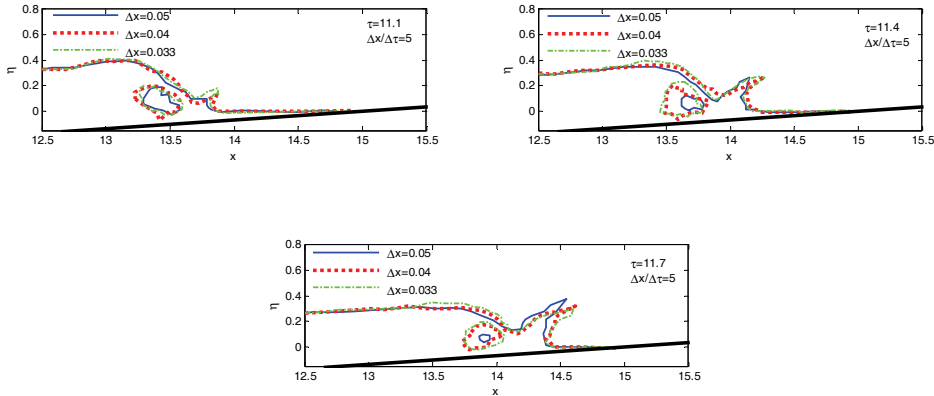


Figure 10: Wave profiles in the post-breaking stage obtained by using different values of Δx when $\Delta x/\Delta\tau = 5$

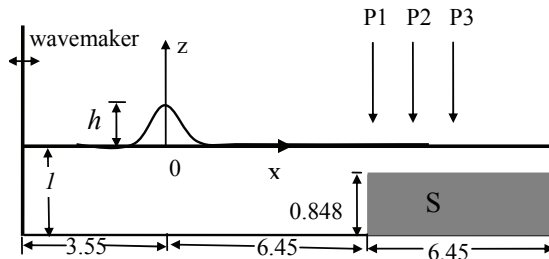


Figure 11: Sketch of the problem about a solitary wave propagating to and over the step

6.4 Further validation

To further validate the method, another case is considered, which has been experimentally studied by Yasuda, Mutsuda and Mizutani (1997). The sketch and the coordinate system are illustrated in Fig. 11, in which there are three wave gauges P1, P2 and P3 located at $x= 6.45, 8.11$ and 9.74 , respectively, to measure the wave elevations at the different positions. The solitary wave height is $h = 0.423$. The particle number (Nz) along the z -axis is 33 in the part before the step and is 5 over the step, respectively. The total particle number used in this case is 14806. The representative distance is chosen as $\Delta x = 0.03$. Taking $\Delta x/\Delta\tau = 5$, the time step is 0.006.

The solitary wave is generated by the piston-type wavemaker amounted at the left

side of tank in the same way above. At earlier stage, the solitary wave propagates towards the step without deformation. When it reaches the step, a part of the wave is reflected and other part of the wave is transmitted to the area over the step. The front of the transmitted wave becomes steep and a jet is formed near its crest. Then the wave is overturning and breaking occurs. To illustrate the process, four snapshots of solitary wave profiles over the step at different time steps are shown in Fig. 12. One can observe that a plunging jet is formed and almost hits the free surface at $\tau = 7.81$. After the jet impacts the free surface, a cavity appears and the second plunging jet is formed at $\tau = 8.43$. This process is largely similar to that for the solitary wave propagating over a sloped beach.

The time histories of the wave elevations recorded at the three wave gauges are shown in Fig. 13, together with the corresponding data from Yasuda, Mutsuda and Mizutani (1997). As can be seen from the figure, the wave form recorded at Gauge P1 is still similar to a full solitary wave form, but the maximum wave height is higher than the height at $x=0$. At Gauges P2 and P3, the wave elevations grows quickly but the wave height is relatively smaller compared with the height at $x=0$. This indicates that the wave has become very steep or broken before the points. As can also be seen, the numerical results are in very good agreement with the experimental data [Yasuda, Mutsuda and Mizutani 1997] at all the gauges. That again shows that the MLPG_R method works well in the cases with violent breaking waves.

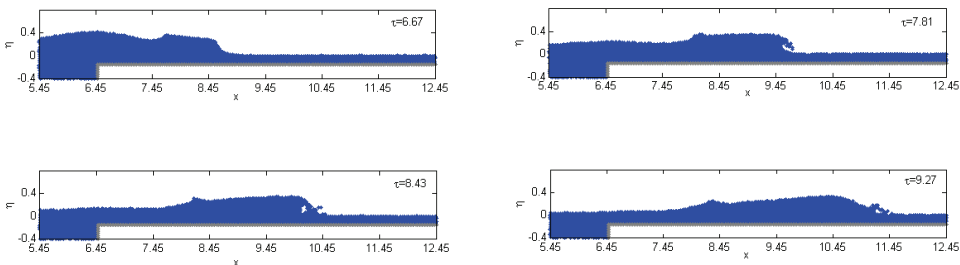


Figure 12: Snapshots of solitary wave evolution over the step at different time steps

7 Conclusions

In this paper, the MLPG_R method is extended to simulate 2D violent breaking waves. Compared with our previous work, the method is improved mainly in two aspects: (1) introducing an extra term in the governing equation for pressure and (2) suggesting a new approach (MPAM) for identifying the free surface particles.

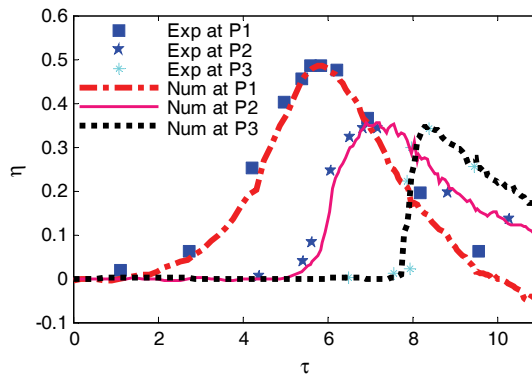


Figure 13: Comparisons of wave elevations between the numerical results (line) and experimental data (mark) (Yasuda et al, 1997) at three different gauges (P1, P2 and P3)

Numerical tests on different configurations under different wave conditions show that the improved method can yield satisfactory results for breaking waves, which agree reasonably well with experimental results.

Acknowledgement: This work is sponsored by The Leverhulme Trust, UK, to which the author is most grateful.

References

- Arefmanesh, A.; Najafi, M.; Abdi, H. (2008):** Meshless Local Petrov-Galerkin Method with Unity Test Function for Non-Isothermal Fluid Flow, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 25, No. 1, pp. 9-22.
- Atluri, S.N.; Shen, S. (2002):** The Meshless Local Petrove-Galerkin (MLPG) Method: A Simple & Less-costly Alternative to the Finite Element and Boundary Element Methods, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 3 (1), pp. 11-52.
- Atluri, S.N.; Zhu, T. (1998):** A New Meshless Local Petrov-Galerkin (MLPG) Approach in Computational Mechanics, *Computational Mechanics*, Vol. 22, pp. 117-127.
- Atluri, S.N.; Zhu, T. (2000):** New Concepts in Meshless Methods, *International J. Numerical Methods in Engineering*, Vol. 47 (1-3), pp. 537-556.
- Atluri, S.N. (2005):** Methods of Computer Modeling in Engineering and the Sciences. Vol. 1, Tech Science Press.

- Atluri, S.N.; Liu, H.T.; Han, Z.D.** (2006): Meshless Local Petrov-Galerkin (MLPG) Mixed Finite Difference Method for Solid Mechanics, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 15, No.1, pp. 1-16.
- Batra, R.C.; Ching, H.-K.** (2002): Analysis of Elastodynamic Deformations near a Crack/Notch Tip by the Meshless Local Petrov-Galerkin (MLPG) Method, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 3 (6), pp. 717-730.
- Belytschko, T.; Lu, Y.Y.; Gu, L.** (1994): Element-Free Galerkin methods, *Int. J. Numr. Meth. Eng.*, Vol. 37, pp. 229-256.
- Bonmarin, P.** (1989): Geometric Properties of Deep-Water Breaking Waves," *J. Fluid Mech.* 209, 405.
- Cai, Y.C.; Zhu, H.H.** (2008): A Local Meshless Shepard and Least Square Interpolation Method Based on Local Weak Form, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 34, No. 2, pp. 179-204.
- Chen, S.S.; Liu, Y.H.; Cen, Z.Z.** (2008): A Combined Approach of the MLPG Method and Nonlinear Programming for Lower-Bound Limit Analysis, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 28, No. 1, pp. 39-56.
- Ching, H.K.; Chen, J.K.** (2006): Thermomechanical Analysis of Functionally Graded Composites under Laser Heating by the MLPG Method, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 13, No. 3, pp. 199-218.
- Dang, T.D.; and Sankar, B.V.** (2008): Meshless Local Petrov-Galerkin Micromechanical Analysis of Periodic Composites Including Shear Loadings, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 26, No. 3, pp. 169-188
- Devrard, D.; Marcer, R.; Grilli, S.T., Fraunie, P.; Rey, V.** (2005): Experimental Validation of a Coupled BEM-Navier-Stokes Model for Solitary Wave Shoaling and Breaking. *Proc. 5th Intl. on Ocean Wave Measurement and Analysis*, pp 166-176.
- Gao, L.; Liu, K.; Liu, Y.** (2006): Applications of MLPG Method in Dynamic Fracture Problems, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 12, No. 3, pp. 181-196.
- Goring, D.G.** (1978): Tsunamis the Propagation of Long Waves onto A Shelf, *Report No. KH-R-38, W.M.Kech Laboratory of Hydraulics and Water Resources*, California Institute of Technology, Pasadena, CA. pp 337.
- Gotoh, H.; Sakai, T.** (2006): Key Issues in the Particle Method for Computation of Wave Breaking, *Coastal Engineering*, 53 (2-3), pp 171-179.
- Grilli, S.T.; Guyenne, P.; Dias, F.** (2001): A fully non-linear model for three-dimensional overturning waves over an arbitrary bottom, *Int. J. Numer. Meth. Fluid.*, Vol 35, pp 829-867.

Greaves, D. (2009): Application of the Finite Volume Method to the Simulation of Nonlinear Water Waves, Ch 10 in *Advances in Numerical Simulation of Nonlinear Water Waves* (ISBN: 978-981-283-649-6 or 978-981-283-649-7), edited by Q.W. Ma, scheduled to be published in Spring 2009 by The world Scientific Publishing Co.

Han, Z. D.; Atluri, S. N. (2004a): Meshless Local Petrov-Galerkin (MLPG) Approach for 3-Dimensional Elasto-dynamics, *CMC: Computers, Materials & Continua*, Vol. 1 (2), pp. 129-140.

Han, Z. D.; Atluri, S. N. (2004b): Meshless Local Petrov-Galerkin (MLPG) Approaches for Solving 3D Problems in Elasto-statics, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 6 (2), pp. 169-188.

Han, Z. D.; Liu, H. T; Rajendran, A. M.; Atluri, S. N. (2006): The Applications of Meshless Local Petrov-Galerkin (MLPG) Approaches in High-Speed Impact, Penetration and Perforation Problems, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 14, No. 2, pp. 119-128.

Idelsohn, S.R.; Storti, M.A.; Onate, E. (2001): Lagrangian Formulation to Solve Free Surface Incompressible Inviscid Fluid Flows, *Comput. Methods Appl. Mech. Engrg.* Vol 191, pp 583-593.

Issa, R.; Violeau, D.; Lee, E.S.; Flament, H. (2009): Modelling Nonlinear Water Waves with RANS and LES SPH Models, Ch 14 in *Advances in Numerical Simulation of Nonlinear Water Waves* (ISBN: 978-981-283-649-6 or 978-981-283-649-7), edited by QW Ma, scheduled to be published in 2009 by The world Scientific Publishing Co.

Jarak, T.; Soric, J.; Hoster, J. (2007): Analysis of Shell Deformation Responses by the Meshless Local Petrov-Galerkin (MLPG) Approach, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 18, No. 3, pp. 235-246.

Khayyer, A.; Gotoh, H. (2008): Modified Moving Particle Semi-implicit Methods for the Prediction of 2D Wave Impact Pressure, *Coastal Engineering*, doi: 10.1016/j.coastaleng. 2008.10.004.

Khayyer, A.; Gotoh, H.; Shao, S.D. (2008): Corrected Incompressible SPH Method for Accurate Water-Surface Tracking in Breaking Waves, *Coastal Engineering*, Vol 55, pp 236-250.

Klessfsman, K.M.T.; Fekken, G.; Veldman, A.E.P.; Iwanowski, B.; Buchner, B. (2005): A Volume-of-Fluid Based Simulation Method for Wave Impact Problems, *Journal of Computational Physics*, Vol 206, pp 363-393.

Koshizuka, S.; Oka, Y. (1996): Moving-Particle Semi-Implicit Method for Fragmentation of Incompressible Fluid, *Nuclear Science and engineering*, Vol 123, pp

421-434.

Lee, B.H.; Park, J.C. (2007): Numerical Simulation of Impact Loads Using a Particle Method, MPS, *Proceeding of the Seventeenth International Offshore and Polar Engineering Conference*, Lisbon, Portugal, July 1-6, pp 2029-2036.

Li, Y.; Raichlen, F. (1998): Discussion- Breaking Criterion and Characteristics for Solitary Waves on Slope, *J. Waterw., Port. Coastal, Ocean Eng.*, Vol 124, pp 329-335.

Li, Y.; Raichlen, F. (2003): Energy Balance Model for Breaking Solitary Wave Runup, *J. Waterw. Port. Coastal, Ocean Eng.*, Vol 129, pp 47-59.

Li, S.; Atluri, S. N. (2008a): Topology-optimization of Structures Based on the MLPG Mixed Collocation Method, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 26, No. 1, pp. 61-74.

Li, S.; Atluri, S.N. (2008b): The MLPG mixed collocation method for material orientation and topology optimization of anisotropic solids and structures, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 30 (1): 37-56.

Lin, P.Z; Liu, Philip L.-F (1998): A numerical study of breaking waves in the surf zone, *J. Fluid Mech*, vol. 359, pp. 239-264.

Lin, H.; Atluri, S.N. (2000): Meshless Local Petrov-Galerkin (MLPG) method for convection-diffusion problems, CMES: Computer Modeling in Engineering & Sciences, Vol. 1 (2), pp. 45–60.

Lin, H.; Atluri, S.N. (2001): The Meshless Local Petrov-Galerkin (MLPG) method for solving incompressible Navier-Stokes equations, CMES: Computer Modeling in Engineering & Sciences, Vol. 2 (2), pp. 117-142.

Lo, E.Y.M.; Shao, S.D. (2002): Simulation of near-shore solitary wave mechanics by an incompressible SPH method”, *Applied Ocean Research*, Vol 24, pp275-286.

Ma, Q.W. (2005a): Meshless Local Petrov-Galerin Method for Two-dimensional Nonlinear Water Wave Problems, *Journal of Computational Physics*, Vol 205, Issue 2, pp 611-625.

Ma, Q.W. (2005b): MLPG Method Based on Rankine Source Solution for Simulating Nonlinear Water Waves, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 9, No 2, pp 193-209.

Ma, Q.W.; Yan, S (2006): Quasi ALE Finite Element Method for Nonlinear Water Waves, *Journal of Computational Physics*, 212, pp 52-72.

Ma, Q.W. (2007): Numerical Generation of Freak Waves Using MLPG_R and QALE-FEM Methods, *CMES: Computer Modeling in Engineering & Sciences*, Vol.18, No.3, pp.223-234.

- Ma, Q.W.** (2008): A New Meshless Interpolation Scheme for MLPG_R Method, *CMES: Computer Modeling in Engineering & Sciences*, Vol 23, No 2, pp 75-89.
- Ma, Q.W.** (2009): MLPG_R Method and Its Applications to Various Nonlinear Water Waves, Ch 15 in *Advances in Numerical Simulation of Nonlinear Water Waves* (ISBN: 978-981-283-649-6 or 978-981-283-649-7), edited by QW Ma, scheduled to be published in 2009 by The world Scientific Publishing Co.
- Mohammadi, M.H.** (2008): **Stabilized Meshless Local Petrov-Galerkin** (MLPG) Method for Incompressible Viscous Fluid Flows, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 29, No. 2, pp. 75-94.
- Miyata, H.** (1986): Finite-difference simulation of breaking waves, *Journal of Computational Physics*, Vol. 65, issue 1, pp. 179-214.
- Monaghan, J.J.** (1994): Simulation Free Surface Flows with SPH, *Journal of Computational Physics*, Vol. 110, pp 399-406.
- Nayroles, B.; Touzot, G.; Villon P.** (1992): Generalizing the Finite Element Method, Diffuse Approximation and Diffuse Elements, *Computational Mechanics*, Vol. 10, pp. 307-318.
- Onate, E.; Idelsohn, S.; Zienkiewicz, O.C.; Taylor, R.L.; Sacco, C.** (1996): A stabilized Finite Point Method for Analysis of Fluid Mechanics Problems, *Comput. Methods Appl. Mech. Engrg.* 139, pp 315-346.
- Pini, G.; Mazzia, A.; Sartoretto, F.** (2008): Accurate MLPG Solution of 3D Potential Problems, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 36, No. 1, pp. 43-64.
- Rapp, R.J; Melville, WK** (1990): Laboratory Measurements of Deep-Water Breaking Waves, *Philos. Trans. R. Soc. London*, Ser. A 3777, 311.
- Sellountos, E.J.; Sequeira, A.; Polyzos, D.** (2009): Elastic transient analysis with MLPG(LBIE) method and local RBFs, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 41, No. 3, pp. 215-242.
- Shao, S.D.; Lo, E.Y.M.** (2003): Incompressible SPH Method for Simulating Newtonian and Non-Newtonian flows with a Free Surface, *Advances in Water Resource*, 26 (7), pp 787-800.
- Sladek, J.; Sladek, V.; Wen, P.H.; Aliabadi, M.H.** (2006): Meshless Local Petrov-Galerkin (MLPG) Method for Shear Deformable Shells Analysis, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 13, No. 2, pp. 103-118.
- Sladek, J.; Sladek, V.; Zhang, Ch.; Tan, C.L.** (2006): Meshless Local Petrov-Galerkin Method for Linear Coupled Thermoelastic Analysis, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 16, No. 1, pp. 57-68.
- Sladek, J.; Sladek, V.; Zhang, Ch.; Solek, P.** (2007): Application of the MLPG to

Thermo-Piezoelectricity, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 22, No. 3, pp. 217-234.

Sladek, J.; Sladek, V.; Solek, P.; Wen, P.H.; Atluri, S.N. (2008): Thermal Analysis of Reissner-Mindlin Shallow Shells with FGM Properties by the MLPG, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 30, No. 2, pp. 77-98.

Sladek, J.; Sladek, V.; Solek, P.; Wen, P.H. (2008): Thermal Bending of Reissner-Mindlin Plates by the MLPG, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 28, No. 1, pp. 57-76.

Sladek, J.; Sladek, V.; Tan, C.L.; Atluri, S.N. (2008): Analysis of Transient Heat Conduction in 3D Anisotropic Functionally Graded Solids, by the MLPG Method, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 32 (3): 161-174.

Vavourakis, V.; Sellountos, E.J.; Polyzos, D. (2006): A comparison study on different MLPG(LBIE) formulations, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 13, No. 3, pp. 171-184.

Wang, K.; Zhou, S.; Nie, Z.; Kong, S. (2008): Natural neighbour Petrov-Galerkin Method for Shape Design/Sensitivity Analysis, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 26, No. 2, pp. 107-122.

Wu, N.J.; Tsay, T.K.; Young, D.L. (2006): Meshless numerical simulation for fully nonlinear water waves, *International Journal for Numerical Methods in Fluids* Vol. 50, pp 219-234.

Yasuda, T.; Mutsuda, H.; Mizutani, N. (1997): Kinematics of overturning solitary waves and their relations to breaker types, *Coastal Engng.* 29, 317-346.

Yan, S.; Ma, Q.W., (2009): QALE-FEM for modelling 3D overturning waves, accepted for publication by *International Journal for Numerical Methods in Fluids*.

Yuan, W.; Chen, P.; Liu, K. (2007): A New Quasi-Unsymmetric Sparse Linear Systems Solver for Meshless Local Petrov-Galerkin Method (MLPG), *CMES: Computer Modeling in Engineering & Sciences*, Vol. 17, No. 2, pp. 115-134.

Zhang, S.; Morita, K.; Kenji, F; Shirakawa, N. (2006): An Improved MPS Method for Numerical Simulations of Convective Heat Transfer Problems, *International Journal for Numerical Methods in Fluids*, pp 51: 31-47.

