

MLPG Method Based on Rankine Source Solution for Modelling 3D Breaking Waves

J.T. Zhou¹ and Q.W. Ma^{1,2}

Abstract: In this paper, the Meshless Local Petrov-Galerkin method based on Rankine source solution (MLPG_R) is further developed to model 3D breaking waves. For this purpose, the technique for identifying free surface particles called Mixed Particle Number Density and Auxiliary Function Method (MPAM) and the semi-analytical technique for estimating the domain integrals for 2D cases are extended to 3D cases. In addition, a new semi-analytical technique is developed to deal with the local spherical surface integrals. The numerical results obtained by the newly developed method will be compared with experimental data available in literature and satisfactory agreement will be shown.

Keywords: MLPG_R method, MPAM, 3D breaking wave, wave impact, semi-analytical technique

1 Introduction

Wave breaking is the phenomenon that can be seen in ocean every day. During storms or natural disasters such as tsunamis, breaking waves can be formed and be very dangerous. As a result, this phenomenon has been of a great concern when designing marine structures, for instance, breakwaters, offshore wind energy structures, offshore oil and gas exploitation platforms and transport vehicles. In order to design the structures that can withstand the severe action of breaking waves, one needs to quite accurately estimate the forces on them due to the waves, which requires a good understanding of the interaction between the breaking waves and structures. In reality, these interactions are three dimensional (3D). Wave breaking and its interaction with structures have been an important topic of research in the coastal/offshore engineering and environmental communities for many years. Nevertheless, it still remains a great challenge due to its high nonlinearity and com-

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

² Communication: q.ma@city.ac.uk

plexities, particularly under 3D case scenarios.

As reviewed in Ma & Zhou (2009), some investigations have been carried out by the laboratory experiments or field observations [Bonmarin, (1989); Rapp and Melville, (1990); Li and Raichlen, (2003)] and produced very useful and reliable results for some cases but are generally very expensive. Many studies on breaking waves have also been attempted by using 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), Ma and Yan (2009), Yan, and Ma, (2009)], boundary element method [e.g. Grilli, Guyenne and Dias, (2001)] and finite volume method [e.g. Greaves (2009); Klessfman, Fekken, Veldman, Iwanowski, Buchner (2005); Devrard, Marcer, Grilli, Fraunie and Rey, (2005), Ingram, Gao, Causon, Mingham and Troch, (2009), Qian, Causon, Mingham, and Ingram, (2006), Qian, Causon, Ingram, and Mingham, (2003), Yang, Causon, and Ingram (2000), Lv, Zou & Reeve (2009)]. 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 task, particularly in the cases with 3D 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, a number of 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); Zhang, Morita, Kenji and Shirakawa (2006)], 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 (2010) made a review on how to deal with turbulent models in the SPH method. Most of them dealt with 2D problems due to prohibitive computational request of 3D cases.

In addition to the meshless methods mentioned above, 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 (2010) and Ma and Zhou (2009). This is a true meshless method without need of any background mesh. It is based on a local weak form over local 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)], beam and plate bending problems [Atluri and Zhu (2000)], 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)] and some unsteady problems without free surface [Avila and Atluri (2009)]. Over the past couple of years, the method has been extended to deal with more advanced problems. Li and Atluri (2008) and Zheng, Long, Xiong, and Li, (2009) 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. Sellountos, Sequeira and Polyzos (2009) performed elastic transient analysis using the MLPG(LBIE) method. Sladek, Sladek, Wünsche and Zhang (2009) solved the interface crack problem between two dissimilar anisotropic elastic solids using the MLPG method. Bergamaschi, Martinez and Pini (2009) developed the efficient parallel code based on the method to tackle the axi-symmetric elastic problems.

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 modeling 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 does 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 integrals involved in this method, which dramatically reduce the CPU time spent on the numerical evaluation of the integrals.

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 the 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)].

More recently, the method has been developed to be able to deal with 2D breaking wave problems by Ma & Zhou (2009). To tackle the challenges associated with breaking waves, some new techniques were proposed. Specifically, the pressure governing equation was modified, which included not only the velocity divergence term but also the fluid density term. Numerical tests showed that this form of the governing equation is more suitable for simulating breaking waves than the one used in Ma (2005b). In addition, a new technique for identifying free surface particles, called Mixed Particle Density and Auxiliary Function Method (MPAM) was suggested, which proved to be able to robustly identify free surface particles. This method has been applied to successfully model various 2D breaking wave problems.

In this paper, the MLPG_R method will be further extended to deal with 3D breaking waves. The new techniques described in the previous paper (Ma and Zhou, 2009) will be extended to 3D cases. In addition, the new semi-analytical technique for estimating the spherical surface integrals in discretised pressure equations will be proposed to accelerate the computation. This newly extended method will be employed to simulate a couple of the 3D problems. The results obtained will be compared with experimental data available in the public domain to show its accuracy.

2 Governing equations and numerical method

The incompressible Navier-Stokes equation (referred as NS equation) and continuity equation together with proper boundary conditions are applied. In the fluid domain,

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

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

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 and pressure satisfy,

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

$$\vec{n} \cdot \nabla p = \rho \left(\vec{n} \cdot \vec{g} - \vec{n} \cdot \vec{U} + \nu \vec{n} \cdot \nabla^2 \vec{u} \right), \quad (3b)$$

where \vec{n} is the unit normal vector of the rigid boundary and \vec{U} is its velocity and \vec{U} is its acceleration. Eq. (3b) can be easily derived by multiplying $\vec{n} \cdot$ on both the side of Eq. (2). On the free surface, the condition is specified by

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

The mathematical model formed by Eqs (1) to (4) is solved using a time marching procedure. For the details of procedure, readers are referred to Ma (2005a) and Ma & Zhou (2009). Only a summary is given 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

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

where α is a coefficient between 0 and 1. ρ^{n+1} and ρ^* are the fluid densities at $(n+1)$ time step and intermediate fluid density, respectively. Based on our previous numerical tests (Ma & Zhou, 2009), $\alpha=0.1$ can yield satisfactory numerical results for violent breaking cases.

3. Calculate the fluid velocity and therefore update the position of the particles using

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

$$\vec{u}^{n+1} = u^* + \vec{u}^{**} = \vec{u}^* - \frac{\Delta t}{\rho} \nabla p^{n+1}, \tag{8b}$$

$$\vec{r}^{n+1} = \vec{r}^n + \vec{u}^{n+1} \Delta t. \tag{9}$$

4. Go to (1) for the next time step.

As indicated in Ma and Zhou (2009), one must compute the term $\nabla^2 \vec{u}$ when applying the boundary condition in Eq. (3b) for solving the pressure. This 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. (3a) and (8b) are combined to give an alternative equation for the boundary condition as follows:

$$\vec{n} \cdot \nabla p^{n+1} = \frac{\rho}{\Delta t} \vec{n} \cdot (\vec{u}^* - \vec{U}^{n+1}). \tag{10}$$

3 MLPG_R formulation and new numerical techniques

As can be seen, the key task in the above procedure is to solve Eq. (7) together with Eq. (4) and Eq. (10) in order to evaluate the pressure and its gradient at $(n + 1)$ time step. For solving the equation, the MLPG_R method is used.

In the MLPG_R method, the whole computational domain is discretized by particles. The particles are separated into three groups: those located on the rigid boundary (referred as wall particles), those on the free surface (referred as free surface particles) and others (referred as inner particles). At each inner particle, a sub-domain is specified, which is a sphere in 3D cases. In the MLPG_R formulation, Eq. (7) is first changed into another form by a weighted integration. The details about how to do so are very similar to those for 2D cases given in Ma & Zhou (2009). Therefore, details will not be described in this paper. Only the final equation is given as follows:

$$\int_{\partial\Omega_I} \vec{n} \cdot (p \nabla \varphi) dS - R_I p = \alpha \frac{\rho}{\Delta t^2} \frac{n_0 - n_I^*}{n_0} \frac{R_I^3}{6} + (1 - \alpha) \int_{\Omega_I} \frac{\rho}{\Delta t} \vec{u}^* \cdot \nabla \varphi d\Omega, \tag{11}$$

where φ is a test function, R_I is the radius of the local domain Ω_I , which is a sphere in 3D cases, and the $\partial\Omega_I$ is the spherical surface of the local domain. n_I^* in Eq. (11) is the intermediate particle number density defined by $n_I^* = \sum_{j=1, j \neq I} W(|r_j^* - r_I^*|)$, where $W(|r_j^* - r_I^*|)$, chosen as the spline function [Ma, 2005b], 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. n_0 is the initial value of the particle number density. In the MLPG_R method, the solution of Rankine source is taken as the test function. i.e., the function φ satisfies $\nabla^2 \varphi = 0$, in Ω_I except for the center and $\varphi = 0$, on $\partial\Omega_I$. The expression of the solution for Rankine source is

$$\varphi = \frac{1}{4\pi}(1 - R_I/r) \text{ for 3D cases,} \quad (12)$$

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

The method to discretize Eq. (11) is very similar to 2D cases and therefore only brief description will be given in the following section for completeness.

3.1 Discretization of the governing equation for pressure

The unknown function p is approximated by a set of discretized variables. Generally, the approximation may be written as

$$p(\vec{x}) \approx \sum_{j=1}^N \Phi_j(\vec{x}) \hat{p}_j, \quad (13)$$

where N is the number of nodes that affect the pressure at point \vec{x} ; \hat{p}_j are nodal variables but not necessarily equal to the nodal values of $p(\vec{x})$; and $\Phi_j(\vec{x})$ the shape function which is formed by using the moving least square method as described in Ma (2005a, b) and Ma (2007).

Inserting Eq. (13) into Eq. (11), it follows that:

$$\mathbf{K} \cdot \hat{\mathbf{P}} = \mathbf{F}, \quad (14)$$

where $\hat{\mathbf{P}}$ is the vector formed by the nodal value of pressure \hat{p}_j and the elements of Matrixes \mathbf{K} and \mathbf{F} are given by

$$K_{IJ} = \int_{\partial\Omega_I} \Phi_J(\vec{x}) \vec{n} \cdot \nabla \varphi dS - R_I \Phi_J(\vec{x}), \quad (15a)$$

$$F_I = \alpha \frac{\rho}{\Delta t^2} \frac{n^0 - n^*}{n^0} \frac{R_I^3}{6} + (1 - \alpha) \int_{\Omega_I} \frac{\rho}{\Delta t} \vec{u}^{(*)} \cdot \nabla \varphi d\Omega, \quad (15b)$$

where Particle I is an inner water particle; and Particle J are those influencing Particle I , determined by the weight function. Using Eq. (12), the surface integral

over the spherical surface of the local domain in Eq. (15a) can be rewritten as:

$$\int_{\partial\Omega_I} \Phi_J(\vec{x}) \vec{n} \cdot \nabla \varphi dS = \frac{R_I}{4\pi} \int_0^{2\pi} \int_0^\pi \Phi_J(\vec{x}) \sin \theta d\theta d\gamma. \tag{16}$$

The domain integral over the sphere in Eq. (15b) can be simplified as:

$$\int_{\Omega_I} \frac{\rho}{\Delta t} (\vec{u}^{(*)} \cdot \nabla \varphi) d\Omega = \frac{\rho R_I}{4\pi \Delta t} \int_0^{R_I} \int_0^{2\pi} \int_0^\pi u_r^{(*)} \sin \theta dr d\theta d\gamma, \tag{17}$$

where $u_r^{(*)}$ is the radial component of $\vec{u}^{(*)}$; θ and γ are defined in Fig. 1.

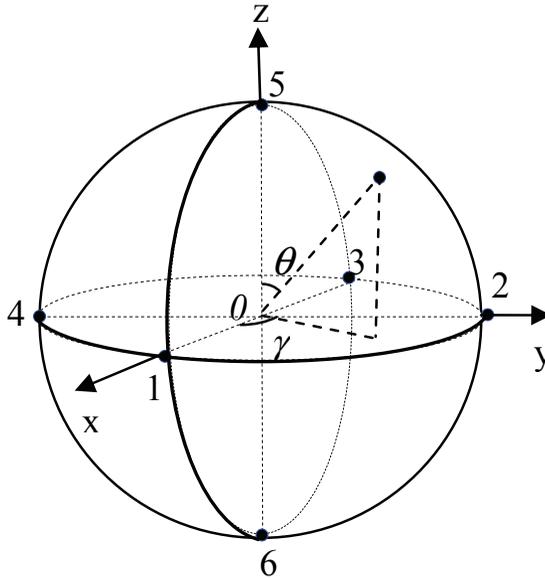


Figure 1: Illustration of division of an integration domain

3.2 Numerical technique for evaluating domain integrals

The domain integral related to velocity in Eq. (17) must be numerically evaluated, perhaps by using the Gaussian quadrature, in which the local sphere domain may be split into 8 quadrants for 3D cases. For the whole local domain, one may need $8M^3$ Gaussian points, where M is the number of the Gaussian point corresponding

to each variable of the integral in one quadrant. M must be larger or equal to 2 and so at least 64 Gaussian points for 3D cases are required to obtain satisfactory results, at which intermediate velocities are estimated. Evaluation of the velocities at so many points is time-consuming, especially in 3D cases with a huge number of particles. In order to make the method more efficient, the following semi-analytical technique has been suggested:

- Dividing an integration domain into several sub-domains;
- Assuming intermediate velocity to linearly vary over each sub-domain;
- Performing the integration over each sub-domain analytically.

In our previous papers (Ma, 2005a, b, Ma & Zhou, 2009), this semi-analytical technique has been applied to 2D problems and proved to be accurate and efficient. This technique will be extended to 3D cases here. For this purpose, let us consider a spherical domain with a radius of R_I , which is centred at (x_0, y_0, z_0) (Fig. 1) and divided into 8 sub-domains. Over each sub-domain, e.g. 0-1-2-5, the intermediate velocity components are assumed to be linear with respect to coordinates and given by

$$u^{(*)} = u_0^{(*)} + C_{ux} \frac{x-x_0}{R_I} + C_{uy} \frac{y-y_0}{R_I} + C_{uz} \frac{z-z_0}{R_I}, \quad (18a)$$

$$v^{(*)} = v_0^{(*)} + C_{vx} \frac{x-x_0}{R_I} + C_{vy} \frac{y-y_0}{R_I} + C_{vz} \frac{z-z_0}{R_I}, \quad (18b)$$

$$w^{(*)} = w_0^{(*)} + C_{wx} \frac{x-x_0}{R_I} + C_{wy} \frac{y-y_0}{R_I} + C_{wz} \frac{z-z_0}{R_I}, \quad (18c)$$

where $(u^{(*)}, v^{(*)}, w^{(*)})$ are the intermediate velocity components at a point (x, y, z) within the sub-domain 0-1-2-5; and $(u_0^{(*)}, v_0^{(*)}, w_0^{(*)})$ are those at its centre. $C_{ux}, C_{uy}, C_{uz}, C_{vx}, C_{vy}, C_{vz}$ and C_{wx}, C_{wy}, C_{wz} are constants, which are determined in such a way that the velocity components equal to those at Point 1, 2 and 5. Taking the x-component as an example, one should have

$$\begin{aligned} C_{ux}(x_1 - x_0) + C_{uy}(y_1 - y_0) + C_{uz}(z_1 - z_0) &= (u_1^{(*)} - u_0^{(*)})R_I \\ C_{ux}(x_2 - x_0) + C_{uy}(y_2 - y_0) + C_{uz}(z_2 - z_0) &= (u_2^{(*)} - u_0^{(*)})R_I \\ C_{ux}(x_5 - x_0) + C_{uy}(y_5 - y_0) + C_{uz}(z_5 - z_0) &= (u_5^{(*)} - u_0^{(*)})R_I \end{aligned} \quad (19a)$$

which yields

$$C_{ux} = u_1^{(*)} - u_0^{(*)},$$

$$C_{uy} = u_2^{(*)} - u_0^{(*)}, \tag{19b}$$

$$C_{uz} = u_5^{(*)} - u_0^{(*)}.$$

C_{vx}, C_{vy}, C_{vz} and C_{wx}, C_{wy}, C_{wz} can be found similarly or obtained by replacing u with v or w in Eq. (19). Thus the velocities ($u^{(*)}, v^{(*)}, w^{(*)}$) in this sub-domain are given by inserting Eq. (19) into Eq. (18). The velocities in other sub domains can also be estimated in this way. The only difference is that they may be related to the velocities at Points 3, 4 and 6, depending on which sub-domain is concerned with. Consequently, the velocities at any point in the spherical local domain are determined by those at only 7 points (0, 1, 2, 3, 4, 5 and 6). The relationship between $u_r^{(*)}$ and ($u^{(*)}, v^{(*)}, w^{(*)}$) is:

$$u_r^{(*)} = u^{(*)} \sin \theta \cos \gamma + v^{(*)} \sin \theta \sin \gamma + w^{(*)} \cos \theta. \tag{20}$$

Based on this, the integral in Eq. (17) is rewritten as:

$$\int_0^{R_I} \int_0^{2\pi} \int_0^\pi u_r^{(*)} \sin \theta dr d\theta d\gamma = \sum_{j=1}^2 \sum_{i=1}^4 \int_{\theta_j}^{\theta_{j+1}} \int_{\gamma_i}^{\gamma_{i+1}} \int_0^{R_I} u_r^{(*)} \sin \theta dr d\theta d\gamma, \tag{21}$$

which includes 8 integrals on the right hand side, one for each sub-domain. Inserting Eqs. (18) to (20) into the integral for the sub-domain 0-1-2-5 in Eq. (21), it follows that

$$\begin{aligned} \int_0^{R_I} \int_0^{2\pi} \int_0^{\pi/2} u_r^{(*)} \sin \theta dr d\theta d\gamma &= \frac{\pi}{4} R_I (u_0^{(*)} + v_0^{(*)} + w_0^{(*)}) + \frac{\pi}{12} R_I (C_{ux} + C_{vy} + C_{wz}) \\ &+ \frac{1}{6} R_I (C_{uy} + C_{uz} + C_{vx} + C_{vz} + C_{wx} + C_{wy}). \end{aligned} \tag{22}$$

Similarly the domain integral over other 7 sub-domains can be obtained in this way. Added all the results together, the integral over the whole local domain in Eq. (21) is given by

$$\int_0^{R_I} \int_0^{2\pi} \int_0^\pi u_r^{(*)} \sin \theta dr d\theta d\gamma = \frac{\pi}{3} R_I (u_1^{(*)} + v_2^{(*)} + w_5^{(*)} - u_3^{(*)} - v_4^{(*)} - w_6^{(*)}). \tag{23}$$

It can be seen that the integral over each local spherical domain is determined by the velocities only at six points rather than at least at 64 Gaussian points if the Gaussian quadrature would be employed.

3.3 Numerical technique for evaluating surface integrals

The integral of Eq. (16) regarding the shape function is an integral over a curve in 2D cases, which was estimated by using Gaussian quadrature in our previous publications for 2D problems. For 3D cases, if we do the same, it will need $8M^2$ Gaussian points assuming the spherical surface is divided into 8 quadrants, where M is the number of Gaussian points corresponding to each variable of the integral in one quadrant. Later, we will demonstrate that M needs to be at least 3, indicating that at least 72 Gaussian points over the whole spherical surface of the local domain are required, which would be very time consuming if the Gaussian quadrature is still used. Therefore, a new numerical technique is developed to evaluate the spherical surface integral in Eq. (16), which is similar to that for evaluating the domain integral described above but based on a different approximate function. To introduce the technique, the integral in Eq. (16) is rewritten as

$$\int_0^{2\pi} \int_0^{\pi} \Phi \sin \theta d\theta d\gamma = \sum_{i=1}^4 \sum_{j=1}^2 \int_{\theta_j}^{\theta_{j+1}} \int_{\gamma_i}^{\gamma_{i+1}} \Phi \sin \theta d\theta d\gamma, \quad (24)$$

where the subscript of Φ is dropped without affecting its meaning. That is, the integration is first performed over each of 8 quadrants and then the results are added together, similar to what have been done for obtaining Eq. (23).

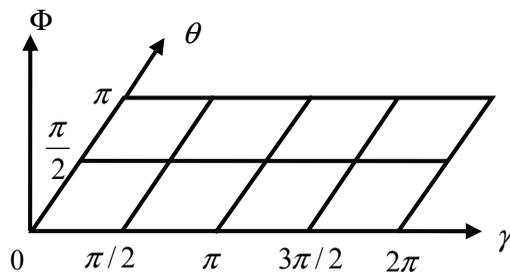


Figure 2: Illustration of integration domain in θ - γ plane

The shape function Φ in the integrand in Eq. (24) is the function of θ - γ and thus it may be transformed to the θ - γ plane that may be split into 8 patches as illustrated in Fig. 2. It is noted, however, that the shape function Φ is the same for any values of γ when $\theta = 0$ or $\theta = \pi$ because it corresponds to a single point 5 or 6 in Fig. 1. Consider the shape function Φ in the quadrant 1-2-5 in Fig. 1 or the patch near the

origin in Fig. 2, we may approximate it by using

$$\Phi = A + B\theta + C\gamma + D\theta\gamma, \tag{25}$$

where A, B, C and D are constants. They can be determined by using the fact that the shape function should be equal to Φ_5 at $\theta = 0$ and $\gamma = 0$ or $\gamma = \pi/2$; equal to Φ_1 at $\theta = \pi/2$ and $\gamma = 0$; and equal to Φ_2 at $\theta = \pi/2$ and $\gamma = \pi/2$, where the subscripts of Φ refer to the number in Fig. 1, that is

$$\Phi_5 = A, \tag{26a}$$

$$\Phi_5 = A + C\pi/2, \tag{26b}$$

$$\Phi_1 = A + B\pi/2, \tag{26c}$$

$$\Phi_2 = A + B\pi/2 + C\pi/2 + D(\pi/2)^2. \tag{26d}$$

It follows that

$$A = \Phi_5, \tag{27a}$$

$$C = 0, \tag{27b}$$

$$B = \frac{2}{\pi}(\Phi_1 - \Phi_5), \tag{27c}$$

$$D = \frac{4}{\pi^2}(\Phi_2 - \Phi_1), \tag{27d}$$

Inserting Eq. (27) into Eq. (25) yields

$$\Phi = \Phi_5 + \frac{2}{\pi}(\Phi_1 - \Phi_5)\theta + \frac{4}{\pi^2}(\Phi_2 - \Phi_1)\gamma\theta, \tag{28}$$

which is the approximate expression of the shape function in the first sub-domain 1-2-5.

The integral of Eq. (24) over the sub-domain 1-2-5 is then given by

$$\int_0^{\pi/2} \int_0^{\pi/2} \Phi \sin \theta d\theta d\gamma = \frac{1}{2}\Phi_1 + \frac{1}{2}\Phi_2 + \left(\frac{\pi}{2} - 1\right)\Phi_5. \tag{29a}$$

The integrals over other sub-domains can be obtained in the similar way and the sum of these results give the expression of the integral over the whole spherical surface domain:

$$\int_0^{2\pi} \int_0^{\pi} \Phi \sin \theta d\theta d\gamma = 2(\Phi_1 + \Phi_2 + \Phi_3 + \Phi_4) + (2\pi - 4)(\Phi_5 + \Phi_6). \tag{29b}$$

It is noted that the integral in Eq. (29b) is only related to the shape function at six points, compared with at least 72 points if the Gaussian quadrature is employed. Therefore, the CPU time spent on evaluation of the integral will be dramatically reduced, which will be demonstrated in the later section of the paper.

3.4 Identification of free surface particles in 3D cases

When solving the governing equation for the pressure, all the particles on boundaries should be known, including the solid wall boundary and the free surface. The particles on rigid walls always remain on them. However, the same does not hold for those on the free surface when simulating breaking wave cases. In these cases, the free surface particles can immerse into the inner fluid domain and the inner water particle can also emerge to become free surface particles. Consequently, it is necessary to identify which particle is on the free surface at each time step when modelling breaking waves.

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:

$$\beta_I = n_I^*/n^0, \quad (30)$$

where n_I^* is the particle number density at Particle I , as defined above, 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. 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 promising 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).

A new numerical technique to identify the free surface particles has been presented and applied to 2D cases in Ma & Zhou (2009), which called Mixed Particle Number Density and Auxiliary Function Method (MPAM). The basic idea of this technique is based not only on how many neighbour particles (denoted by Eq. (30)) there are but also on how they are distributed. This technique will be extended to deal with 3D cases for violent breaking waves in this paper.

In 3D cases, the auxiliary functions involved in the MPAM method are defined in a 3D local domain as shown in Fig. 3. The details are as follows:

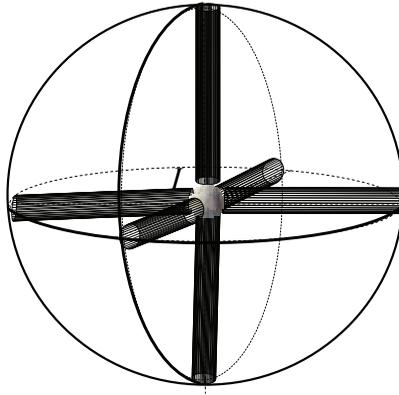


Figure 3: Local domain at Node I for the definition of auxiliary function (the inner sphere denotes integration domain of the particle; the outer sphere denotes the influence domain; the 6 coloured cylinders have the same diameter as the inner sphere)

The first auxiliary function is defined by

$$f_{sp_a}(I) = \begin{cases} 1, & NumA \geq 1 \\ 0, & NumA = 0 \end{cases} \quad (31)$$

where $NumA$ represents the number of the free surface particles existing in the influence domain (D_f) of Node I in previous time step. The second auxiliary function is

$$f_{sp_b}(I) = \begin{cases} 1, & NumB = 8 \\ 0, & NumB \leq 7 \end{cases} \quad (32)$$

where $NumB$ represents the number of quadrants occupied by the neighbor particles of Node I as shown in Fig. 3.

The third auxiliary function is given by

$$fsp_c(I) = \begin{cases} 1, & NumC = 6 \\ 0, & NumC \leq 5 \end{cases}, \quad (33)$$

where $NumC$ represents the number of colored cylinders in Fig. 3 in which there is at least one fluid particle.

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

- (a) no inner particle in D_f except for I ;
- (b) $\beta_I < \beta$ and $fsp_a(I) = 1$;
- (c) $\beta_I \geq \beta$, $fsp_a(I) = 1$ and $fsp_b(I) = 0$;
- (d) $\beta_I \geq \beta$, $fsp_a(I) = 1$ 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. In 2D cases, β is taken as 0.97; but in 3D cases, the value of β is slightly adjusted to 0.9, based on numerical tests for 3D cases.

4 Validation and discussions

In the above sections, the semi-analytical technique for evaluating the domain integrals and the method for identifying the free surface developed for 2D simulation are extended to 3D cases. A new semi-analytical technique for evaluating the surface integrals is developed to accelerate the discretization of the governing equation for pressure. In the sections below, it will be shown that the new technique works effectively and the MLPG_R method as a whole can yield results that have acceptable agreement with experimental data in the cases considered.

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 initial water depth d , the time by $\sqrt{d/g}$ and the pressure by $\rho g d$, unless mentioned otherwise.

4.1 Effectiveness of the semi-analytical technique for surface integrals

As pointed above, our main aim of developing this semi-analytical technique for evaluating the surface integral given in Eq. (29) is to reduce the computational time while maintaining the same or similar accuracy of results. This will be demonstrated in this section by modelling two examples by using the Gaussian quadrature and the semi-analytical technique for estimating the surface integral with all others being the same. In the first example, we consider a static case in which the solution for pressure is just the hydrostatic pressure. To this end, a tank with the height, length and width being 1, 3 and 1, respectively, is used. A schematic view of the tank is shown in Fig. 4, where grey particles and blue particles represent wall particles and water particles, respectively. To make it clearer, only those of boundary particles on the right wall, the left wall and the bottom are plotted in the Fig. 4. The particles are uniformly distributed initially in all the directions. The number of particles along z-direction is represented by N_z . N_z is equal to 10 in this case. The total water particles are 2900 excluding the particles on boundaries and 5260 including those on the boundaries.

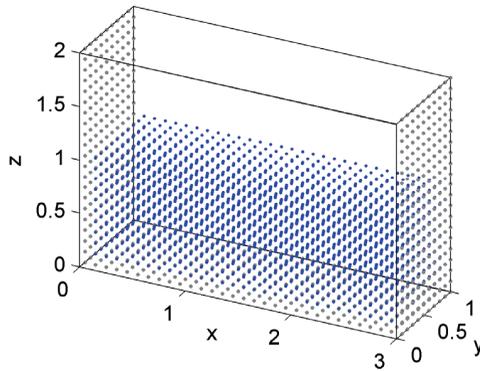


Figure 4: Schematic view of the tank and initial distribution of particles

Figs. 5a, b and c present the pressure distribution along a vertical line in the middle of the tank after running 200, 400 and 600 time steps, respectively. As can be seen, Fig. 5a is the results obtained by using the semi-analytical technique (Eq. 29b), Figs. 5b and 5c are those by using the Gaussian quadrature but different Gaussian points. It can be seen from these figures that the results are almost the same, which indicates that the semi-analytical technique produces the same results as the Gaussian quadrature. The CPU time required for forming Matrix \mathbf{K} and \mathbf{F}

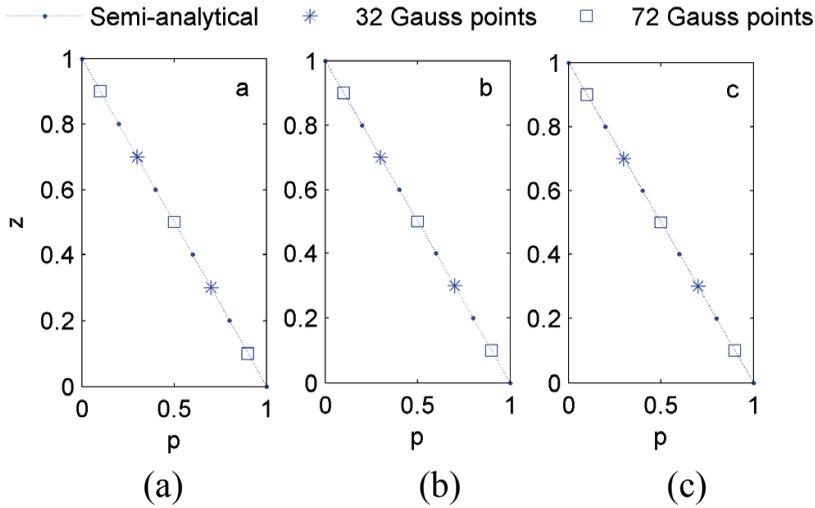


Figure 5: Comparison of static pressure obtained by using the Gaussian quadrature and the semi-analytical technique for estimating the surface integrals

Table 1: Comparison of CPU time required by using different methods to evaluate the surface integrals (GQ: Gaussian quadrature) for a static case

Method	Semi-analytical	32 Gauss points (GQ)	72 Gauss points (GQ)
Average CPU time in one time step	1	5.67	11.17

in Eq. (14) in the cases shown in Fig. 5 is summarised in Table. 1, where the CPU time required by the Gaussian quadrature is divided by the CPU time required by the semi-analytical technique. This table shows that the CPU time required by the Gaussian quadrature is at least 5 times longer than that required by the semi-analytical technique.

In the second example, the propagation of a solitary wave is modelled, again by using the semi-analytical technique and Gaussian quadrature with 32 ($M = 2$) and 72 ($M = 3$) Gaussian points, respectively. The solitary wave is generated by a piston-type wavemaker according to the theory given by Goring (1978), in which the motion of the wavemaker 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

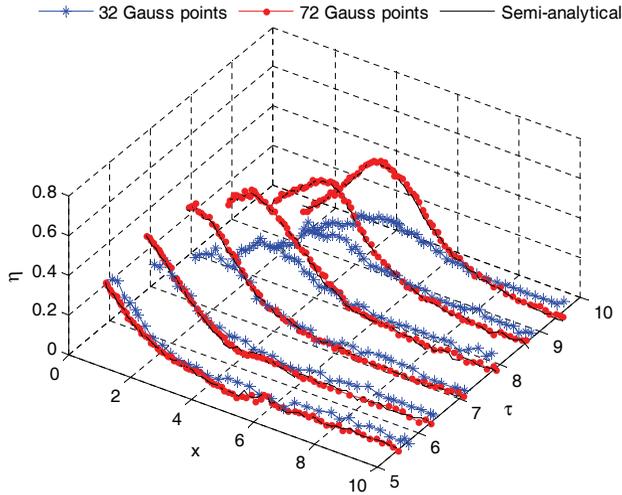


Figure 6: Comparison of wave profiles at different instants obtained by using the semi-analytical technique and the Gaussian quadrature with 32 and 72 Gaussian points, respectively.

the dimensionless celerity $c = \sqrt{(1+h)}$. The numerical water tank and coordinate system are shown in Fig. 7 and the wave height targeted is $h = 0.45$. For this case, the particle number along z-axis, i.e., N_z , is 20, which yields 89,492 particles totally. These cases are run up to 500 time steps with the length of 0.02 each step. Wave profiles at 6 different instants are compared in Fig. 6. From this figure, one may see that the results of the Gaussian quadrature with 32 Gaussian points are very different from these of the Gaussian quadrature with 72 Gaussian points. The later is very similar to those obtained by using the semi-analytical technique. This indicates that at least 72 ($M=3$) Gaussian points are required to compute the integrals in Eq. (16) in this case. It is highly likely that such a number is needed for modelling other 3D violent waves, which has been confirmed for the cases we considered so far. This also indicates that the semi-analytical technique can produce the results that are as accurate as the Gaussian quadrature. Similar comparison of CPU time to that shown in Table 1 is made in Table 2. This table demonstrates that the CPU time required by the semi-analytical technique is only 1/17.65 of that required by the Gaussian quadrature to achieve acceptable results. Although the CPU time required for the case with 32 Gaussian points is only half of that with 72, the accuracy of the former is not satisfactory as demonstrated in Fig. 6.

Investigations on these static and dynamic cases demonstrate that the semi-analytical technique spends only a small fraction of the CPU time spent by the Gaussian

Table 2: Comparison of CPU time required by using different methods to evaluate the surface integrals (GQ: Gaussian quadrature) for a dynamic case

Method	Semi-analytical	32 Gauss points (GQ)	72 Gauss points (GQ)
Average CPU time in one time step	1	8.57	17.65

quadrature to achieve the results with similar accuracy. Because of this advance, we can model the 3D breaking waves on a normal PC within a reasonable time, which has proved to be impossible if using the Gaussian quadrature.

4.2 Parameter study and comparison with experimental data

In order to show the robustness of the MLPG_R for modelling 3D breaking waves, two cases are presented in this section. In both cases, the numerical results will be compared with experimental data. The convergent properties will also be considered for one of the two cases.

The first case to be considered is about a wave breaking over slope that is the same as the one used in the previous section. However, the attention here is focused on how the results are affected by choosing different computational parameters and how they are correlated with the experimental data. In order to further save the CPU time for this 3D case, computational process is split into two stages. The first stage starts from time =0 and ends at the time when the wave propagates just to the toe of the slope ($x=10$ in Fig. 7). The wave propagation in this stage is simulated by using the 2D model. The second stage continues from the first stage, in which the 3D model is employed. It is noted that the 3D model can be used for the entire process, though it may need a longer CPU time.

In 2D simulations discussed by Ma & Zhou, (2009), two aspects of the convergence property have been investigated: one is how the ratio $\Delta x/\Delta \tau$ affects the numerical results, where Δx is the initial particle distance and $\Delta \tau$ is the time step; and the other is how the results vary by changing the value of Δx with $\Delta x/\Delta \tau$ fixed. It has been concluded that a suitable value for $\Delta x/\Delta \tau$ is 5 in studies on 2D cases in the cited paper. Similar investigations have also been made for the 3D cases and confirm that $\Delta x/\Delta \tau = 5$ is still suitable. Corresponding results about this will not be presented here because they are quite similar to the 2D cases.

What will be shown are the results obtained by using the three different values of Δx , i.e., 0.05, 0.04 and 0.033 (corresponding to $N_z=20, 25$ and 30 , respectively),

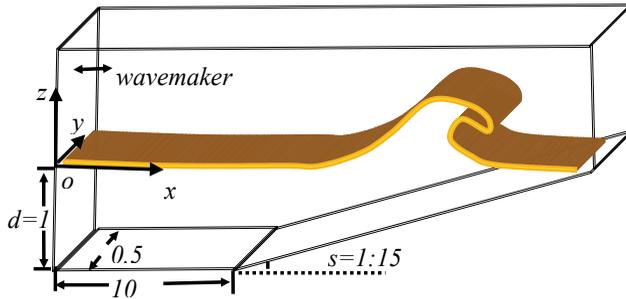


Figure 7: Illustration of model set-up for the solitary wave

with $\Delta x/\Delta \tau = 5$ kept. The resulting wave profiles near the section when overturning occurs at four instants and at three different transverse positions ($y=0$, $y=0.25$ and $y=0.5$) are shown in Fig. 8. The four instants correspond to the instants when the wave becomes very steep, just overturning, a jet and a jet touching the water surface in its front. The experimental data from Li and Raichlen (1998) are also shown in the figure, which should be considered as average wave profiles with respect to y -coordinates because they are drawn from the photograph taken from a side during laboratory experiments.

Fig. 8 shows that at all the four instants the wave profiles at the different sections with different y -coordinates are very similar, as what is expected for this case. The figure also shows that the results corresponding to different Δx values are very close to each other and that all of them agree quite well with the experimental profiles. This demonstrates that the MLPG_R methods with the numerical techniques developed in the paper can yield good results as long as the number of particles is sufficient large.

The wave profiles at the post-breaking stage are plotted in Fig. 9. At this stage, the plunging jets hit the free surface in their front and cause splash, leading to a cavity formed behind the new jet, as one can see from this figure. In addition, the profiles for $\Delta x=0.04$ ($N_z=25$) and $\Delta x=0.033$ ($N_z=30$) are very similar to each other, but have visible discrepancy with those for $\Delta x=0.05$, especially in the shape of the cavities. This indicates that one may need more particles to model the post-breaking waves. It is noted that the profiles at different y -coordinates are not the same, unlike what we have seen in Fig. 8. That is perhaps because some unknown random factors arise due to the turbulence at the post-breaking stage. The similar phenomenon can be observed in laboratory experiments. Further investigations are required to find the reasons in future.

The second case considered is about impact on a 3D obstacle by the dam breaking

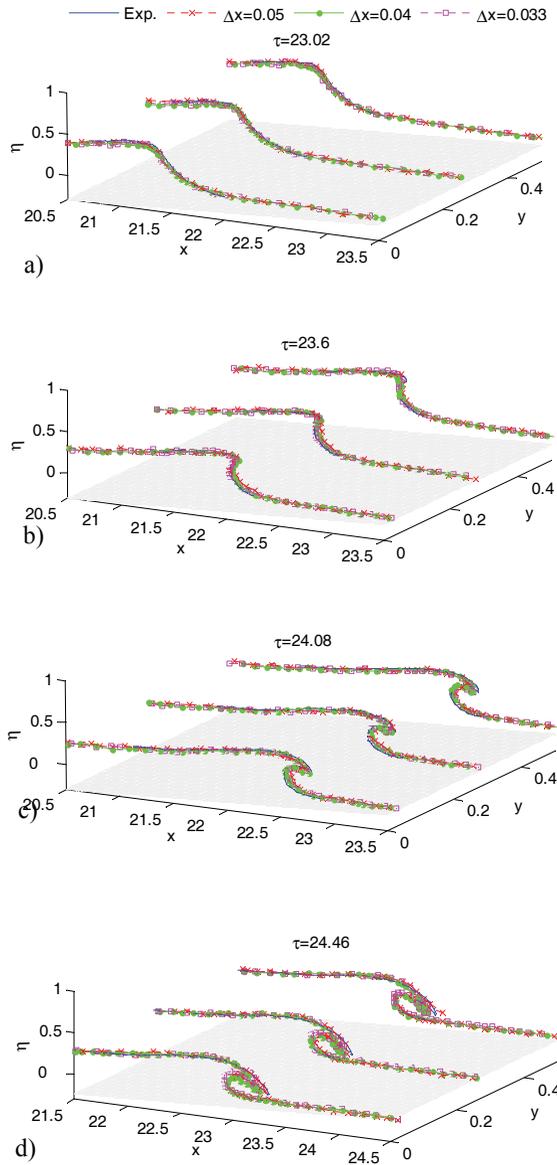


Figure 8: Comparison between experimental wave profiles [Li and Raichlen (1998)] and numerical results obtained by using different values of Δx when $\Delta x/\Delta \tau = 5$

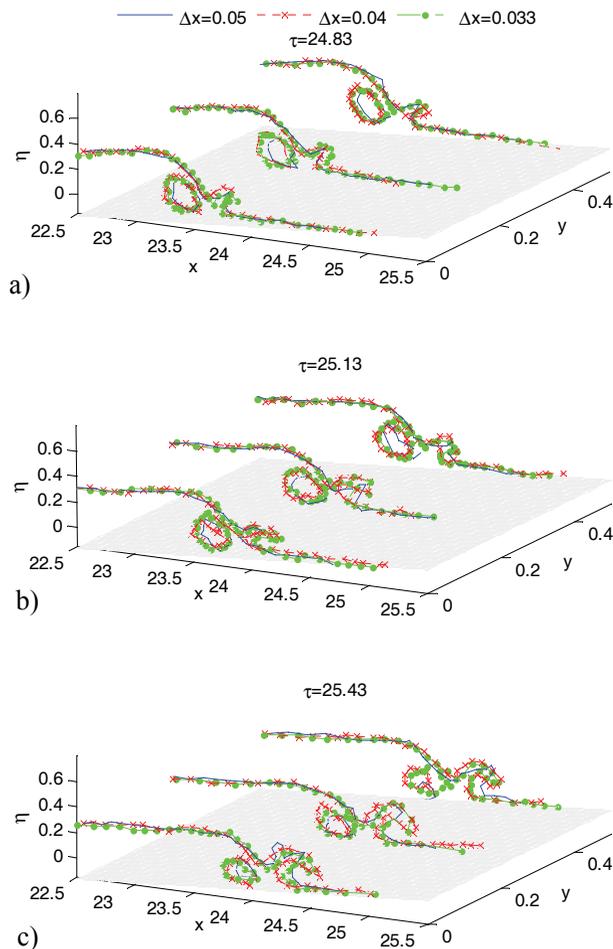


Figure 9: Wave profiles in the post-breaking stage obtained by using different values of Δx when $\Delta x/\Delta \tau = 5$

wave. This case has been used to validate our results in Zhou, Ma, Zhang and Yan (2009). The reason for using the same case here to verify our improved method is because the experimental data related to breaking wave's impact on 3D solid objects are very limited. Nevertheless, the results shown here are much improved than those in the previous paper. The set-up of the case is the same as in Kleefsman, Fekken, Veldman, Iwanowski and Buchner (2005) that presented the experimental results, as illustrated in Fig. 10. For convenience of comparing our results with

theirs, the parameters presented below are all dimensional. Specifically, the size of the rectangle tank in Fig. 10 is 3.22 meter long, 1 meter high and 1 meter wide. The water column, whose length, width and height are 1.228m, 1m and 0.55m, respectively, is initially held to the right end of the tank by a rigid gate. A box-type obstacle is mounted facing the water column. The distance between the centre of the obstacle and the left vertical wall of the tank is 0.744 m. On its surface of the obstacle, a number of pressure transducers are installed in their experiments but only two of them are shown in Fig. 11, where the size of the obstacle is also given. In their experiment, four probes are used to record the time history of the wave height. The data from two of them (H3 and H4 shown in Fig. 10) will be compared with our numerical results. For simulating this case, the initial distance between two particles are taken as 0.022m ($N_z=25$) and the time step as 0.00189s (equivalent dimensionless time step $\Delta\tau = 0.008$) is used according to the above convergent investigation.

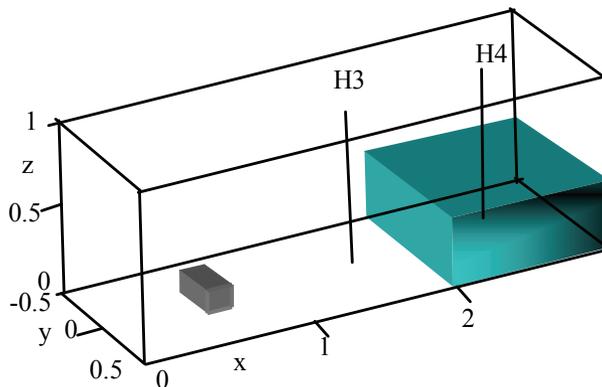


Figure 10: Schematic view of the tank and the obstacle (unit: m). H3 and H4 represent the two wave probes, located at 1.488m and 2.638m from the left wall, respectively.

The numerical simulation starts when the gate is suddenly released. Some snapshots of the free surface profiles are shown in Fig. 12. In this figure different color highlights different water surface height. The figure shows the different stages of the wave evolution, illustrating the flow of water just after collapses (Fig.12a), the collapsed water hitting the obstacle with very high speed (Fig.12b) and generating an impact on the surface of the obstacle, and splash of water after that (Fig.12c). The free surface profiles at two instants $t \approx 0.4$ s and $t \approx 0.56$ s are compared with the experimental photographs and the numerical results by Klefsman, Fekken,

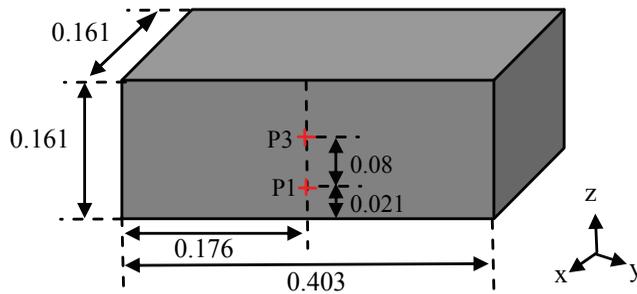


Figure 11: Sizes of the obstacle and the locations of pressure transducers (unit: m)

Veldman, Iwanowski and Buchner (2005) in Fig. 13. Only the free surfaces near the obstacle are plotted. From this figure, it is found that the present results are very similar to the Kleefsman's numerical ones and both the numerical results are also similar to the experimental profiles.

The comparison shown in Fig.13 qualitatively demonstrates an accuracy of the MLPG_R method in predicting the spatial distribution of the free surface elevation when interaction between waves and 3D obstacles is involved. To show its robustness quantitatively, the comparison is also made of the time histories of the wave height recorded at different positions with experimental data in Figs. 14 and 15. Fig. 14 depicts the time history of the surface elevation at Probe H3 while Fig.15 depicts that at H4. It is observed from these figures that our numerical results correlate very well with the experimental data, though some inevitable difference exists.

Apart from the wave profiles, the comparisons of the pressure are given in Figs. 16 and 17, which display the time histories of pressure recorded at Point P1 and P3 on the surface of obstacle (shown in Fig. 11), respectively. From the Figs. 16 and 17, one can notice that there are two pressure rises in each curve. The first pressure rise is generated due to the primary impact on the obstacle after the water column collapses while the second rise occurs when the returning wave front hits the obstacle again from the right after it is reflected by the left and then right walls. The first rise is extreme large and last for a very short period, demonstrating the typical impulsive feature caused by impact. The second rise is not so large and grows relatively slowly. One can also see that the shapes of the numerical curves are largely similar to the experimental ones. Particularly, our numerical method does not only predict the first pressure rise well, but also give a good estimation to the second rise of experimental data. It is noted that although there are still some

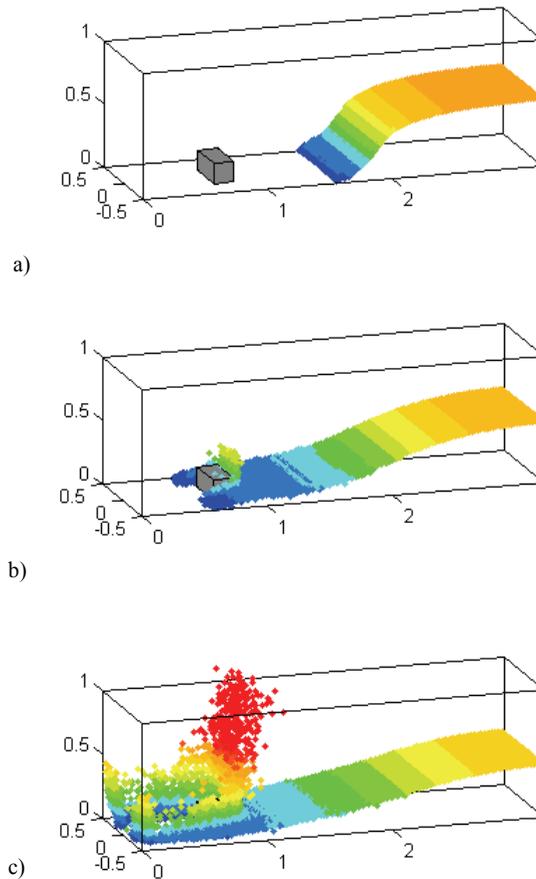


Figure 12: Snapshots of the free surface at (a) $t=0.227s$ (b) $t=0.493s$ and (c) $t=0.701s$.

fluctuations in the time history of pressure, the curves are much smoother than what we presented in Zhou, Ma, Zhang and Yan (2009).

5 Conclusions

In this paper, the MLPG_R method is extended to simulate 3D violent breaking waves. Some technical improvements have been made: (1) extending the MPAM method for identifying the free surface particles into 3D cases; (2) extending the semi-analytical technique for the local domain integration into 3D cases and (3) developing the new semi-analytical technique for computing the spherical surface

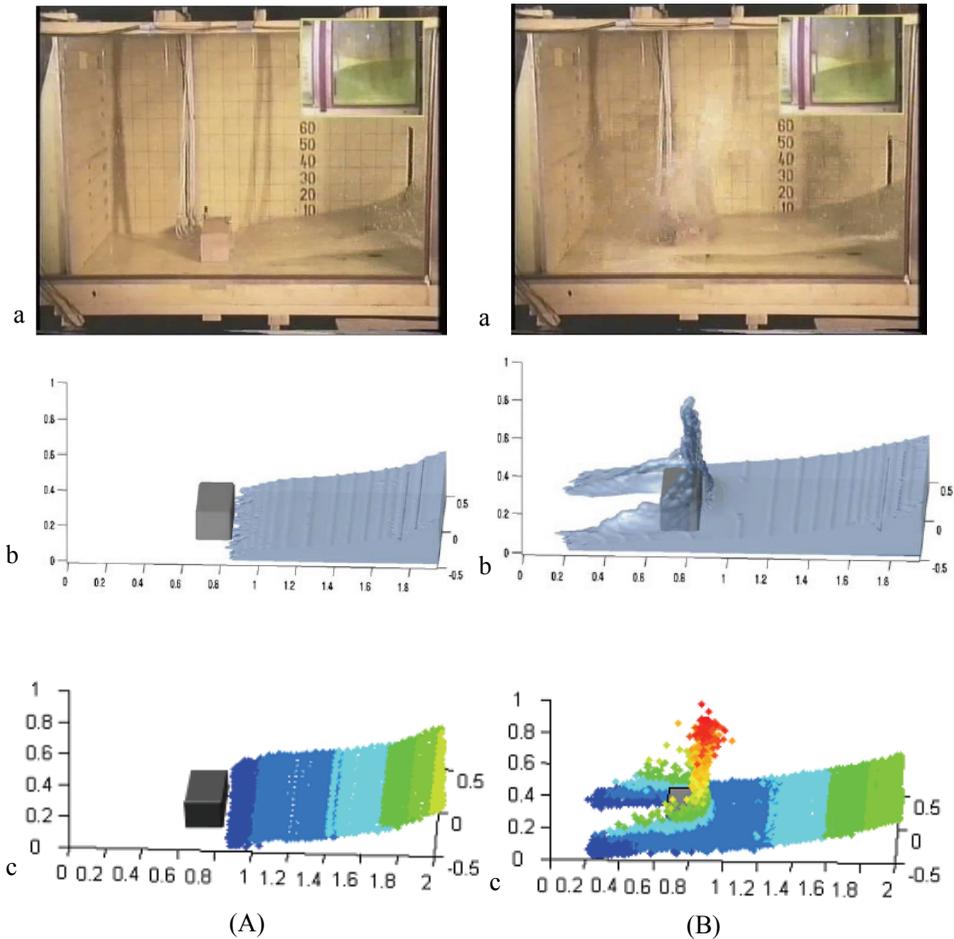


Figure 13: (a) Comparison of the free surface profiles at $t \approx 0.4s$ (a: experiment, b: FVM numerical results, both results from Klessfsman, et al, 2005; c: present method); (b) Comparison of the free surface profiles at $t \approx 0.56s$ (a: experiment, b: FVM numerical results, both results from Klessfsman, et al, 2005; c: present method)

integrals. The semi-analytical techniques can dramatically reduce the CPU time. Without them, it would have taken too long time to run the 3D cases on a normal PC available nowadays.

Comparison of the numerical results produced by the MLPG_R method with experimental data in a couple of violent breaking wave cases shows that the MLPG_R

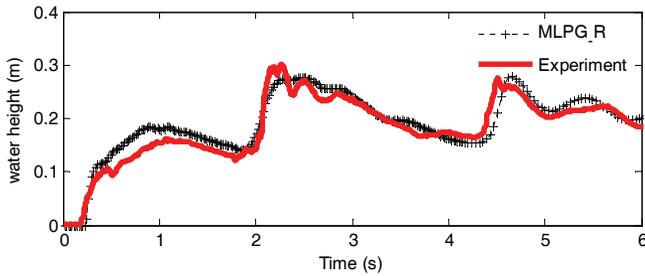


Figure 14: The time histories of the free surface elevation at H3

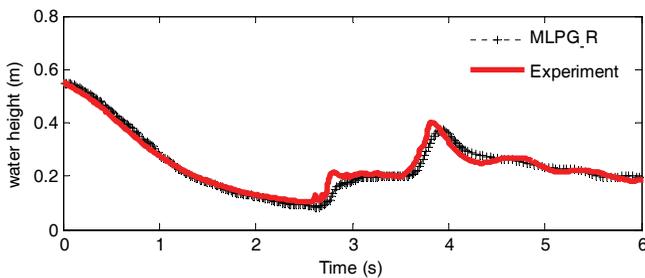


Figure 15: The time histories of the free surface elevation at H4

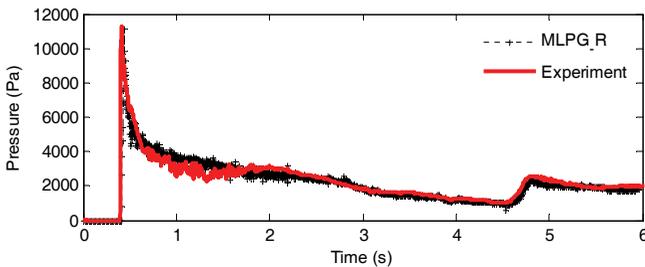


Figure 16: Pressure time history at P1 in Fig. 10

method is robust and potentially one of good options to model the violent waves and their interaction with structures. Nevertheless, the investigations also reveal that the further improvement on the method is still required in some areas, such as enhancing the accuracy of simulating post-breaking waves.

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

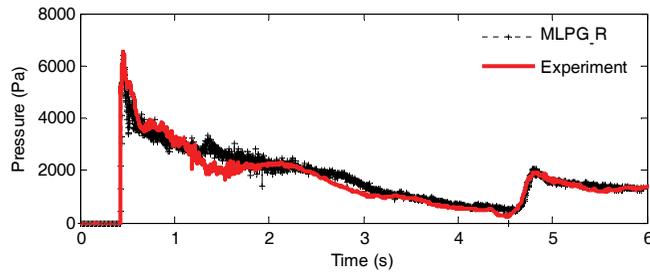


Figure 17: Pressure time history at P3 in Fig. 10

References

- 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.; 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.
- Avila. R.; Atluri. S.N.** (2009): Numerical Solution of Non-steady Flows, Around Surfaces in Spatially and Temporally Arbitrary Motions, by using the MLPG method, *CMES: Computer Modelling in Engineering & Sciences*, Vol. 54, No.1, pp. 15-64.
- 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.
- Bergamaschi, L; Martinez, A; Pini, G.** (2009): An Efficient Parallel MLPG Method for Poroelastic Models, *CMES: Computer Modelling in Engineering & Sciences*, Vol. 49, No.3, pp. 191-214.
- Bonmarin, P.** (1989): Geometric Properties of Deep-Water Breaking Waves,” *J. Fluid Mech.* 209, 405.

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

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.

Ingram, D.M.; Gao, F.; Causon, DM.; Mingham, C.G.; Troch, P. (2009). Numerical Investigations of Wave Overtopping at Coastal Structures. *Coastal Engineering*, Vol. 56(2), pp. 190-202.

Issa, R.; Violeau, D.; Lee, E.S.; Flament, H. (2010): 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.

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, KMT.; Fekken, G.; Veldman, AEP.; 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, BH.; Park, JC. (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. (2008): The MLPG mixed collocation method for material orientation and topology optimization of anisotropic solids and structures, *CMES: Computer Modeling Engineering & Sciences*, 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 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 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.

Lv, X.; Q.-P. Zou, & D. E. Reeve (2009): A Hybrid LS and VOF method for 3-D simulation of wave breaking and overtopping, *International Journal of Offshore and Polar Engineering*, Vol. 19, No. 4, December 2009, pp. 308-316.

Ma, Q.W. (2005a): Meshless Local Petrov-Galerkin 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 Modelling 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.; Yan, S. (2009): QALE-FEM for Numerical Modelling of Nonlinear Interaction between 3D Moored Floating Bodies and Steep Waves, *International Journal for Numerical Methods in Engineering*, Vol. 78, pp. 713-756.

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 Modelling in Engineering & Sciences*, Vol 23, No 2, pp 75-89.

Ma, Q.W. (2010): 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.

Ma, Q. W.; Zhou, J. T. (2009): MLPG_R Method for Numerical Simulation of 2D Breaking Waves, *CMES: Computer Modelling in Engineering & Sciences*, Vol. 43, No 3, pp 277-303.

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.

Qian, L.; Causon, D.M.; Mingham, C.G.; Ingram, D.M. (2006), A Free-Surface Capturing Method for Two Fluid Flows with Moving Bodies, *Proceedings of the Royal Society: A* Vol. 462 (2065), pp. 21-42.

Qian, L.; Causon, D.M.; Ingram, D.M.; Mingham, C.G. (2003), Cartesian Cut Cell Two-Fluid Solver for Hydraulic Flow Problems, *Journal of Hydraulic Engineering*, Vol. 129(9), pp.688-696.

Rapp, R.J.; Melville, W.K. (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, EYM. (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; Tan, CL; 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.

Sladek. J; Sladek. V; Wünsche. M; Zhang. Ch. (2009): Interface Crack Problems in Anisotropic Solids Analyzed by the MLPG, *CMES: Computer Modelling in Engineering & Sciences*, Vol. 54, No.2, pp. 223-252.

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 Engrn.*, 29, 317-346.

Yang, G.; Causon, D.M.; Ingram, D.M. (2000): Calculation of Compressible Flows About Complex Moving Geometries Using a 3D Cartesian Cut Cell Method. *International Journal of Numerical Methods in Fluids*, Vol. 33, pp. 1121-1151.

Yan, S.; Ma, Q.W. (2009): QALE-FEM for modelling 3D overturning waves, *International Journal for Numerical Methods in Fluids*, available online from Jun 22 2009, DOI: 10.1002/fld.2100.

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.

Zheng, J.; Long, S.; Xiong, Y.; Li, G. (2009): A Finite Volume Meshless Local Petrov-Galerkin Method for Topology Optimization Design of the Continuum Structures, *CMES: Computer Modelling in Engineering & Sciences*, Vol. 42, No.1, pp. 19-34.

Zhou, J.; Ma, Q.W.; Zhang, L; Yan, S. (2009): Numerical Investigation of Violent Wave Impact on Offshore Wind Energy Structures using MLPG_R Method, *Proceedings of ISOPE*, ISBN 978-188065353-1, Vol 3, pp. 503-509.