An improved immersed boundary method with new forcing point searching scheme for simulation of bodies in free surface flows

Bin Yan[a], Wei Bai[b,*], Ser Tong Quek[a]

[a]*Department of Civil and Environmental Engineering, National University of Singapore, Kent Ridge, Singapore 117576, Singapore*
[b]*School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, Chester Street, Manchester M1 5GD, UK*

Abstract

An improved immersed boundary method is proposed and applied to simulate fluid-structure interactions by combining a level set method for free water surface capturing. An efficient Navier-Stokes equation solver adopting the fractional step method at a staggered Cartesian grid system is used to solve the incompressible fluid motion. A new efficient algorithm to search forcing points near the immersed body boundary is developed. The searching schemes for forcing points located both inside and outside the solid phase with the linear interpolation schemes for the determination of velocities at forcing points are presented and compared via the case of dam break over obstacles. The accuracy and effectiveness of the proposed forcing point searching schemes are further demonstrated by the study of wave propagation over a submerged bar and more challenging cases of wedge with prescribed velocity or falling freely into the water. By the extensive comparison of present numerical results with other experimental and numerical data, it suggests that the present improved immersed boundary method with the new forcing point searching scheme has a better performance and is very promising due to its accuracy, efficiency and ease of implementation. Furthermore, the present numerical results show that the outside forcing scheme is superior over the inside forcing scheme.

Keywords: Immersed boundary method, Level set method, Forcing point searching scheme, Dam break, Free fall wedge

1. Introduction

Fluid-structure interaction (FSI) is a classical hydrodynamic problem and has a wide range of applications in many ocean and coastal engineering problems. Numerical simulations gain its popularities to handle the FSI problems. However, numerical simulation of fluid-structure interactions is extremely complicated, especially when involving moving objects with irregular boundaries and complex free surface evolutions. With the rapid advance in computing technology, more researchers and engineers have paid extensive attentions to the development of efficient numerical methods to study fluid-structure interactions. Traditionally, the

---

*Corresponding author
Email address:* w.bai@mmu.ac.uk (Wei Bai)

problems can be solved by the boundary-fitted method (Yan and Ma, 2007; Yang et al., 2008), which generates the curvilinear structured or unstructured grids conforming to the body boundary. However, grid regeneration is entailed with a heavy cost in computational time as well as manpower. The drawback of the method due to its inapplicability to the multi-grid acceleration solver is also obvious.

Over the last few decades, Immersed Boundary Method (IBM) becomes increasingly popular among the numerical methods to simulate bodies in fluid domain. It introduces a body force to the momentum equations at certain points in the domain, without the necessity of performing the mapping procedures, aiming to simulate the effect of the investigated body in the flow. As a result, generation of grids is greatly simplified. The immersed boundary method was firstly proposed by Peskin (1972), based on which Goldstein et al. (1993) and Saiki and Biringen (1996) introduced a feedback forcing mechanism to enforce the desired boundary condition at the immersed boundary. It can be implemented into an existing Navier-Stokes solver with relative ease due to its advantage of being formulated relatively independent of the spatial discretization. However, this technique may induce high-frequency spurious oscillations and restrict the computational time step, which makes the simulation of flow fields in complex domains very expensive.

To address the issue of too small time step, Mohd-Yusof (1997) proposed a discrete-time immersed boundary method combining with a B-spline spectral method, which allows the implementation of complex moving geometries in the pseudo spectral codes. Due to the expense of calculating the B-spline coefficients at each time step, it requires much memory so that the availability is restricted. Fadlun et al. (2000) applied the discrete-time forcing scheme on a staggered grid and compared with the feedback forcing scheme proposed by Goldstein et al. (1993) and Saiki and Biringen (1996). The comparison indicated that the discrete-time forcing scheme is more efficient than the feedback forcing scheme. In addition, Fadlun et al. (2000) imposed the forcing term inside the flow field while in Mohd-Yusof (1997) the momentum forcing was applied only on the body surface or inside the body. Fadlun et al. (2000) also tested the three interpolation procedures, stepwise geometry, volume fraction and linear interpolation. It was shown that linear interpolation can yield most accurate solution.

Kim et al. (2001) developed a new immersed boundary method by introducing a mass source/sink as well as a momentum forcing applied on the body surface or inside the body. Although Kim et al. (2001) adopted both the linear and bilinear interpolation schemes, no comparison was given to determine which scheme is better. Furthermore, the combination of the immersed boundary method and the free surface simulation was not tested in Kim et al. (2001). Based on the direct forcing scheme, Balaras (2004) performed large eddy simulations around complex boundaries on fixed Cartesian grids. In his method, the forcing term was added to the points in the flow field near the body boundary. Chiu et al. (2010) developed a differential-based interpolation scheme for the direct forcing term in the immersed boundary method and compared to the algebraical interpolation method. Although the differential-based interpolation can obtain high order accuracy, it requires iterations to get the velocity for the direct forcing term, which may demand large computer resource.

Before the work in Balaras (2004), Mohd-Yusof (1997), Fadlun et al. (2000) and Kim et al. (2001) only described little about the procedure of locating the forcing points. Balaras (2004) proposed a tedious scheme to determine the forcing points. First step is to tag all the velocity points in the vicinity of the immersed boundary with -1 (fluid phase) and $+1$ (solid phase) flags according to the normal vector. Second step is to examine all the points with -1 flag to find out if they have at list one neighbour with $+1$ flag. If the points with -1 flag satisfy the requirement, they are determined as forcing points. Recently, Mittal et al. (2008) described a highly versatile immersed boundary method (based on a discrete-forcing scheme) by the calculation of variables on 'ghost-cells' inside the body to satisfy the boundary conditions. In spite of a 'sharp' representation of the moving immersed boundary, Mittal et al. (2008) only applied the model to flows without the free surface. At the same time, Zhang et al. (2010) developed a new level set immersed boundary method to investigate the interaction between free surface flows and structures. In their model, an algorithm to locate exactly the forcing points inside the solid phase was proposed. However, the algorithm of searching forcing points via a bounding box with triangular grids is complicated to implement and it is not easy to locate forcing points around a sharp boundary. Similar to the approach of Balaras (2004), Zhang et al. (2010) still required additional steps to exclude the flagged solid points which do not have one or more neighbouring points in the fluid phase.

From the above discussion, one may note that the importance of searching algorithm to locate forcing points can be easily overlooked in the previous work, or the searching algorithm is quite complicated for implementation, which motivates this piece of work. In fact, an accurate and efficient searching scheme is essential for the success of the immersed boundary method. In the present paper, a simple and straightforward forcing point searching scheme is proposed, which exhibits the advantage of simplicity and ease of implementation with remaining desirable accuracy. For a node around the immersed body boundary to be identified as a forcing point, it may lie in either the solid phase or the fluid phase. However, there is no generally accepted standard in choosing forcing points inside or outside the solid phase so far. This work attempts to shed light on this issue by the comparison of forcing points both inside and outside the solid phase, and come out with a recommendation. The developed immersed boundary method is used to study the complicated multi-phase flows in conjunction with a level-set method for free surface capturing developed by Archer and Bai (2015), which is also rare in the previous studies. Four testing examples involving both fixed and moving bodies with irregular geometries are used to demonstrate the effectiveness of the newly developed forcing point searching schemes. Extensive comparisons are made to confirm the accuracy of the present numerical model.

3

2. Mathematical formulation

2.1. Governing equations

In the study of two dimensional incompressible viscous flows, the motion of the fluid is governed by the Navier-Stokes equations,

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = \frac{1}{\rho}\left(-\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}\right) + g_i + f_i, \tag{1}$$

and the continuity equation,

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{2}$$

where the Cartesian tensor notation is used, u_i is the fluid velocity, p is the pressure, x_i is the spatial coordinate, t is the time, g_i is the gravitational acceleration, f_i is the momentum forcing component used to enforce the desired boundary condition on an immersed boundary interface in the present study. ρ is the fluid density and τ_{ij} are the viscous stress components given by

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{3}$$

where μ is the fluid viscosity.

2.2. Free surface simulation

As the present study focuses on the investigation of flows with free surface undergoing topological changes, splitting and merging, the level set method is adopted to capture the air-water interface in the frame of two-phase flow model. In the level set method, a scalar level set function ϕ is defined throughout the domain to represent the location of grid cell relative to the water surface. The level set function is a signed distance function, which measures the shortest distance from the grid cell to the water surface (i.e. $|\nabla\phi| = 1$) and is positive in one fluid phase and negative in the other. The evolution of the level set function ϕ is governed by

$$\frac{\partial \phi}{\partial t} + u_i \frac{\partial \phi}{\partial x_i} = 0. \tag{4}$$

In the Navier-Stokes equations, both ρ and μ depend on the local fluid phase properties. If ρ and μ are discontinuous, the numerical solution of the Navier-Stokes equations may yield instabilities at the interface. In order to avoid the possible numerical instability caused by the sharp gradients of fluid properties, ρ and μ are smoothed over a small distance $\varepsilon = 2\Delta x$ across the interface by the use of a Heaviside function H, where Δx is the typical grid size. We calculate ρ and μ by

$$\rho(\phi) = \rho_{air} + H(\phi)(\rho_{water} + \rho_{air}), \tag{5}$$

$$\mu(\phi) = \mu_{air} + H(\phi)(\mu_{water} + \mu_{air}), \tag{6}$$

where the subscripts *air* and *water* denote values of the air and water respectively, and the Heaviside function

98 is defined by

$$H(\phi) = \begin{cases} 0 & \text{if } \phi < -\epsilon \\ \dfrac{1}{2}\left[1 + \dfrac{\phi}{\varepsilon} + \dfrac{sin\left(\pi\phi/\varepsilon\right)}{\pi}\right] & \text{if } |\phi| \leq \epsilon \\ 1 & \text{if } \phi > \epsilon \end{cases} . \tag{7}$$

99 3. Numerical method

100 *3.1. Navier-Stokes equation solver*

101   The Navier-Stokes equations are discretized using a finite difference method on a staggered grid, in which

102 the velocity components are defined at the centre of cell face in the x and y directions respectively, with all

103 the other variables, i.e. p, ϕ, ρ and μ defined at the grid cell centre. With a second-order Runge-Kutta Total

104 Variation Diminishing (RK-TVD) scheme adopted to discretize the temporal gradient, the Navier-Stokes

105 equations can be solved by using a fractional step method,

$$u_i^{**} = u_i^n + \Delta t \left(\frac{1}{\rho^n\left(\phi\right)}\left(\frac{\partial \tau_{ij}^n}{\partial x_j}\right) - u_j^n \frac{\partial u_i^n}{\partial x_j} + g_i + f_i\right), \tag{8}$$

$$\frac{\partial}{\partial x_i}\left(\frac{1}{\rho^n\left(\phi\right)}\frac{\partial p^n}{\partial x_i}\right) = \frac{1}{\Delta t}\left(\frac{\partial u_i^{**}}{\partial x_i}\right), \tag{9}$$

$$\hat{u}_i^{n+1} = u_i^{**} - \frac{\Delta t}{\rho^n\left(\phi\right)}\frac{\partial p^n}{\partial x_i}, \tag{10}$$

106 where u^{**} is the predicted velocity and the superscript n denotes values at the time step n. The computed

107 pressure field is used to predict the new velocity field $\hat{u}_i^{n+1}$ in Eq. 10. The same procedure is then repeated

108 based on the predicted velocity field, and another new velocity field $\hat{u}_i^{n+2}$ can be determined, by which the

109 corrected velocity field at the time step $n+1$ can be eventually calculated by averaging the velocity field at

110 the time step n and the second prediction,

$$u_i^{n+1} = \frac{1}{2}\left(u_i^n + \hat{u}_i^{n+2}\right). \tag{11}$$

111   As it is essential to avoid the introduction of numerical instabilities due to the sharp gradients at the

112 interface, the first-order upwinding scheme is adopted to discretise the convective term in Eq. 8. The spatial

113 gradients are discretized with a second-order central difference scheme.

114 *3.2. Free surface solver*

115   Accurate solution of the level set equation (Eq. 4) is crucial to capture the air-water interface accurately.

116 Here, the velocity gradients in Eq. 4 is discretised by a fifth-order HJ-WENO scheme (Jiang and Peng,

2000), with a third-order RK-TVD scheme for the temporal gradient. Since only the location of free surface is of interest, the level set function ϕ can only be solved in a narrow band around the interface. In the present study the narrow band occupies six cells at each side of the interface (Peng et al., 1999). As the evolution of the level set function in time may cause ϕ deviate from being a signed distance function (i.e. $|\nabla\phi| \neq 1$), the reinitialisation is required. Here, we reinitialize ϕ using an efficient fast marching technique at every time step, see Sethian (1996) for details.

4. Immersed boundary method

Fluid flow over a body can exert a force on the no-slip body surface and the body will, in turn, apply a force with the same magnitude but in the opposite direction on the local flow. As a result, the fluid flow can be brought to rest on the body surface (Goldstein et al., 1993). Determination of momentum force exerted by the body on the fluid at the immersed body boundary is the key issue and also the main difficulty in the development of an immersed boundary method.

4.1. Identification of forcing points

The first step in the immersed boundary method is to predict the forcing points on which the momentum force is applied on the fluid. In the present searching method, the boundary of the solid body is represented by a series of straight line segments defined in an anti-clockwise direction, such that the solid phase is always located at the left hand side of the line segments. The present algorithm is applied to each boundary line segment. For the purpose of demonstration, the line 1-2 shown in Figs. 1 and 2 is a boundary segment, and the shadowed areas in the figures represent the solid phase, where Figs. 1 and 2 show the forcing points located inside and outside the solid phase respectively. The angle between the line 1-2 and the x direction needs to be first calculated. According to the angle being located at each quadrant, there are four possible situations,

$$
\text{Quadrant} = \begin{cases} 1, & \text{when } x_1 > x_2 \text{ and } y_1 < y_2 \\ 2, & \text{when } x_1 > x_2 \text{ and } y_1 > y_2 \\ 3, & \text{when } x_1 < x_2 \text{ and } y_1 > y_2 \\ 4, & \text{when } x_1 < x_2 \text{ and } y_1 < y_2. \end{cases}
$$
(12)

In Figs. 1 and 2, only the situation when the line 1-2 is located at the first quadrant is discussed in detail, which can be extended easily to the other three quadrants. After identifying the quadrant that the line segment belongs to, we further consider two possibilities by comparing the line segment with the diagonal line in that quadrant, and discuss these two possibilities separately.

If the angle between the line segment and the x direction is smaller than that of the diagonal line in the corresponding quadrant (or equivalently the line segment is ahead of the diagonal line), the forcing points

6

(a) Line segment slope < diagonal line slope

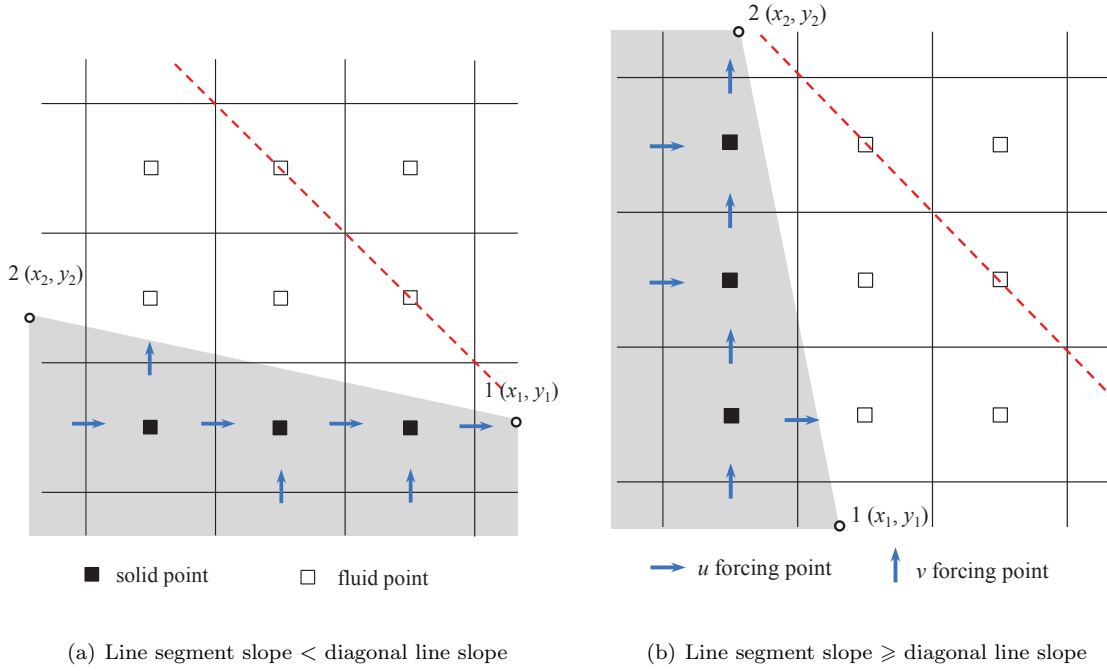(b) Line segment slope ⩾ diagonal line slope

Figure 1: Demonstration of searching procedure for forcing points located inside the solid phase. The red dash line is denoted as the diagonal line.

around the solid boundary are searched along the x direction. Otherwise, the searching along the y direction would easily lead to the missing of forcing points because the vertical distance between Point 1 and Point 2 (shown in Figs. 1(a) and 2(a)) is much smaller than the horizontal distance. In the searching process, an imaginary Lagrangian point travels from Point 1 along the line segment. When the Lagrangian point meets the first vertical grid line, the intersection between the line segment and the vertical grid line is recorded. Depending on the forcing points chosen to be inside or outside the solid phase, the nearest u velocity position in the corresponding phase is identified, and defined as a u forcing point. The Lagrangian point continues to travel by a half grid in the x direction, such that it locates on the same vertical line with the v velocity. Along this vertical line, the nearest v velocity position in the corresponding phase is recorded as a v forcing point. When the Lagrangian point eventually reaches Point 2, all required forcing point information can be gathered, which will be used in the interpolation. The rule discussed and corresponding results are demonstrated by the blue arrows in Figs. 1(a) and 2(a). Discussion on the forcing points inside or outside the solid boundary indicates that the difference between the two types of forcing point searching schemes is very little, which is physically just a cell apart between them. The main difference between these two searching schemes lies in the interpolation, which will be discussed later.

The searching procedure is also applicable to the case when the slope of the line segment is larger than 1 (or equivalently the line segment is behind the diagonal line), as shown in Figs. 1(b) and 2(b), but the searching direction should be changed to the y direction. The similar procedure can be implemented in the
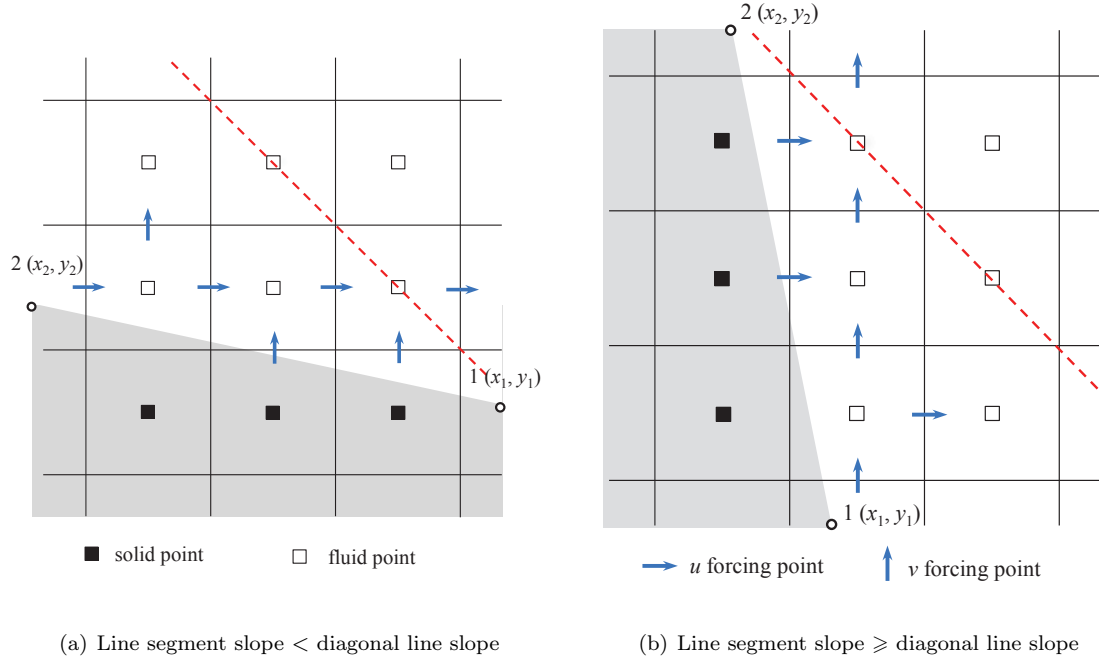
(a) Line segment slope $<$ diagonal line slope

(b) Line segment slope $\geqslant$ diagonal line slope

Figure 2: Demonstration of searching procedure for forcing points located outside the solid phase. The red dash line is denoted as the diagonal line.

other 3 quadrants. Especially, if the line segment is horizontal or vertical, which is identical to the horizontal or vertical grid line, the nearest u and v velocity positions in the corresponding phase can be simply chosen as the forcing points. In addition, for a very sharp corner, the forcing points obtained by searching along two different line segments might overlap. In this situation, the momentum forces obtained by the interpolation with respect to the two different line segments are averaged to provide a unique value at this forcing point, which is different from the method proposed by Mittal et al. (2008) who selected the forcing point closest to the boundary and applied the momentum forcing on it.

It should be noted that the numerical exercise indicates that no obvious additional computer time is required in the current new forcing point searching algorithm, as all the calculations are quite straightforward, without any complicated searching and sorting operations.

4.2. Determination of momentum forcing component

It is noted that the forcing term f_i in Eq. 8 needs to be determined prior to the computation of velocities u^{**}. This forcing term is prescribed at each time step to satisfy the desired boundary velocity u_b on the boundary surface. To achieve the expression of the forcing term, the discretized momentum equation can be reformulated as

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = RHS_i^n + f_i, \tag{13}$$

8

where RHS includes all the convective, viscous, pressure gradient and body force terms in the governing equations. When the grid line coincides with the immersed solid boundary, the forcing term f_i must yield $u_i^{n+1} = u_b$ on the immersed solid boundary by having the formulation as

$$f_i = \frac{u_b - u_i^n}{\Delta t} - RHS_i^n. \tag{14}$$

This forcing term is direct in the sense that the desired boundary condition can be satisfied at every time step but only holds when the immersed solid boundary coincides with the grid line. However, in the general situations the Eulerian grid does not necessarily conform to the immersed boundary geometry, where the prescribed momentum forcing term acts only on the points adjacent to the immersed boundary. Therefore, f_i needs to be computed at the forcing points that are close to but not exactly located on the immersed boundary. The velocity at the forcing point, u_f, has to be constructed using the information from the boundary condition and surrounding field. With the velocity at the forcing point, the forcing term at the forcing point can be expressed as

$$f_i = \frac{u_f - u_i^n}{\Delta t} - RHS_i^n. \tag{15}$$

4.3. Interpolation technique

If the forcing point coincides with the immersed boundary, u_f should be equal to the velocity of the moving body, u_b, and it is zero when the body is fixed. Otherwise, the interpolation scheme ought to be used to obtain u_f. In Kim et al. (2001), the linear interpolation was implemented to calculate the enforced velocity at the forcing point. Zhang et al. (2010) proposed a method of second-order accuracy that requires a point on the immersed boundary which possesses a normal passing through the forcing point to be identified for every forcing point.

We continue to consider the velocity u in the x direction as an example to demonstrate the implementation of the linear interpolation. The different situations are shown in Fig. 3, where B and F denote the vertical coordinates of boundary and forcing point respectively. In addition, A and C in the figure are the positions of u nodes in the water phase adjacent to the solid boundary and next to A, and the vertical distances between various nodes are also indicated in the figure. We first look at the linear interpolation scheme when the forcing point is inside the solid phase, as seen in Figs. 3(a) and 3(b).

We can notice that when A and B are close, the linear interpolation may lead to a velocity u_f at the forcing point with large error, which may cause the numerical instability. Therefore, we consider two separate situations by comparing the distances between B and A (defined as h) and between B and F (defined as h_A). If the distance between A and B is relatively small in Fig. 3(a), i.e. $h \leqslant h_A$, we can calculate u_f directly by

$$u_f = -\frac{h}{h_A} \cdot u_A + \frac{h + h_A}{h_A} \cdot u_b. \tag{16}$$

9
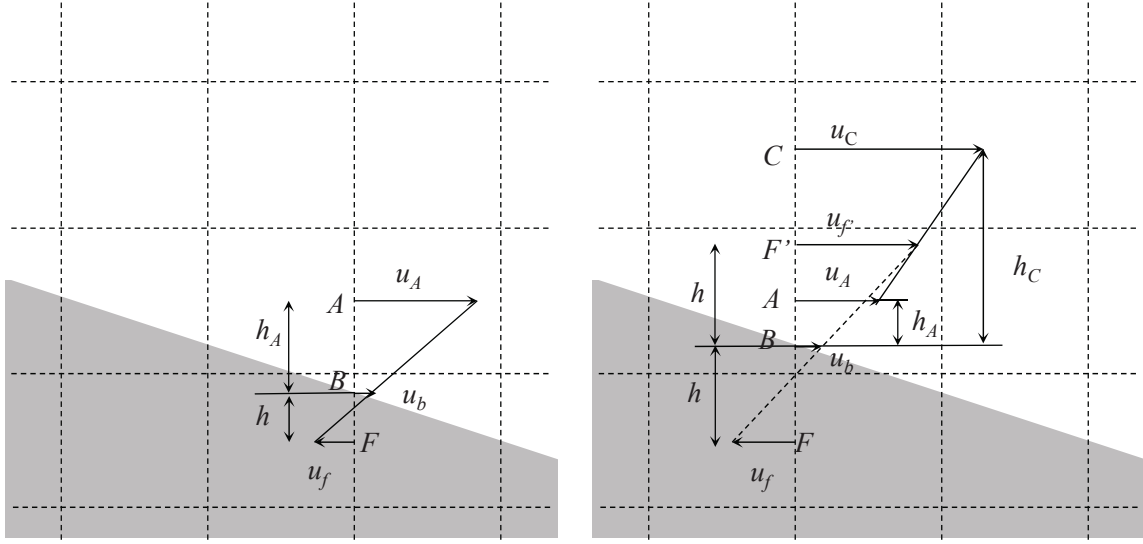
(a) Inside solid phase when $h \leqslant h_A$

(b) Inside solid phase when $h > h_A$
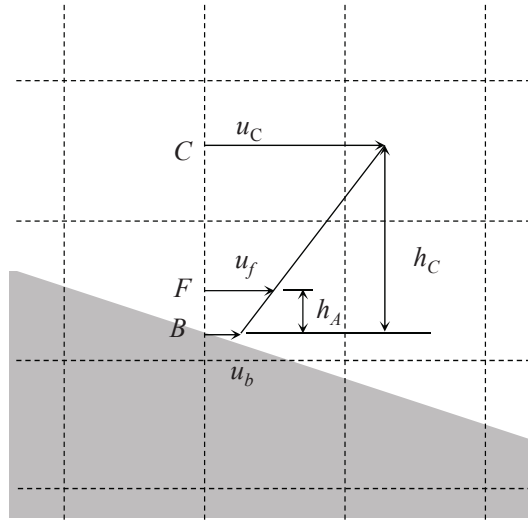
(c) Outside solid phase

Figure 3: Sketch of linear interpolation scheme for u velocity at forcing points inside and outside the solid phase. Here, F' is a virtual mirror point of F about the solid boundary B in the vertical direction.

Otherwise, the following linear interpolation is used for a relatively large distance between A and B when $h > h_A$ in Fig. 3(b),

$$u_f = -\frac{(h_C - h)\,u_A + (h - h_A)\,u_C}{h_C - h_A} + 2u_b. \tag{17}$$

It should be noted that in Eq. 17, the velocity u_C at C and the distance h_C between B and C are also adopted in the formulation to minimize the possible error in the prediction of u_f. To achieve this, a virtual

10

mirror point $F´$ of the point F about the solid boundary B in the vertical direction is required, on which the velocity is interpolated by using Points A and C.

However, if the forcing point is outside the solid phase, the issue discussed above does not exist, as the distance between B and F is always smaller than that between B and C, as shown in Fig. 3(c). Therefore, the linear interpolation can be simply expressed as

$$u_f = \frac{h}{h_C}\left(u_C - u_b\right) + u_b. \tag{18}$$

It should be noted that when the forcing point is inside the solid phase, the term "interpolation" actually means "extrapolation". It is known that the extrapolation scheme may be less accurate than the interpolation scheme, although an improvement has been proposed for the forcing points inside the solid phase in Eqs. 16 and 17.

5. Numerical results

5.1. Comparison of forcing point schemes via dam break

The dam break test presents an extreme challenge, as the flow experiences strong overturning, splitting and merging. This test case becomes even more complicated due to the presence of a fixed body in the domain, which may be very suitable to assess the strengths of the present immersed boundary method. Here we choose a circular cylinder that consists of many line segments of different slopes, and in this case, we focus on testing the accuracy of forcing point allocation schemes. The sketch of dam break over a circular cylinder is shown in Fig. 4, with a computational domain measured as $2.4m \times 2.4m$. The initial volume of the dam is $0.6m \times 1.2m$. In the numerical simulations throughout the study, a non-uniform grid is adopted to achieve a better performance. In this case, 150×150 cells are used to discretize the domain, and the mesh near the circular cylinder is much finer with the size of $0.01m \times 0.01m$.

To begin with, the comparison of forcing points inside and outside the solid phase can help to determine which forcing point searching scheme can achieve a better efficiency. Fig. 5 gives the distribution of forcing points for the u and v velocities around the circular cylinder. In this case, 12 line segments are employed to approximate the boundary of the circular cylinder. As can be seen in the figure, the forcing points are located adjacent to the cylinder surface, and they are at the correct side of the boundary by not crossing the boundary when different searching schemes are considered. Furthermore, we can observe that the distance between forcing points inside and outside the solid phase is just one grid.

Fig. 6 shows the snapshots of the dam break at several time instants obtained by the two different searching schemes, from which the whole process of the dam break can be observed. At the beginning, the dam breaks and flows towards the circular cylinder. After hitting the cylinder, the flow separates: part of the water volume jumps over the cylinder; while the other flows underneath the cylinder and impacts on the right wall, as shown at $T = 1.0s$. The water column on the right wall overturns and flows back to contact with the

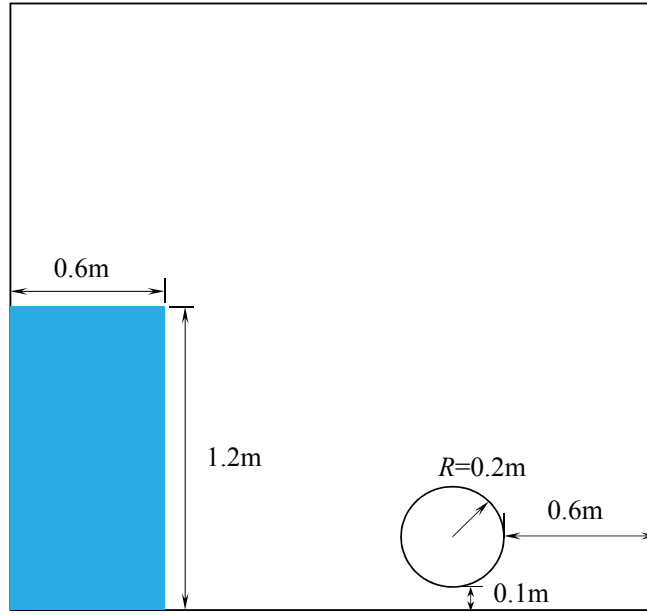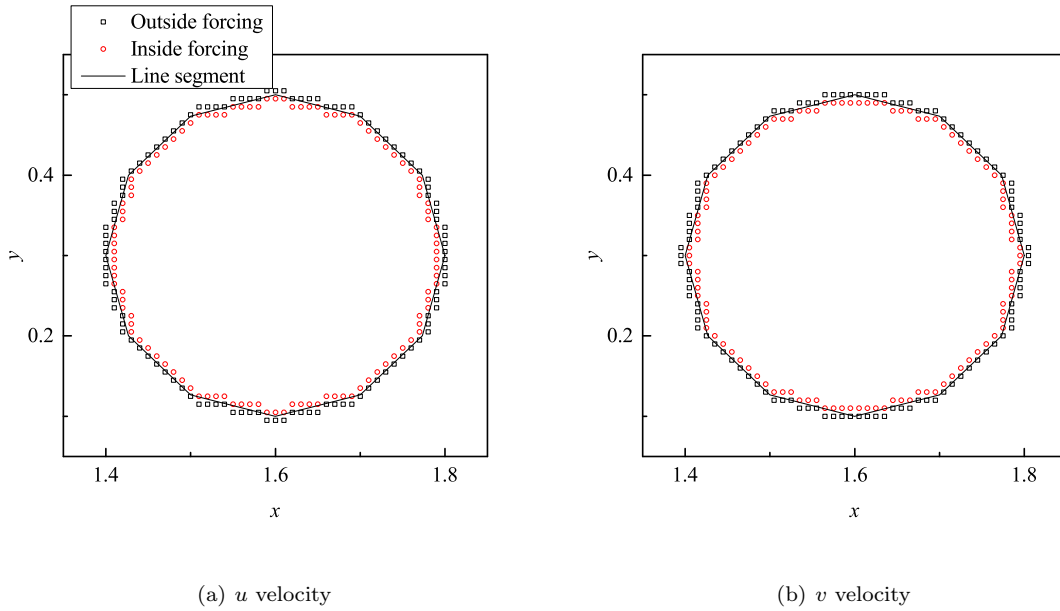Figure 4: Sketch of dam break over a circular cylinder.

(a) u velocity

(b) v velocity

Figure 5: Position of forcing point inside and outside the solid phase around the circular cylinder.

cylinder again. As a result, the opposite water jump develops at $T = 2.0s$. After $T = 2.0s$, the water sloshes between the left and right side walls, and eventually calms down. Generally, the two searching schemes for forcing points inside and outside the solid phase can provide similar results. However, after careful

observation we can notice that the inside forcing scheme can cause much more unrealistic water flow into the body volume, especially after $T = 2.0s$, which indicates that larger error occurs. In principle, the water should not flow into the cylinder volume if the forcing term in the governing equations is predicted without any error. However, in the immersed boundary method different forcing point searching and interpolation schemes can definitely affect the unavoidable numerical error produced during simulations.

To further quantize the water volume in the circular cylinder (which is the numerical error) with the use of different forcing point searching schemes, Fig. 7 shows the ratio of water volume in the cylinder to the cylinder volume against the time. As can be seen in the figure, the difference of water volume between the two different searching schemes becomes obvious from $T = 1.5s$. The water volume caused by the outside forcing scheme reduces after $T = 2.5s$, and it eventually tends to a steady value of 2%. However, the error in the inside forcing scheme keeps increasing with time, and it may reach over 10%, which is significantly larger than that in the outside forcing scheme. The main reason lies in that the inside forcing scheme actually adopts an extrapolation to predict the velocities at forcing points, which can easily cause large errors, compared to the interpolation in the outside forcing scheme. Therefore, the numerical results suggest that the outside forcing scheme performs better in this case.

5.2. Dam break over a rectangular bar

To further demonstrate the accuracy of the proposed immersed boundary method, another classical dam break over a cuboid is presented in this section. This case was investigated experimentally by Koshizuka et al. (1995). Fig. 8 provides the schematic view of the case. The dimension of the tank is the same as that in the experiment, and a rectangular obstacle is located in the middle of the tank, with the dimension $h \times 2h$ where $h = 24mm$ is adopted in the numerical simulation to be consistent to the experiment. For more details, please refer to Koshizuka et al. (1995). In the numerical simulation, the obstacle is represented by 3 vertical or horizontal line segments. The grid size of $0.002m \times 0.002m$ is adopted around the obstacle to ensure the accurate approximation of the geometry.

The ratio of water volume in the cuboid volume against time is tracked for different forcing point searching schemes, as shown in Fig. 9. It can be seen from the figure that the outside forcing scheme shows less volume of water entering the solid phase, indicating its better performance in controlling the numerical error compared to the inside forcing scheme, which is consistent to the conclusion drawn in the last section. Fig. 10 shows the process of dam break when it hits the cuboid, where the comparison of free surface profile at several time instants between the present and others' results is provided. The left column in the figure shows the experimental results captured by Koshizuka et al. (1995). The right column presents the numerical results by the present numerical model. At $T = 0.1s$, the water column collapses and impacts the obstacle, and then runs up to a certain level due to block effect of the obstacle at $T = 0.2s$. Following the further development, the jet occurs and hits the right wall. Finally, the water falls off the right wall, and it calms down gradually, as shown in the last two instants.
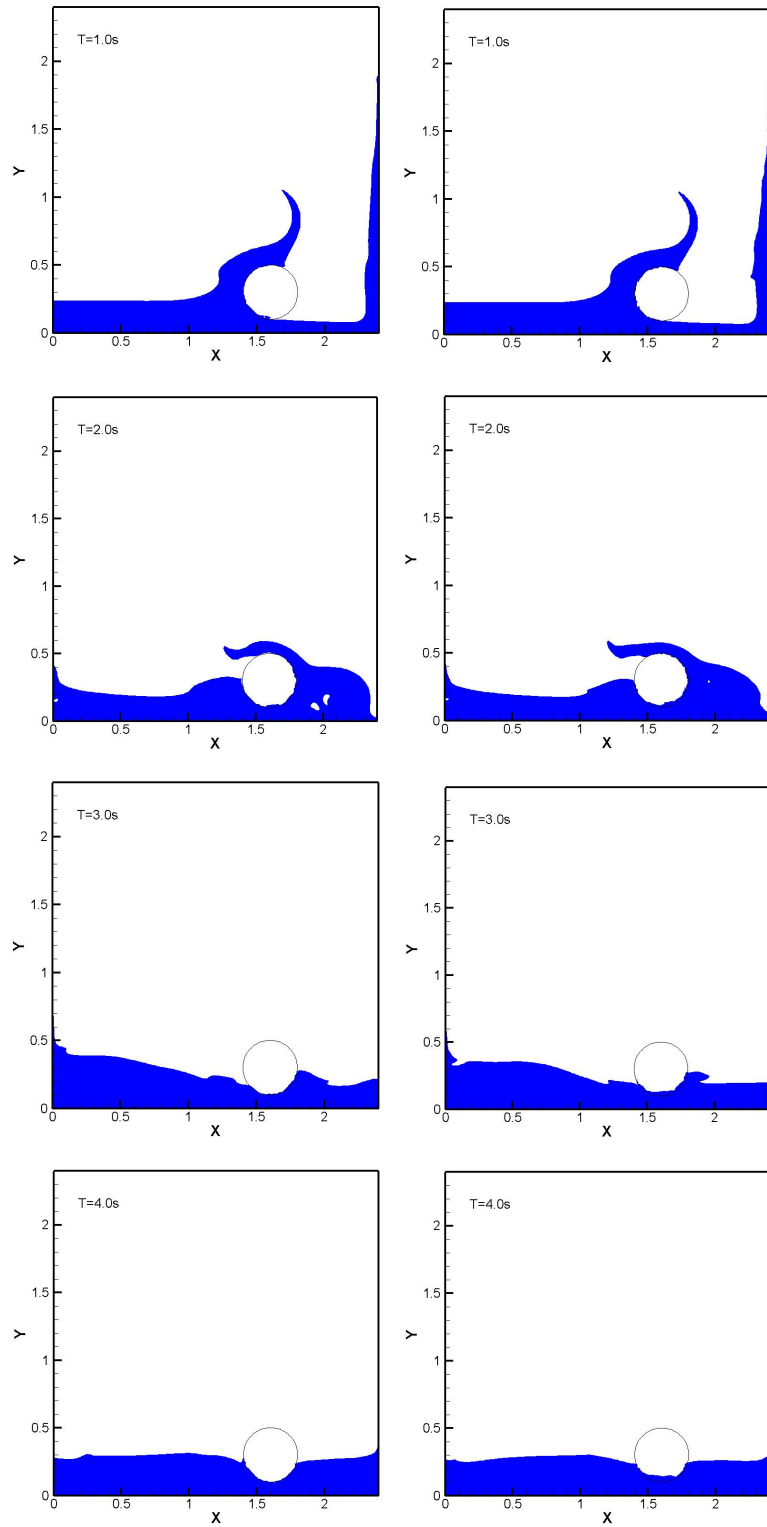
13

Figure 6: Snapshot of the dam break at several time instants: left column for the outside forcing scheme and right column for the inside forcing scheme.
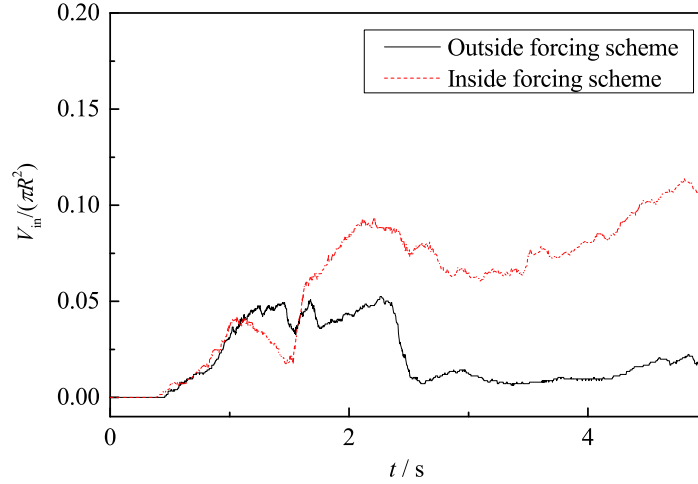
Figure 7: Ratio of water volume V_{in} in the cylinder to the cylinder volume obtained by two searching schemes.
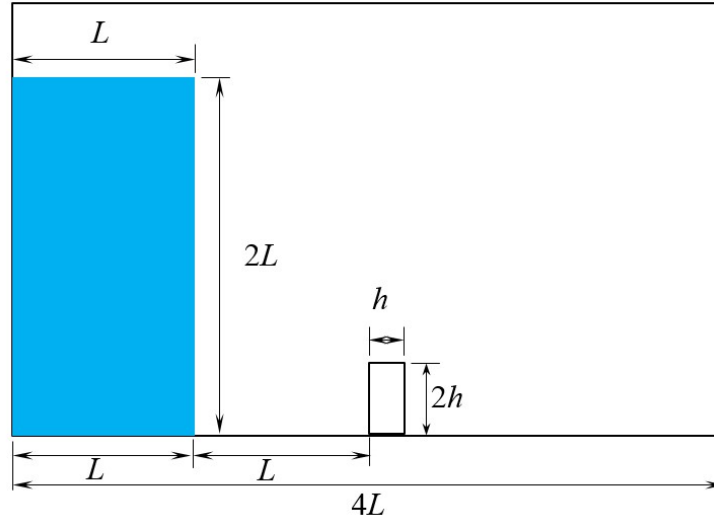

Figure 8: Sketch of dam break over a rectangular obstacle.

Generally speaking, good agreement is obtained between the present numerical results and the experimental measurement. Snapshot of the present numerical results at $t = 1.0s$ differs slightly with that of the experiment where less air is entrapped in the water. On the other hand, the numerical results by PFEM (Particle Finite Element Method) in Larese et al. (2008) is shown in the middle column of the figure for the purpose of comparison. It is obvious that the present numerical results agree better with the experiment than that in Larese et al. (2008), especially when $t = 0.4s$ and $t = 0.5s$. For example, the snapshot at $t = 0.5s$ from Larese et al. (2008) appears to have too many zigzags and less air pocket.

15
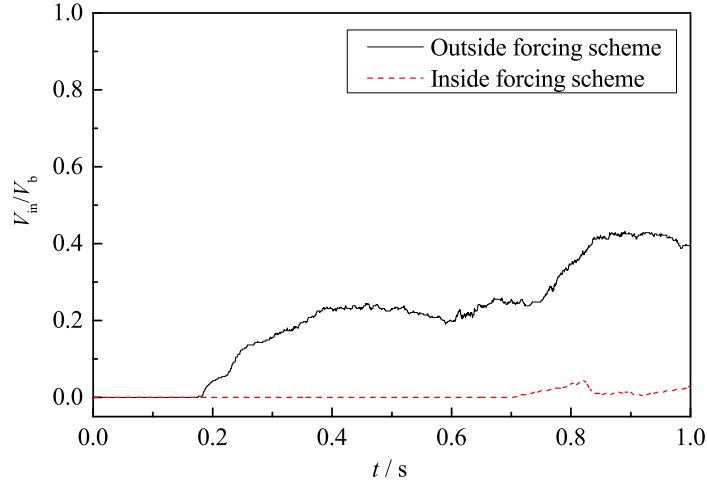
Figure 9: Ratio of water volume V_{in} in the cuboid to the cuboid volume V_b in different forcing search schemes.

5.3. Wave propagation over a submerged bar

In this section, a progressive wave travelling over a submerged bar is tested and compared with the experimental data. This case is a classical benchmark for the numerical method, which has been widely adopted by many researchers, such as Huang and Dong (1999), Lin and Li (2002) and Beji and Battjes (1994). In the present study, the model setup is the same as the physical experiment of Beji and Battjes (1994), as shown in Fig. 11. At the inlet boundary, the velocity according to the linear wave theory is specified to generate a wave with the period of $2s$ and amplitude of $0.01m$. The non-uniform mesh is also adopted in the simulation, where much finer cells are distributed around the submerged bar, so as to capture the accurate body shape. In addition, the mesh is also finer around the free water surface, where 8 cells are adopted in the vertical direction to cover one wave height.

The performance of different forcing point searching schemes is compared to further validate the effectiveness of the outside forcing scheme, as shown in Fig. 12. As the wave elevations at $x = 14.5m$ and $x = 15.7m$ experience strong nonlinearity, they can be more convincing to assess the accuracy of various schemes than that at other stations. From the figure, we can see that the result by the outside forcing scheme shows slightly better agreement with the experimental data, although these two searching schemes can provide almost identical results, as this case is less challenging compared to the case of dam break past an obstacle.

It should be mentioned that the numerical results shown above are obtained at a mesh denoted as Mesh_2 where the horizontal mesh size $\Delta x = 0.04m$ is adopted around the body. To test the mesh convergence of the present numerical model with the linear interpolation and outside forcing scheme, two other meshes are used: a coarser mesh with the horizontal mesh size $\Delta x = 0.08m$ denoted as Mesh_1 and a finer mesh with the horizontal mesh size $\Delta x = 0.02m$ denoted as Mesh_3, and the vertical mesh size is constant at $\Delta y = 0.0025m$ for all the three meshes to capture the relatively small wave amplitude. The time history of wave elevations at $x = 14.5m$ and $x = 15.7m$ is again shown in Fig. 13 for the three different meshes. In the simulation at
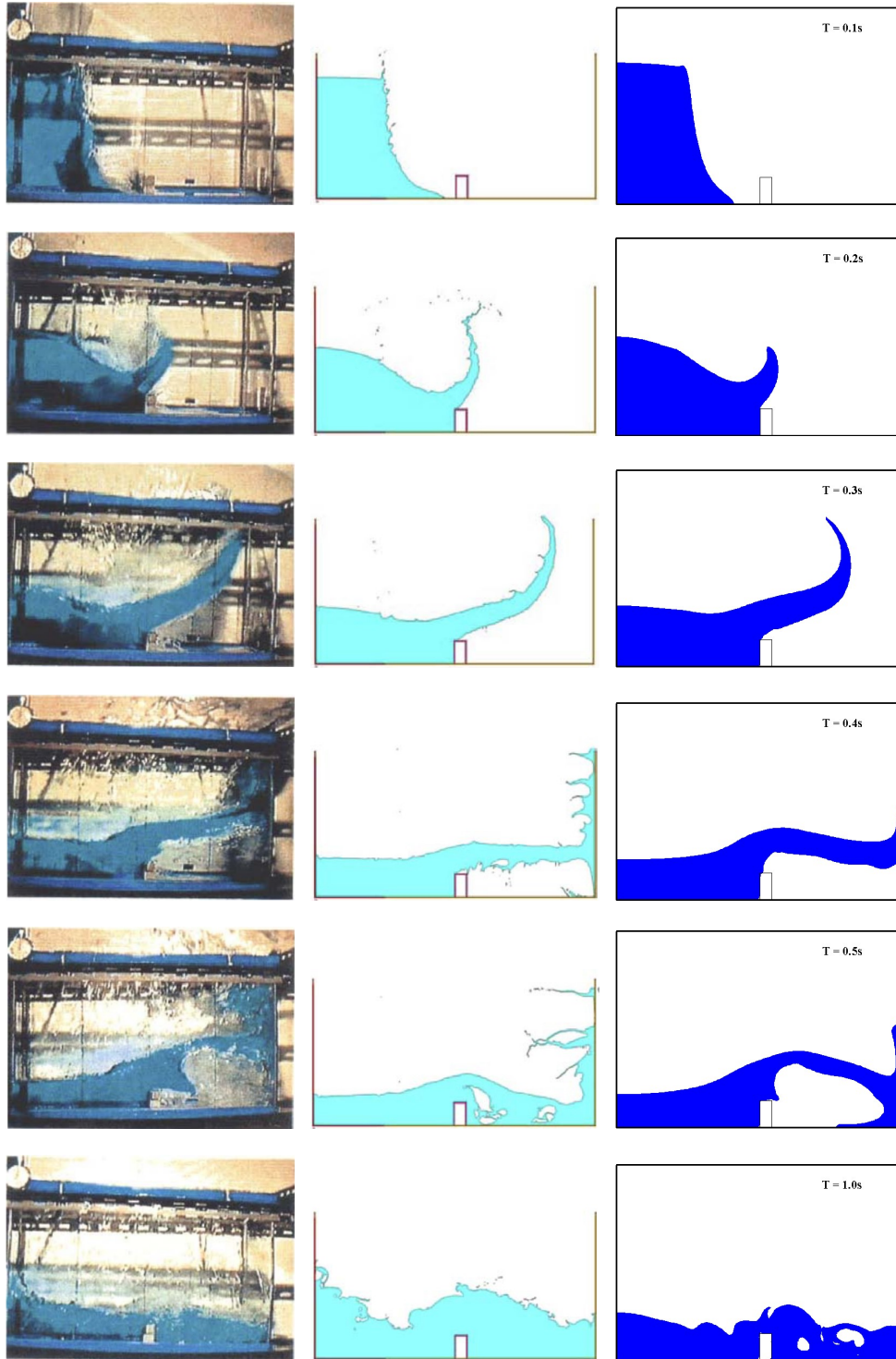
16

Figure 10: Comparison of free surface profile at different time instants for the dam break over an obstacle. Left column: experimental results in Koshizuka et al. (1995); middle column: numerical results by PFEM in Larese et al. (2008); right column: present numerical results.
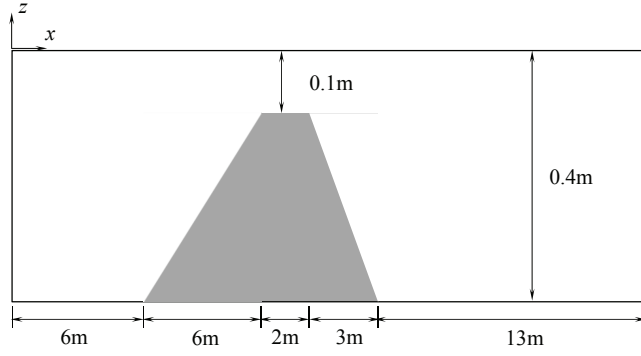
Figure 11: Sketch of computational domain for wave propagation over a submerged bar.
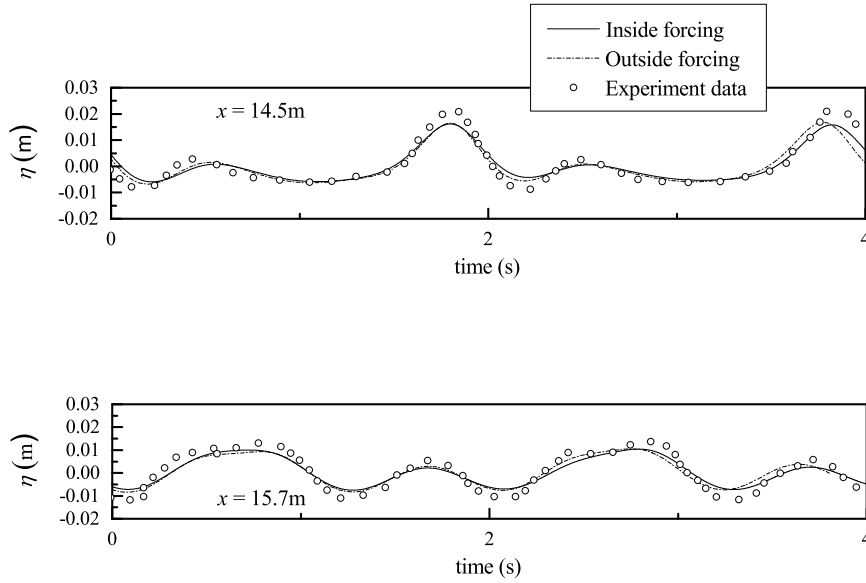

Figure 12: Time history of wave elevations at two stations obtained by different forcing point searching schemes.

Mesh_1, the time series of wave elevation obviously cannot agree with the experimental data, whereas the results at Mesh_2 and Mesh_3 are very close and the fine meshes tend to provide better agreement with the experimental data. It indicates that the result at Mesh_2 is convergent with respect to computational mesh, and the convergence rate is fast in the proposed numerical model.

Fig. 14 shows the final comparison of wave elevations at all the six stations. At the first two stations, the wave elevation is regular and sinusoidal, due to the fact that the submerged bar is far from the two stations and possesses little impact on the wave elevation at these two stations. The free surface at the third and fourth stations shows the nonlinear phenomenon, i.e. sharper wave crest and flatter trough. When the wave travels to the topside of the submerged bar, the reduction in water depth leads to larger wave heights. The
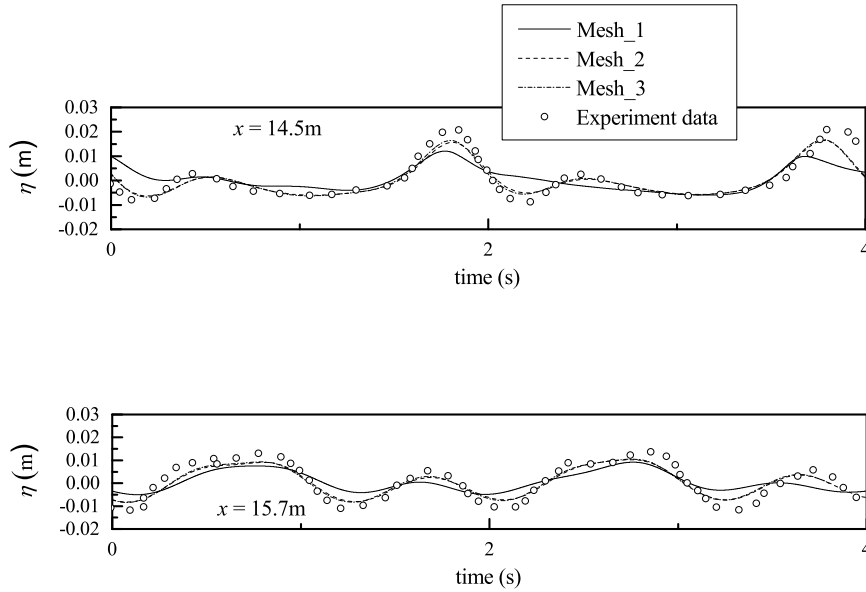
18

Figure 13: Time history of wave elevations with three different meshes to test the mesh convergence.

last two stations lie in the lee side of the submerged bar, where the wave elevation becomes very complicated because of the higher order wave harmonics induced. Overall, the numerical results agree well with the experimental data.

From the close comparison, we can notice that the present numerical results are better than that in Huang and Dong (1999) using the MAC method, although little discrepancy at the last two stations can be observed for both of the numerical results. At the same time, the same problem has been investigated in Shen and Chan (2008), where the direct forcing method based on Mohd-Yusof (1997) was adopted to impose the forcing term on the points nearest to the boundary. In their work, the numerical results under a finer mesh with the grid size of $0.02m \times 0.002m$ were compared with the experimental data, which seem to provide a similar accuracy compared to the present numerical results under Mesh 2. However, the present Mesh 2 is coarser than that in Shen and Chan (2008); this reflects the importance and effectiveness of the proposed forcing point searching scheme.

5.4. Water entry of a wedge with prescribed velocity

To further validate the present numerical model in terms of the convergence and accuracy, impact of a rigid V-shaped wedge moving with a constant downward speed V into the water is investigated in this section. The impact loading and hydrodynamic pressure on the wedge with different dead-rise angles β and various downward speeds V are compared with the analytical solution and other numerical results in literature. Fig. 15 shows the schematic diagram of the problem, in which the water penetration of wedge $h(t)$ and the resulting wetted length on the horizontal projection $r(t)$ are defined. The typical Reynolds number

19
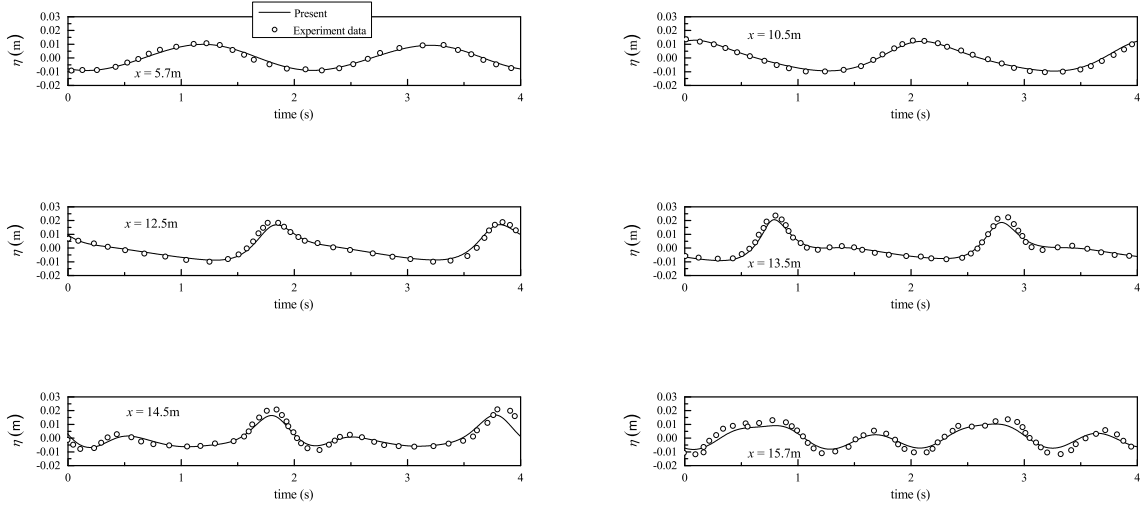
Figure 14: Comparison of time history of wave elevations at six stations with experimental data.

is $Re = VD/\nu = 3000$, where D is the maximum wetted surface length and ν is the kinematic viscosity. The Mach number is $Ma = 0.02$ here, which confirms that the viscosity and compressibility are considerably low for this case.
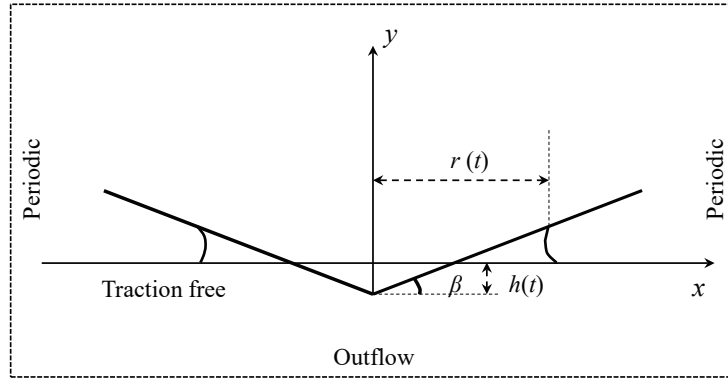

Figure 15: Sketch of water entry of a rigid wedge with prescribed velocity.

Firstly, the mesh convergence test is carried out on three different grids: the coarse grid ($\triangle x \times \triangle y = 0.01m \times 0.01m$), the median grid ($\triangle x \times \triangle y = 0.005m \times 0.005m$) and the fine grid ($\triangle x \times \triangle y = 0.0025m \times 0.0025m$). Fig. 16 shows the hydrodynamic load versus the horizontal projection of the wetted semi-length $r(t)$ on these three different grids for the wedge with the dead-rise angle $\beta = 15°$ and penetration velocity $V = 10m/s$. It is observed that the results on the median grid almost coincide with those on the fine grid, which indicates that the median grid can provide the convergent numerical results and is used in the following calculations for this problem. It should be noted that the convergent grid used here is much coarser than that used in De Rosis et al. (2014) adopting the Lattice Boltzmann model, and the comparison of the

20

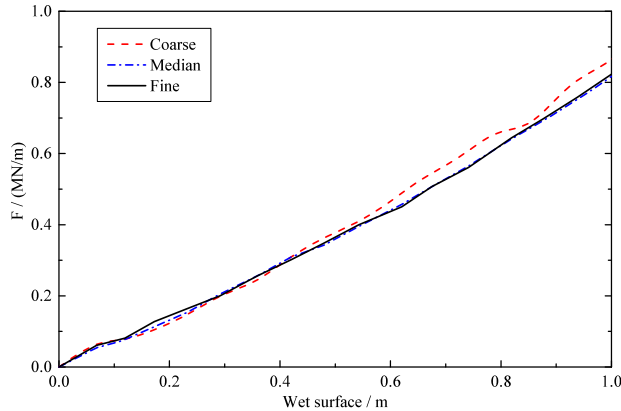impact load between the two models will be presented below.

Figure 16: Hydrodynamic load versus the horizontal projection of the wetted semi-length on three different grids for the case with the dead-rise angle $\beta = 15°$ and velocity $V = 10$m/s.

For the demonstration of accuracy of the numerical model, the present results of hydrodynamic load are compared with the analytical solutions presented in Von Karman (1929) and Wagner (1932), and the numerical results in De Rosis et al. (2014). Fig. 17 shows the comparisons for different dead-rise angles and penetration velocities. From the figure, it can be seen that compared to the analytical solutions the present numerical results are closer to the numerical results in De Rosis et al. (2014). However, with the small dead-rise angle at $\beta = 15°$ the present results are slightly over-estimated compared to the numerical results in De Rosis et al. (2014) at the last phase of the simulation. With the larger dead-rise angle at $\beta = 25°$, the comparison is favorable for the small speed, but the present results seem to be under-estimated. Generally, the present results fall into the range of the two analytical solutions, and agree reasonably well with other numerical results.

As shown in Fig. 18, the pressure coefficient $C_p = p/(0.5\rho_{water}V^2)$ for the wedge with $\beta = 30°$ and $\beta = 45°$ respectively is compared with the analytical solution in Mei et al. (1999), the similarity solution in Dobrovol'Skaya (1969), and the numerical results obtained using the Lattice Boltzmann model in Zarghami et al. (2014). It can be found that when $\beta = 30°$, the present numerical results agree better with the analytical solution in Mei et al. (1999) than the similarity solution and the other numerical results, especially in terms of the peak value. For $\beta = 45°$, the present numerical results are also closer to the analytical solution. The good comparison shown in the figure again confirms the accuracy of the present numerical model.

5.5. Free fall of wedge

The free falling of a wedge is another challenging test case to every numerical model, as this problem involves a complicated free surface interface induced by a moving body, and therefore, an air-water-solid three phase model should be considered. In the present study, the water entry of a free fall wedge is simulated by the proposed numerical model, and compared with the experimental and numerical work conducted by

21
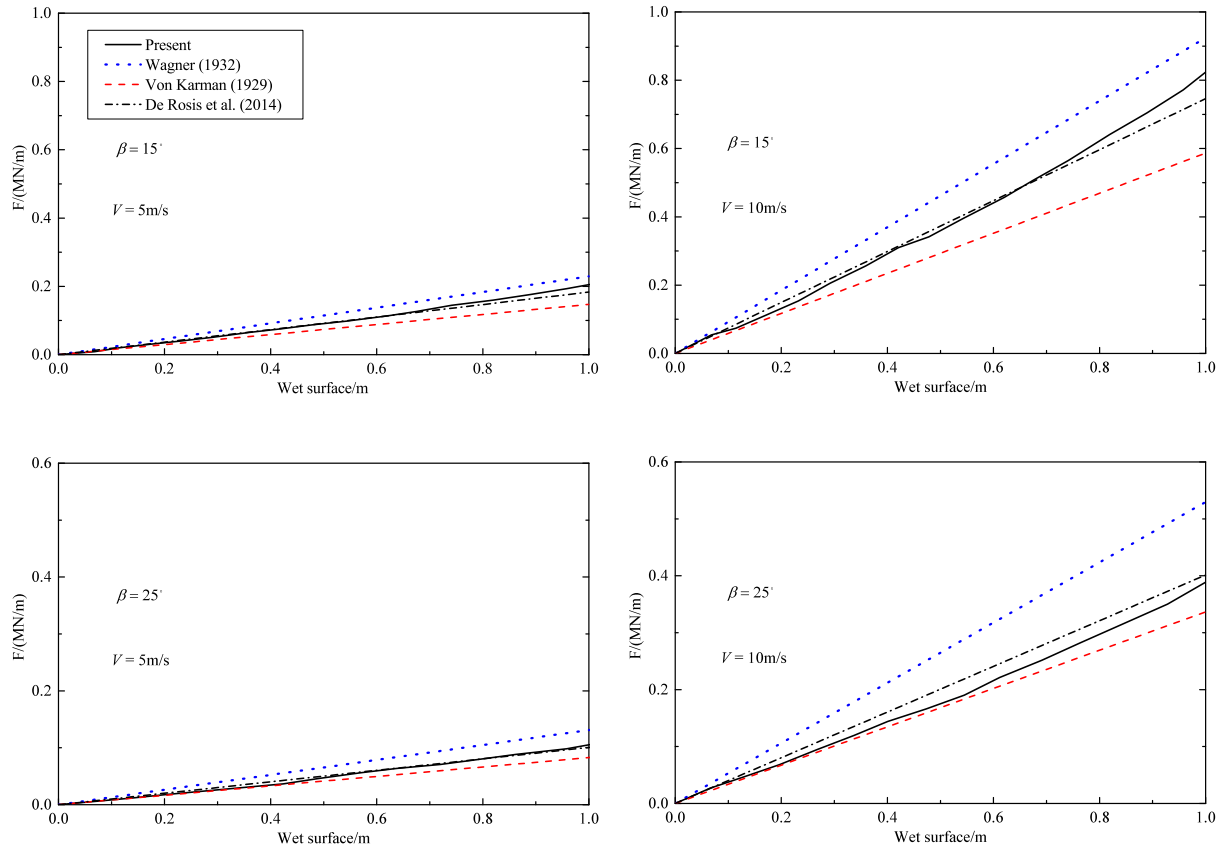
Figure 17: Comparison of hydrodynamic load with the analytical solutions and other numerical results for different dead-rise angles: $\beta = 15°$, $\beta = 25°$; and different speeds: $V = 5m/s$, $V = 10m/s$.
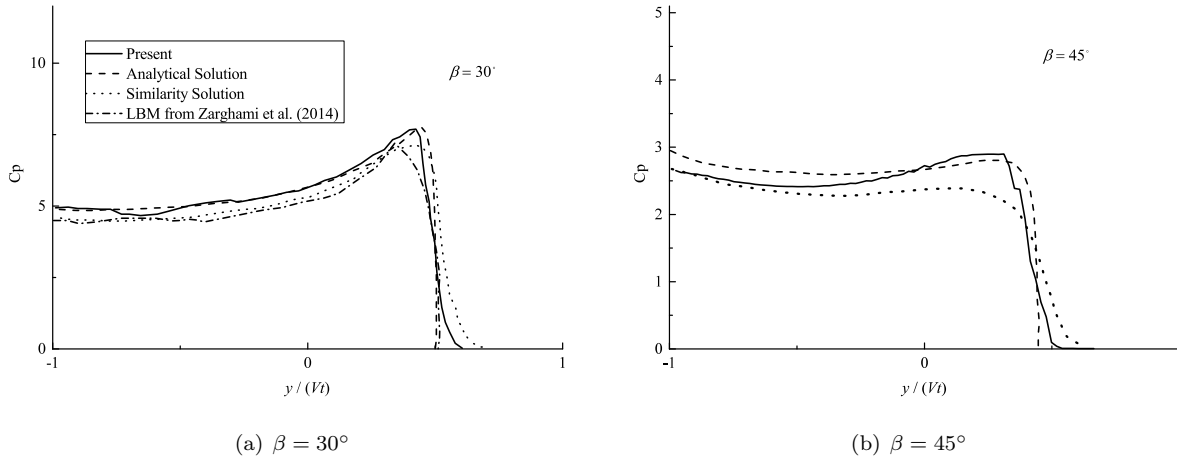

(a) $\beta = 30°$

(b) $\beta = 45°$

Figure 18: Slamming pressure coefficient C_p along the wetted semi-wedge with the speed $V = 10m/s$.

Zhao et al. (1997) and Zhang et al. (2010). The experiment setup in Zhao et al. (1997) is shown in Fig. 19, in which the breadth of the free fall V-shaped wedge is $500mm$ with a 30°dead-rise angle. The weight of the drop rig is $141kg$ with a ballast weight of $100kg$. The wedge can fall freely in the vertical direction only, and five pressure gauges are installed with the configurations shown in the figure to measure the pressure distribution at various time instants (see more detailed description of the experiment in Zhao et al., 1997). The numerical simulations are carried out in a numerical tank of $2m \times 1m$ in width and depth, with the specified water and air dynamic viscosity to be $1 \times 10^{-3}kg/m/s$ and $1.8 \times 10^{-5}kg/m/s$, respectively. The density of water and air is set as $1000kg/m^3$ and $1kg/m^3$, and the initial velocity of the wedge is prescribed as $V = -6.15m/s$.
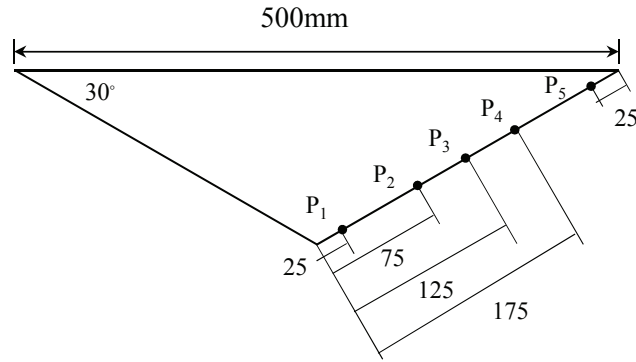
Figure 19: Wedge section adopted in the experiment and numerical simulation and arrangement of pressure gauges $P1 - P5$.

Firstly, the grid convergence tests are presented in Fig. 20. The grid sizes varying from $\Delta x = \Delta y = 0.0025m$ to $\Delta x = \Delta y = 0.01m$ are adopted and the results are compared with the experimental data in Zhao et al. (1997). Here, we define the coarse mesh $\Delta x = \Delta y = 0.01m$ as Mesh_1, the intermediate mesh $\Delta x = \Delta y = 0.005m$ as Mesh_2 and the fine mesh $\Delta x = \Delta y = 0.0025m$ as Mesh_3. From the water surface profile shown in Fig. 20(a), it can be observed that the better water jet can be captured at the fine mesh, where more detailed information can be presented, and the result at Mesh_2 approaches closely to that at Mesh_3. Furthermore, the free fall velocity in Fig. 20(b) and the slamming force in Fig. 20(c) are very close at these three grids, except that the slamming force at the fine mesh is much smoother than that at the other two coarser grids. It indicates that the global hydrodynamics, such as the free fall velocity and force, converge very fast with regards to computational mesh, whereas the detailed free surface profile is more sensitive to the density of computational grid.

It is also seen from Fig. 20 that the body motion and the hydrodynamic force on the body exhibit two main stages during the whole process. At the first stage when $0 < t < 0.016s$, the impact of the free fall wedge leads to an increase in the slamming force, until the force reaches its maximum when the wedge is fully submerged in the water. After that, when the wedge further slows down during $0.016s < t < 0.025s$,
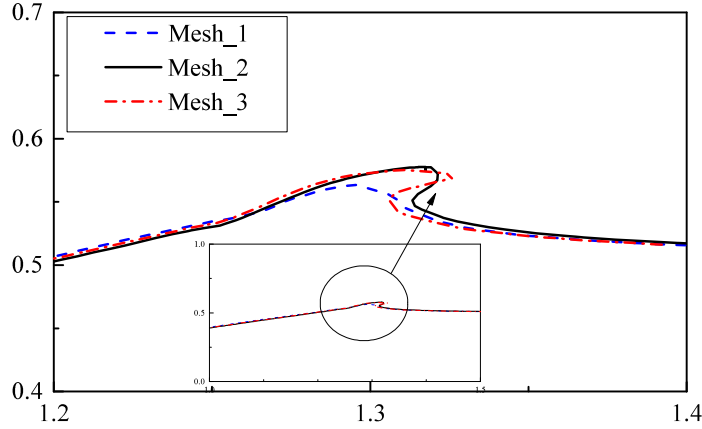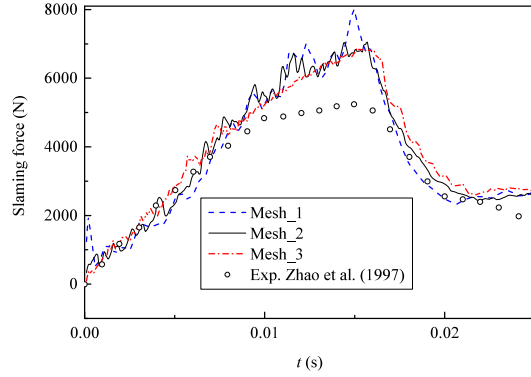
(a) Free surface profile at $t = 0.0202s$
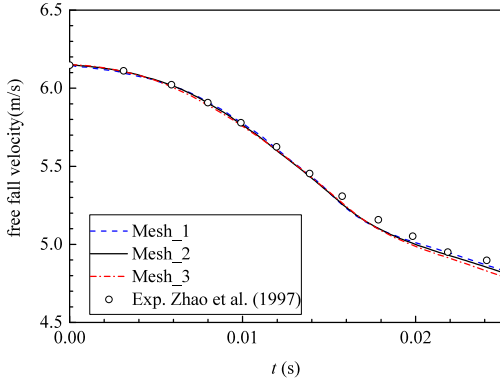
(b) Velocity of the free fall wedge

(c) Slamming force on the wedge

Figure 20: Grid convergence tests for the free fall wedge.

the fully submerged wedge experiences a decreased slamming force, as the wetted surface remains the same. The present numerical results agree well with the experiment data before $t = 0.01s$ and after $t = 0.017s$. Between these two time instants, the force is over-predicted, which generates a larger deceleration and slows down the motion of wedge falling into the water.

In addition, the comparison of two forcing point searching schemes with the same linear interpolation is shown in Fig. 21, where the outside forcing scheme shows better agreement with the experimental data in Zhao et al. (1997) for both the free fall velocity (Fig. 21(a)) and the slamming force (Fig. 21(b)), especially after $t = 0.016s$. Before $t = 0.016s$ the results obtained by these two schemes are quite close. Therefore, the conclusion drawn from the previous three cases that the outside forcing scheme can perform better still stands for this case.

To further test the accuracy of the present combined immersed boundary and level set method, Fig. 22

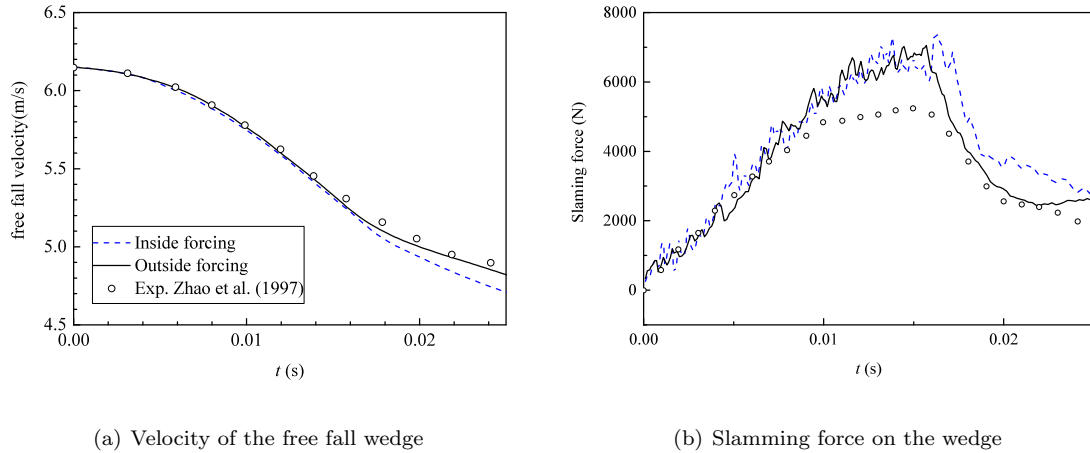(a) Velocity of the free fall wedge

(b) Slamming force on the wedge

Figure 21: Time history of flow characteristics for the free fall wedge obtained by two different forcing point searching schemes.
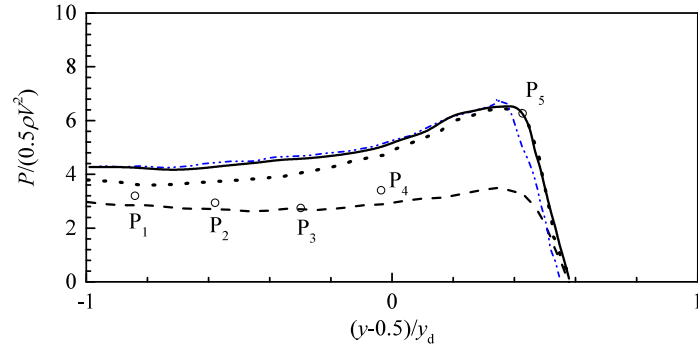
shows the pressure distribution at three different time instants, and the comparison with the experimental measurement by the five pressure gauges located at the wedge surface. At the time instant $t = 0.00435s$, the numerical simulation cannot capture the pressure at the gauge P_2. It should be noted that at this time instant the water just inundates the gauge P_2 in the experiment. Any small delay in the body motion could significantly influence the pressure at this position. It seems that the present numerical prediction of the body motion is slightly slower than that in the experiment. However, the present result still possesses the advantage over that in Zhang et al. (2010) where the inside forcing scheme was used. At $t = 0.0158s$, the numerical simulation over-predicts the pressure at the first four pressure gauges compared to the experimental data, but the present results are closer to the numerical work in Zhao et al. (1997) where the boundary element method was adopted to predict slamming loads on a general two-dimensional body. However, the result in Zhang et al. (2010) is unable to capture the peak pressure at the gauge P_5. At $t = 0.0202s$, the present numerical model shows the best performance over the other two numerical simulations according to the comparison with the experimental data.

In Fig. 22 the present numerical results under both Mesh_2 and Mesh_3 are shown, from which the good convergence can be seen, and Mesh_2 seems to be able to provide the satisfactory simulation. However, the results shown in the same figure from Zhang et al. (2010) for the purpose of comparison are obtained under the fine Mesh_3. Therefore, it indicates that the present outside forcing point searching scheme has a better performance than the one proposed in Zhang et al. (2010).

Finally, Fig. 23 shows the snapshots of free surface profile when the wedge penetrates the still water obtained at the fine mesh. It can be seen that the body shape remains the same to the initial geometry when the wedge moves into the water, and no unrealistic water is observed to flow into the wedge, which proves the capability of the present immersed boundary method in modelling a moving body with complicated

25
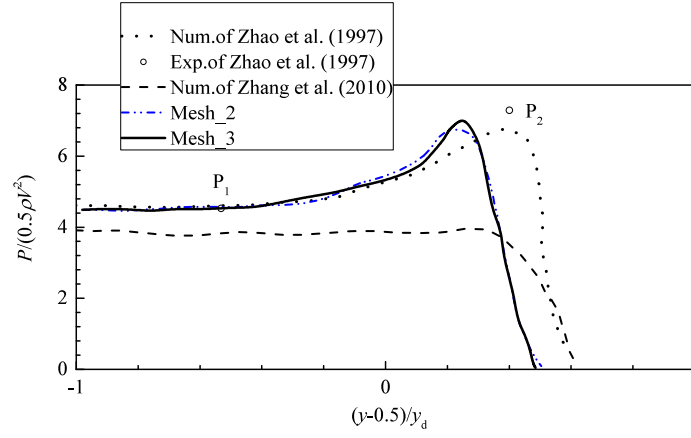
(a) $t = 0.00435s$

(b) $t = 0.0158s$
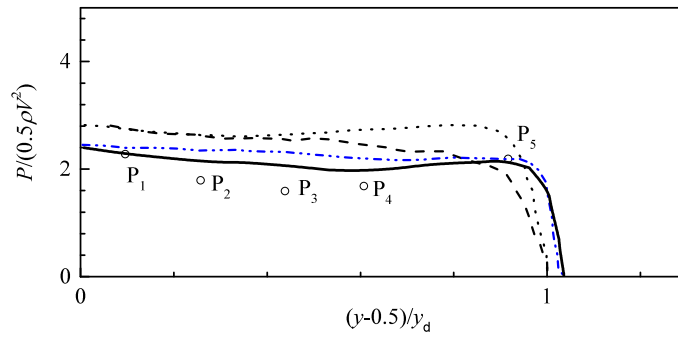
(c) $t = 0.0202s$

Figure 22: Comparison of pressure distributions on the wedge surface at three different time instants. P is the pressure, V represents the wedge vertical velocity, y is the vertical coordinate along the wedge surface, and y_d is the draft of the wedge.

geometry. At $t = 0.015s$, the bottom of the wedge is fully submerged in the water, which is consistent to the discussion on two phases shown in the global hydrodynamic features in Fig. 21. The water jet becomes visible as the time reaches $0.020s$, while at $t = 0.025s$ the jet is fully developed. At the last time instant, the water jet starts to split, a water splashing thus occurs.
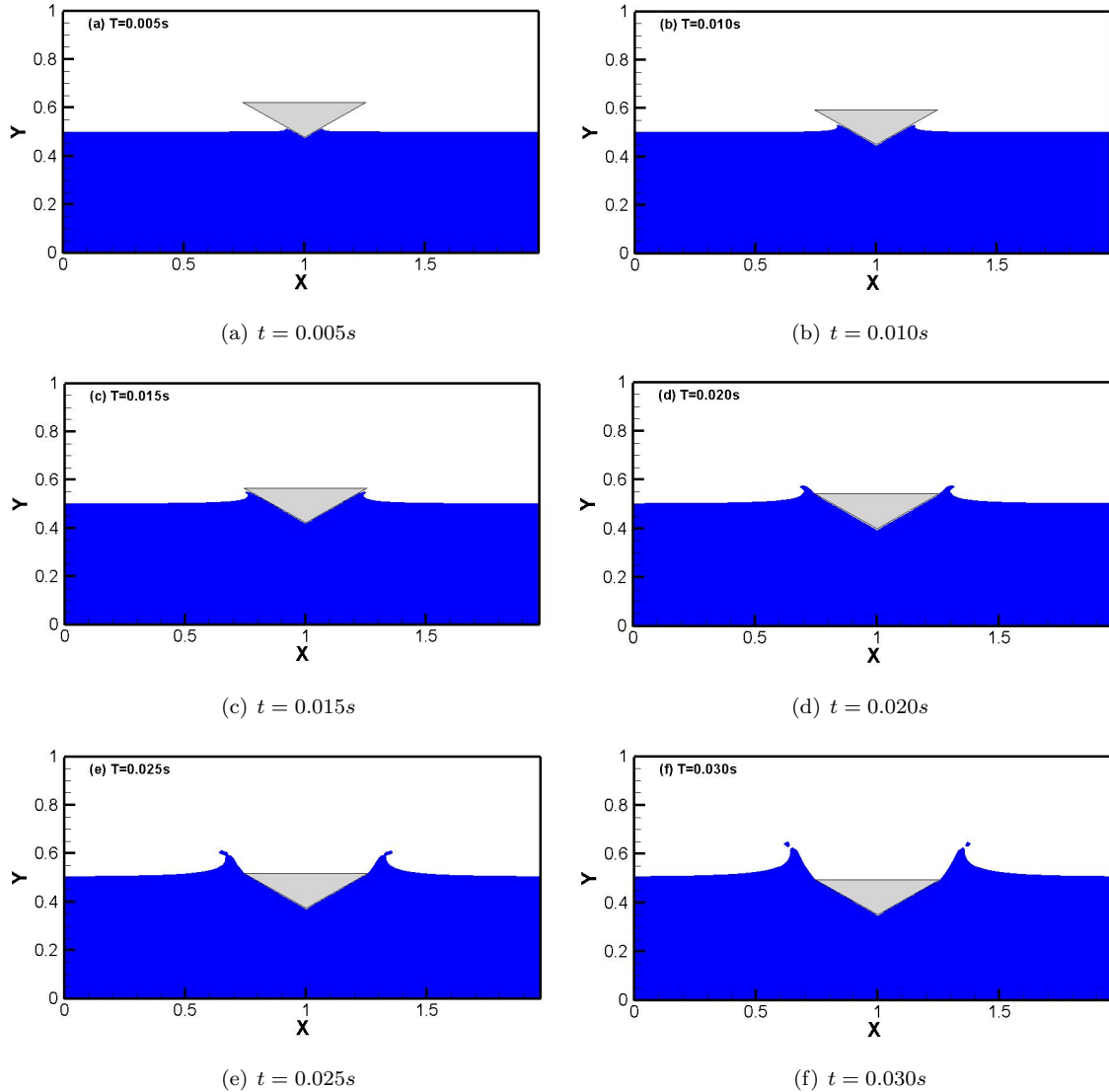

(a) $t = 0.005s$

(b) $t = 0.010s$

(c) $t = 0.015s$

(d) $t = 0.020s$

(e) $t = 0.025s$

(f) $t = 0.030s$

Figure 23: Free surface profiles of free fall wedge in water at various time instants.

6. Conclusions

A new immersed boundary method is proposed to simulate complicated interactions between fluid and fixed or moving structures, in conjunction with the level set method for free surface capturing. In the present numerical model, an effective and straightforward forcing point searching scheme is developed for forcing points located both inside and outside the solid phase. This simpler searching scheme for the determination

27

of velocities at forcing points proposes an advantage of ease of implementation, with remaining desirable accuracy. To validate the effectiveness of the present numerical model, five testing cases, including dam break past a circular cylinder, dam break over a rectangular obstacle, wave travelling over a submerged bar, water entry of a wedge with prescribed velocity and free fall of a wedge with initial velocity are considered.

The dam break cases suggest that the outside forcing points searching scheme leads to better results. Further, comparisons of free surface profile between the numerical and experimental results show considerably good agreement. In the case of wave propagation over a submerged bar, the accuracy of the proposed numerical model is validated by the comparison of wave profile with the experimental results. The numerical results again reveal that the outside forcing scheme is superior over the inside forcing scheme, even though there is only little difference between the results of the two schemes. In addition, the case of water entry of a wedge with prescribed velocity also demonstrates the convergence and accuracy of the proposed numerical model, through the comparisons with the analytical solution and other numerical results. After the extensive validation and comparison through the final case of free fall wedge that is more difficult to simulate due to the existence of both free surface and moving body, the overall numerical results suggest that the present outside forcing point searching scheme is more efficient and shows better performance than the other immersed boundary method in literature.

References

Archer, P. and Bai, W. (2015). A new non-overlapping concept to improve the hybrid particle level set method in multi-phase fluid flows, *Journal of Computational Physics* **282**: 317 – 333.

Balaras, E. (2004). Modeling complex boundaries using an external force field on fixed cartesian grids in large-eddy simulations, *Computers and Fluids* **33**: 375 – 404.

Beji, S. and Battjes, J. A. (1994). Numerical simulation of nonlinear wave propagation over a bar, *Coastal Engineering* **23**: 1 – 16.

Chiu, P. H., Lin, R. K. and Sheu, W. H. T. (2010). A differentially interpolated direct forcing immersed boundary method for predicting incompressible navier-stokes equations in time-varying complex geometries, *Journal of Computational Physics* **229**: 4476 – 4500.

De Rosis, A., Falcucci, G., Porfiri, M., Ubertini, F. and Ubertini, S. (2014). Hydroelastic analysis of hull slamming coupling lattice boltzmann and finite element methods, *Computers and Structures* **138**: 24–35.

Dobrovol'Skaya, Z. N. (1969). On some problems of similarity flow of fluid with a free surface, *Journal of Fluid Mechanics* **36**(4): 805–829.

Fadlun, E. A., Verzicco, R., Orlandi, P. and Mohd-Yusof, J. (2000). Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *Journal of Computational Physics* **61**: 35 – 60.

Goldstein, D., Haandler, R. and Sirovich, L. (1993). Modeling a no-slip flow boundary with an external force field, *Journal of Computational Physics* **105**: 336 – 354.

Huang, C. J. and Dong, C. M. (1999). Wave deformation and vortex generation in water waves propagating over submerged dike, *Coastal Engineering* **37**: 123 – 148.

Jiang, G. S. and Peng, D. (2000). Weighted eno schemes for hamilton-jacobi equations, *SIAM Journal on Scientific Computing* **21**(6): 2126 – 2143.

Kim, J., Kim, D. and Choi, H. (2001). An immersed boundary finite-volume method for simulations of flow in complex geometries, *Journal of Computational Physics* **171**: 132 – 150.

Koshizuka, S., Tamako, H. and Oka, Y. (1995). A particle method for incompressible viscous flow with fluid fragmentation, *Computational Fluid Dynamic Journal* **4**: 29 – 46.

Larese, A., Rossi, R., Oñate, E. and Idelsohn, S. R. (2008). Validation of the particle finite element method (PFEM) for simulation of free surface flows, *Engineering Computations* **25**(4): 385 – 425.

Lin, P. and Li, C. (2002). A σ-coordinate three-dimensional numerical model for surface wave propagation, *International Journal for Numerical Methods in Fluids* **38**: 1045 – 1068.

Mei, X., Liu, Y. and Yue, D. K. P. (1999). On the water impact of general two-dimensional sections, *Applied Ocean Research* **21**(1): 1–15.

Mittal, R., Dong, H., Bozkurttas, M., Najjar, F. M., Vargas, A. and Loebbecke, A. V. (2008). A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *Journal of Computational Physics* **227**: 4825 – 4852.

Mohd-Yusof, J. (1997). Combined immersed boundary/b-spline method for simulations of flows in complex geometries, *Technical report*, Center Annual Research Briefs, NASA Ames/Stanford University.

Peng, D., Merriman, B., Osher, S., Zhao, H. and Kang, M. (1999). A pde-based fast local level set method, *Journal of Computational Physics* **155**: 410 – 438.

Peskin, C. S. (1972). Flow patterns around heart valves: a numerical method, *Journal of Computational Physics* **10**: 252 – 271.

Saiki, E. M. and Biringen, S. (1996). Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method, *Journal of Computational Physics* **123**: 450 – 465.

Sethian, J. A. (1996). A fast marching level set method for monotonically advancing fronts, *Proceedings of the National Academy of Sciences of the United States of America* **93**: 1591 – 1595.

Shen, L. and Chan, E. S. (2008). Numerical simulation of fluidstructure interaction using a combined volume of fluid and immersed boundary method, *Ocean Engineering* **35**: 939 – 952.

Von Karman, T. (1929). The impact on seaplane floats during landing, *Technical Report 321*, National Advisory Committee for Aeronautics.

Wagner, H. (1932). Uber stoss und gleitvorgange an der oberache von flussigkeiten, *Zeit Angew Math Mech* **12**(4): 192–215.

Yan, S. and Ma, Q. W. (2007). Numerical simulation of fully nonlinear interaction between steep waves and 2d floating bodies using the qale-fem method, *Journal of Computational Physics* **221**: 666 – 692.

Yang, J., Preidikman, S. and Balaras, E. (2008). A strongly coupled embedded-boundary method for fluid-structure interactions of elastically mounted rigid bodies, *Journal of Fluids and Structures* **24**: 167 – 182.

Zarghami, A., Falcucci, G., Jannelli, E., Succi, S., Porfiri, M. and Ubertini, S. (2014). Lattice boltzmann modeling of water entry problems, *International Journal of Modern Physics C* **25**(12): 1441012.

Zhang, Y., Zou, Q., Greaves, D., Reeve, D., Alison, H. R., Graham, D., Phil, J. and Lv, X. (2010). A level set immersed boundary method for water entry and exit, *Communications in Computational Physics* **8**(2): 265 – 288.

Zhao, R., Faltinsen, O. and Aarsnes, J. (1997). Water entry of arbitrary two-dimensional sections with and without flow separation, *Proceeding of 21st Symposium on Naval Hydrodynamics*.