

Please cite the Published Version

Yan, Bin, Bai, Wei , Jiang, Sheng-Chao, Cong, Peiwen, Ning, Dezhi and Qian, Ling  (2021)
A three-dimensional immersed boundary method based on an algebraic forcing-point-searching
scheme for water impact problems. *Ocean Engineering*, 233. ISSN 0029-8018

DOI: <https://doi.org/10.1016/j.oceaneng.2021.109189>

Publisher: Elsevier

Version: Accepted Version

Downloaded from: <https://e-space.mmu.ac.uk/627793/>

Usage rights:  [Creative Commons: Attribution-Noncommercial-No Derivative Works 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Additional Information: Author accepted manuscript published by and copyright Elsevier.

Enquiries:

If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from <https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines>)

A three-dimensional immersed boundary method based on an algebraic forcing-point-searching scheme for water impact problems

Bin Yan^a, Wei Bai^{b,*}, Sheng-Chao Jiang^c, Peiwen Cong^c, Dezhi Ning^c, Ling Qian^b

^a*Department of Civil and Environmental Engineering, National University of Singapore, Kent Ridge, Singapore 117576, Singapore*

^b*Department of Computing and Mathematics, Manchester Metropolitan University, Chester Street, Manchester M1 5GD, UK*

^c*State key laboratory of Coastal and Offshore Engineering, Dalian University of Technology, Dalian 116024, China*

Abstract

A new three-dimensional immersed boundary method combined with the level set method for the interface capturing is developed to simulate the interaction between the fixed\moving structure and the two-phase fluid flow. The concept of the forcing point searching scheme developed for the two-dimensional situations in Yan et al. (2018) is extended in the present study to three dimensions, where the determination of intersections between the arbitrary body surface and the Cartesian background grid system is the major issue. This problem can be converted to the prediction of triangle-triangle intersection, which was traditionally solved from the geometrical point of view. Here, an algebraic algorithm is adopted for the triangle-triangle intersection, based on which the forcing points can be determined in the three-dimensional immersed boundary method. This algebraic algorithm is robust for any body geometry and easy for implementation. To demonstrate the accuracy and capability of the developed numerical model, three benchmark testing cases for water impact problems are conducted, including dam break over a fixed obstacle, water entry of a wedge and free decay of a bobber. Extensive comparisons with the experimental data and the numerical results obtained by other immersed boundary methods suggest that the developed immersed boundary method is accurate and effective for both fixed and moving bodies with complex geometries.

Keywords: Immersed boundary method, Level set method, Forcing point, Algebraic algorithm, Triangle-Triangle intersection, Water impact

1. Introduction

Fluid flow interaction with structures is a common problem in nature, which can find massive applications in various disciplines. In particular, with the increasing demand of modern society for energy, different types of marine structures have come into use in offshore and ocean engineering. Therefore, understanding the hydrodynamic performance of those structures under wave actions becomes an essential task for the safe design and efficient operation. Due to the geometry and character, some structures can be simplified to a

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

7 two-dimensional (2D) problem. However, the realistic marine structures are complex and certainly three-
8 dimensional (3D), which makes the modelling of 3D structures in ocean waves a very interesting research
9 problem.

10 In a broad sense, there are two methods to model a structure in the computational domain in the
11 mesh-based approach: body-fitted method and non-conforming method. The recently developed overset
12 method (Ma et al., 2018; Chen et al., 2019) can be considered as a body-fitted method, which requires
13 extensive interpolations to transfer information with the background grid. In the body-fitted method (Yan
14 and Ma, 2007; Yang et al., 2008), curvilinear structured grids or unstructured grids that conform to the
15 body boundary are generated. This method seems straightforward, however, it could require a heavy cost in
16 computational time as well as manpower or even fail when dealing with a complicated body geometry with
17 large displacement. In recent years, the non-conforming method becomes increasingly popular for simulating
18 bodies in fluid-structure interactions. The cut cell method, one non-conforming method, may face difficulty
19 in simulating irregular “interface-cells”, which requires different “special treatments”. Thus, the extension of
20 cut cell method from 2D to 3D is sometimes prohibited if not impossible, although the limited successful work
21 has been shown in Hu et al. (2013). The basic concept of another non-conforming method, the immersed
22 boundary method (Wu et al., 2013; Wu and Young, 2014) is to add a body force to the momentum equations
23 at certain points around the boundary, without the necessity of performing the mapping procedures, aiming
24 to model the effect of investigated bodies in the flow. By means of this robust method, generation of grids
25 can be greatly simplified. The immersed boundary method was firstly proposed by Peskin (1972) to simulate
26 the cardiac flow with the feedback forcing scheme, which had one disadvantage of requiring small time step.
27 To resolve such problem, Mohd-Yusof (1997) proposed a discrete forcing scheme, which was implemented
28 successfully by Fadlun et al. (2000); Kim et al. (2001); Zhang et al. (2010); Wu (2019).

29 For the 3D situation, the immersed boundary method was extended by Fadlun et al. (2000) on a staggered
30 grid with volume fraction to determine the forcing point location, which is rather simple but inaccurate. Both
31 Fadlun et al. (2000) and Kim et al. (2001) conformed the background grids to the sphere surface by the
32 use of cylindrical coordinate system. They aimed to enable the forcing points to coincide with the sphere
33 surface manually. In addition, unstructured triangular grid was adopted by Gilmanov and Sotiropoulos
34 (2005) in their 3D immersed boundary method to approximate the sharp body interface, where the nodes
35 near the boundary were utilized to determine whether a given point was an internal or external forcing
36 point, according to the sign of scalar product between the position vector and the normal vector. Based on
37 the work in Kim et al. (2001), Kim and Choi (2006) applied the immersed boundary method to simulate
38 moving bodies via the non-inertial reference frame. As the non-inertial reference frame system was fixed on
39 the body, the procedure for determining forcing terms was the same as that for a stationary case. Similar
40 to Gilmanov and Sotiropoulos (2005), Mittal et al. (2008) also employed the concept of scalar product to
41 determine the forcing points. However, the scalar product method requires many product iterations in the
42 whole domain, which causes the computational effort increase dramatically in the 3D cases.

43 Borazjani et al. (2008) proposed a curvilinear immersed boundary method for modeling complex 3D rigid
44 bodies represented by unstructured triangular grids. In their work, the point-in-polyhedron identification
45 procedure was used to locate forcing points, which starts by emitting a line from a given point P (to be
46 identified) to a point S outside the polyhedron, and the number of intersections is counted to determine
47 whether P is in the solid phase. Similarly, a ray-tracing procedure was adopted in Roman et al. (2009) to
48 address situations involving multiple or concave immersed boundaries. However, both Borazjani et al. (2008)
49 and Roman et al. (2009) did not show the procedure to handle the special situations when the ray passes
50 through the vertices of a triangular cell. This special situation always causes confusion in the determination
51 of intersection numbers, leading to the inaccurate forcing point positions.

52 Recently, Seo and Mittal (2011) developed a sharp-interface immersed boundary method with a partial
53 cut-cell based approach to strictly satisfy the boundary condition with geometric conservation. Spurious
54 pressure oscillations can be reduced, but the treatment was always complicated and the simulations were
55 limited to single-phase flows. To suppress the spurious force oscillations in the simulations of moving bodies,
56 a fully-implicit ghost-cell immersed boundary method was developed in Lee and You (2013), based on the
57 work in Mittal et al. (2008), by combining the mass source/sink algorithm and implicit discretizations. Fur-
58 thermore, Calderer et al. (2014) developed a CURVIB immersed boundary method for calculating the forces
59 on a body by imposing the pressure projection boundary condition (PPBC). To define forcing points, the
60 point-in-polyhedron algorithm in Borazjani et al. (2008) was used to classify all nodes in the computational
61 domain. Again based on Mittal et al. (2008), Nicolaou et al. (2015) proposed a robust immersed boundary
62 method on the curvilinear grids with semi-implicit discretizations to solve the Navier-Stokes equations. How-
63 ever, the force term applied in the solid phase required an iterative approach, which was time-consuming
64 despite the smaller errors at the boundary. In Bihs and Kamath (2017), the approach for the geometry
65 description with a local directional ghost cell method based on Berthelsen and Faltinsen (2008) was pro-
66 posed. To determine the ghost cell, the extrapolation along the normal direction of the regular or irregular
67 boundary was adopted, which needs to be conducted in the x -, y -, and z -directions, respectively.

68 For the 2D problem, the authors have developed an efficient and straightforward scheme on the Cartesian
69 background grid to identify forcing points in the immersed boundary method (Yan et al., 2018), which
70 exhibits the advantage of ease of implementation with desirable accuracy. In addition, it locates the u
71 and v forcing points precisely to ensure the accurate boundary enforcement rather than introducing any
72 assumption or approximation in the determination of forcing points. However, the extension from 2D to
73 3D problem requires tremendous effort. The discussion on past studies shows that approaches to locate 3D
74 forcing points mainly include simplification for special bodies, scalar product of point vector and surface
75 normal vector and ray-tracing scheme. They are all directed at adopting a geometrical view to obtain the
76 positions of intersection and hence determine forcing points. Therefore, they may be more complicated and
77 face more challenges when dealing with complicated 3D geometries. Different to the geometrical view, in
78 the present study an algebraic algorithm for predicting triangle-triangle intersections is developed to locate

79 forcing points. The algebraic algorithm was originally proposed in the discipline of computational graphics
80 (Tropp et al., 2006), with the good features of being robust and efficient. By following the basic concept
81 developed for 2D situations, the main advantages of the immersed boundary method developed in Yan
82 et al. (2018) can be carried over to 3D situations. With the combination of level set method, several water
83 impact problems with complex breaking surface are investigated in the present study. To validate the newly
84 developed 3D immersed boundary method, the present results are compared with other experimental and
85 numerical results extensively for both the fixed and moving 3D structures with complex geometries.

86 2. Two-phase flow model

87 To simulate the 3D incompressible viscous flows, a two-phase flow model is adopted in the present study.
88 This model was developed in Archer and Bai (2015) where more details have been discussed, and only the key
89 information is provided in the following. The fluid motion can be described by the Navier-Stokes equations,

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

90 and the continuity equation,

$$\frac{\partial u_i}{\partial x_i} = 0, \quad (2)$$

91 where i and j denote the x, y, z directions ($i, j = 1, 2, 3$), u_i is the fluid velocity, x_i is the spatial coordinate, t
92 is the time, p is the pressure, g_i is the gravitational acceleration, ρ is the fluid density and f_i is the momentum
93 forcing component used to enforce the desired boundary condition on an immersed boundary interface in
94 the present study. τ_{ij} are the viscous stress components, which is dependent on the dynamic fluid viscosity
95 μ . Here the Cartesian tensor notation is used.

96 The level set method is adopted to capture the interface (free water surface) between the two water-air
97 phases. In the level set method, a signed distance function ϕ is defined throughout the domain to measure
98 the shortest distance from the grid cell center to the interface. A positive value of ϕ indicates one phase while
99 a negative value indicates the other. The zero value of level set function represents the interface position.
100 The evolution of the level set function ϕ satisfies the following convective equation,

$$\frac{\partial \phi}{\partial t} + u_i \frac{\partial \phi}{\partial x_i} = 0. \quad (3)$$

101 As the two-phase flow model needs to consider air and water simultaneously, ρ and μ in the Navier-
102 Stokes equations should be determined by the properties of the local fluid phase. However, the interface
103 between the air and water is sharp, so the sudden change in the values of ρ and μ between two phases could
104 cause instabilities when solving the Navier-Stokes equations. To overcome such problem, a smeared form of
105 Heaviside function H is applied in a small band around the interface to smooth out ρ and μ between two
106 phases. Therefore, ρ and μ can be calculated by

$$\begin{aligned}\rho(\phi) &= \rho_{air} + H(\phi)(\rho_{water} - \rho_{air}) \\ \mu(\phi) &= \mu_{air} + H(\phi)(\mu_{water} - \mu_{air})\end{aligned}\tag{4}$$

107 where the subscripts *air* and *water* denote the values of air and water, respectively.

108 The Navier-Stokes equations are solved numerically by the finite difference method on a staggered grid
109 system. The level set equation (Eq. 3) is discretised by a fifth-order scheme to ensure the accuracy of the
110 calculation. For the detailed numerical discretisation and implementation, the reader can refer to Archer
111 and Bai (2015).

112 To calculate the responses of a 3D moving body in 6 degrees of freedom, the motion equations for the
113 translational and rotational responses can be adopted based on the Newton's second law. As the body only
114 undergoes translational motions in the present study, the motion equations in the translational directions
115 are considered, which take the following general form for a body with the damping and spring system,

$$M \frac{\partial^2 Y^i}{\partial t^2} + C \frac{\partial Y^i}{\partial t} + KY^i = F_{fluid}^i + F_{ext}^i,\tag{5}$$

116 where the superscript *i* denotes the direction as in Eq. 2, Y^i is the displacement in the *i*th direction, M
117 the mass of the moving body, C the damping coefficient, K the spring stiffness coefficient, F_{fluid}^i the force
118 exerted by the fluid, and F_{ext}^i the external force. In the following cases, C is required to be determined
119 by the numerical simulation, and the spring system is not considered. Furthermore, the structure is only
120 subjected to the force exerted by the fluid F_{fluid}^i , which can be expressed as

$$F_{fluid}^i = \int_{\Omega} (-p n_i) d\Omega + \int_{\Omega} (\tau_{ij} n_j) d\Omega,\tag{6}$$

121 where the first and second parts on the right of the equation represent the pressure force and shear force
122 obtained by the integration over the body surface Ω . Based on the obtained displacement, the velocity of
123 the structure can be predicted by

$$u_i = \frac{\partial Y^i}{\partial t},\tag{7}$$

124 where u_i is the velocity component at the *i*th direction.

125 3. 3D immersed boundary method

126 In the present method, the triangle to triangle intersection is solved algebraically that can locate the u ,
127 v , and w forcing points respectively. It guarantees the precise body boundary in a relatively coarser grid
128 compared to other methods, such as the simplification for special bodies, a scalar product of point vector
129 and surface normal, and the ray-tracing scheme. The scalar product and ray-tracing scheme require much
130 computational cost for iterations, while the algebraic algorithm adopted here is easy for the speed-up in the

131 matrix solution. Meanwhile, by solving the matrix the accuracy of forcing points can be ensured and the
132 missing forcing points can be avoided.

133 3.1. Solid geometry description

134 Firstly, structures in the computational domain need to be defined and described in a certain way. In
135 the present study, the surface of a solid body is described by a series of unstructured triangular grids. The
136 unstructured triangular grids are generated by means of the open source software “NETGEN” (Schöberl,
137 1997). NETGEN is an automatic mesh generation tool in both the 2D and 3D spaces, and it can generate
138 triangular or quadrilateral grids, as well as tetrahedral grids. To generate unstructured triangular grids on
139 the curved surface of a 3D solid body, the constructive solid geometries (CSG) or the standard template
140 library (STL) file format is created first and then imported into NETGEN. Once the geometry of the solid
141 body is defined by NETGEN, the information of nodes on the body surface is provided to the present
142 numerical model for the subsequent determination of forcing points near the body surface.

143 3.2. Triangle-triangle intersection in 3D space

144 With the description of a solid body, it is possible to find the forcing points. In the present study, the
145 forcing points locate outside the solid phase (surface mesh), which has been demonstrated to give better
146 performance in Yan et al. (2018) for the 2D situation. In order to apply the basic concept of the improved
147 forcing point searching scheme developed for the 2D case in Yan et al. (2018) to the present 3D problem, the
148 key is to project the body surface onto the corresponding plane, so that the 3D problem can be converted to
149 the 2D problem to some extent. To construct the projection area, three bounded planes are adopted, which
150 are shown in Fig. 1(a) as red squares. Those three bounded planes are denoted as the uv , uw , vw planes,
151 respectively, according to the directions of two velocity components that belong to the plane, each of which is
152 also perpendicular to a particular coordinate axis. For the convenience of capturing the intersection between
153 the plane and the body surface, each bounded plane is further divided into two triangles. Therefore, the
154 problem is transferred to the determination of 3D triangle-triangle intersections, before the forcing points
155 can be found.

156 For the purpose of further demonstration, one triangle on the body surface ($\triangle ABC$ shown in Fig. 1(a))
157 is taken as an example. To search the forcing points, one needs to find the interaction between $\triangle ABC$ and
158 the uv bounded plane (for example), which is the line segment cd as shown in Fig. 1(a). Fig. 1(b) shows
159 the sketch of $\triangle ABC$, uv bounded plane and intersection line segment cd only. It should be noted that the
160 bounded plane is divided into two triangles by a diagonal line. By using the triangle-triangle intersection
161 algorithm, the intersection line cd can be found. As a result, the dimension of the problem is reduced from
162 three to two in a sense.

163 As the intersection between the edge of one triangle with another triangle can be described by a linear
164 equation, there are in total six such equations to be solved for the triangle-triangle intersection problem. A
165 few fast representative algorithms (Moller, 1997; Held, 1997; Guigue and Devillers, 2003) were proposed by

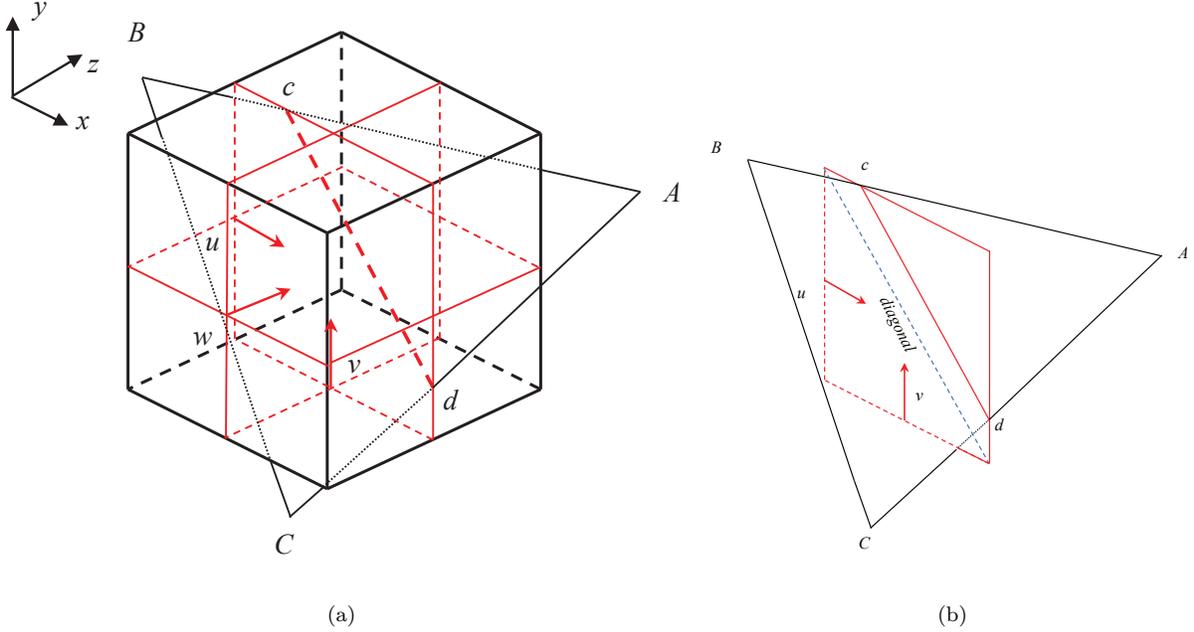


Figure 1: Triangle-triangle interaction in the simulation: (a) overall view (b) $\triangle ABC$ for clear demonstration.

166 the utilization of the intersection line between the planes supported by the two triangles. In addition to the
 167 above-mentioned algorithms to solve the intersection problem geometrically in the conventional view, Tropp
 168 et al. (2006) proposed an algebraic approach. As there is a strong relationship among the linear equations,
 169 this relationship can be applied to expedite computational efficiency of the solution. In other words, the
 170 algebraic algorithm requires fewer arithmetic operations than those geometric algorithms, and the accurate
 171 intersection coordinates could be obtained with a much lower effort than before. Thus, we introduce the
 172 algebraic algorithm into the development of 3D immersed boundary method.

173 For completeness, we recapitulate the theory for the intersection between two triangles presented in Tropp
 174 et al. (2006) in the following. In this algorithm, Tri A and Tri B are defined as two triangles. The intersection
 175 between Tri A and Tri B means that the edges belonging to one triangle intersect the other triangle. With
 176 all possible edge being examined, the intersection could be determined. To demonstrate, define \vec{p}_1 and \vec{p}_2
 177 as the edges of Tri B sharing the same vertex P_1 , and the edges of Tri A originating from vertices Q_i as
 178 \vec{q}_i ($1 \leq i \leq 3$) (see Fig. 2). To reside the intersection coordinates between two supporting planes of two
 179 triangles, the following set of equations is solved:

$$P_1 + \alpha_1 \times \vec{p}_1 + \alpha_2 \times \vec{p}_2 = Q_i + \beta_i \times \vec{q}_i \quad (8)$$

180 where α and β_i are the scalar factor to be determined in the following content.

181 As the intersection point is inside the triangle, Eq. 8 should be solved with considering the inequalities:
 182 $0 \leq \beta_i \leq 1$, $\alpha_1 \alpha_2 \geq 0$ and $\alpha_1 + \alpha_2 \leq 0$. Six such intersection tests ought to be carried out: three to examine

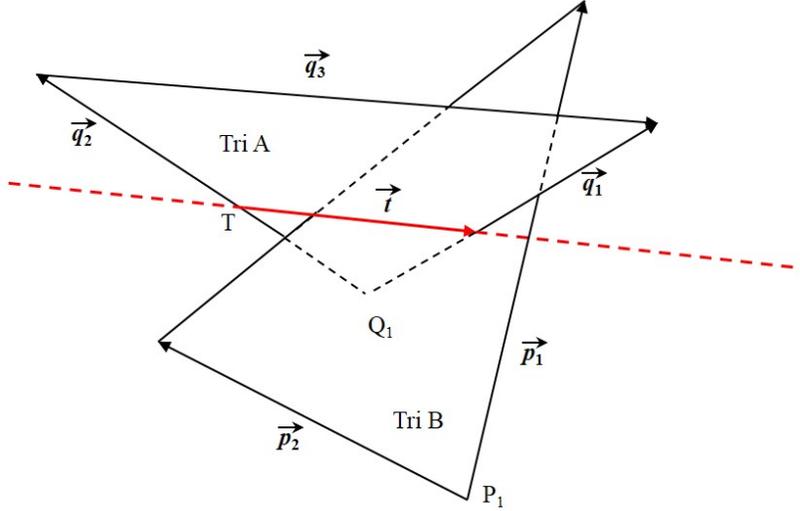


Figure 2: Problem setting for triangle-triangle intersection.

183 whether Tri A intersects the edges of Tri B, and three to examine whether Tri B intersects the edges of Tri
 184 A. By reusing the common elements and utilizing the linearity of matrix operations, the algebraic algorithm
 185 is applied only to three equations.

186 In the beginning, three linear equations can be partially solved to determine the parameters β_i , $1 \leq i \leq$
 187 3, namely the three edges of Tri A intersecting with the supporting plane of Tri B (Eq. 8). The values of
 188 β_i can determine whether a fast rejection could occur, concluding no intersection. If such rejection does not
 189 happen, the values of β_i are combined with the vertices and edges to construct the intersection line between
 190 Tri A and the supporting plane of Tri B (vector \vec{t} shown in Fig. 2). Thus, the 3D problem is reduced to a
 191 planar intersection between the intersection line segment \vec{t} and Tri B. The flow chart of different stages of
 192 this algorithm is shown in Figure 3.

193 At the first step, Eq. 8 can be rewritten in the matrix form with $r_i = Q_i - P_1$,

$$(\vec{p}_1 \quad \vec{p}_2 \quad \vec{q}_i) \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ -\beta_i \end{pmatrix} = (r_i). \quad (9)$$

194 Letting the matrix $\mathbf{A}(\vec{q}_i) = (\vec{p}_1 \quad \vec{p}_2 \quad \vec{q}_i)$, Eq. 9 can be written in the simplified form,

$$\mathbf{A}(\vec{q}_i)\mathbf{x}_i = r_i \quad (10)$$

195 where $\mathbf{x}_i = (\alpha_1, \alpha_2, -\beta_i)$. As the intersection point always resides on the edge \vec{q}_i , the legal β_i is subjected
 196 to the inequality $0 < \beta_i < 1$. β_i is obtained via the determinants, $\beta_i = -|\mathbf{A}(\vec{q}_i)|/|\mathbf{A}(r_i)|$. Thus, the
 197 equivalent inequality condition becomes $0 < \beta_i|\mathbf{A}(r_i)|^2 < |\mathbf{A}(\vec{q}_i)|^2$ mathematically. As a result, $0 <$
 198 $-|\mathbf{A}(r_i)||\mathbf{A}(\vec{q}_i)| < |\mathbf{A}(r_i)|^2$ can be employed to examine whether the division is required.

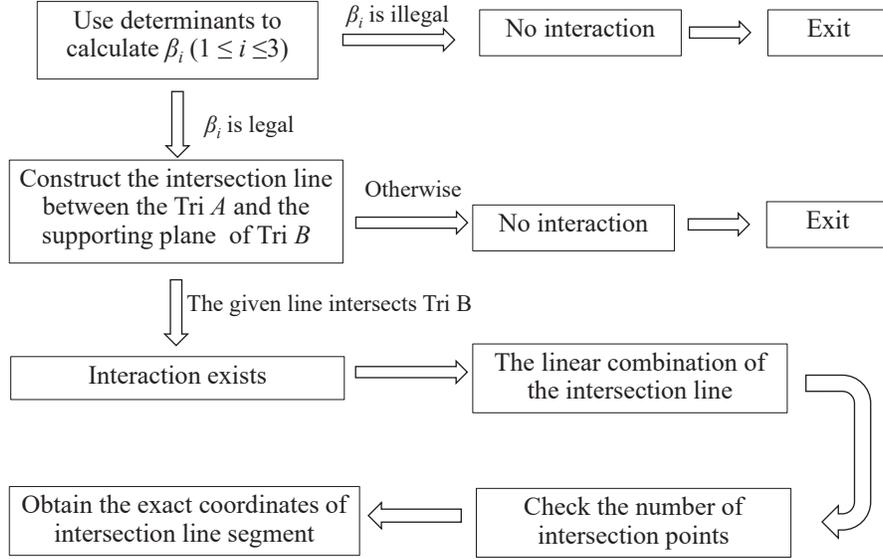


Figure 3: The flow chart for triangle-triangle intersection.

199 At the second stage, if all the values of β_i are illegal, Tri A is located on one side of the supporting plane
 200 of Tri B without any intersection or parallel to the supporting plane of Tri B . This situation concludes
 201 no intersection directly. If $|\mathbf{A}(\vec{q}_i)| = 0$ is always satisfied, Tri A and Tri B are overlapped in one plane.
 202 Otherwise, discussion on the common situation continues as follows.

203 At the third step, the following equations can be utilized to obtain the intersection point T and the line
 204 segment \vec{t} . For instance, if the legal parameters (β_1, β_2) exist, T and \vec{t} can be written as,

$$T = Q_1 + \beta_1 q_1, \vec{t} = \beta_2 q_2 - \beta_1 q_1. \quad (11)$$

205 At the stage four, two cases occur when the two triangles intersect each other: 1) there is at least one
 206 intersection between the surface of Tri B and the line segment \vec{t} ; 2) the line segment \vec{t} is fully enclosed
 207 by Tri B . Thus, the following equations for intersection coordinates shall be solved:

$$\begin{pmatrix} P_1 + \delta_1 \vec{p}_1 = T + \gamma_1 \vec{t} \\ P_1 + \delta_2 \vec{p}_2 = T + \gamma_2 \vec{t} \\ P_1 + \delta_3 (\vec{p}_2 - \vec{p}_1) = T + \gamma_3 \vec{t} \end{pmatrix}. \quad (12)$$

208 The legal values of β_i lead to the residence of intersection on the edge \vec{p}_i . Only when $0 < \gamma_i < 1$ the
 209 intersection points are on the line segment \vec{t} . Only when both the inequalities $(0 < \gamma_i < 1, 0 < \beta_i < 1)$ are
 210 satisfied, the intersection exists between the edge \vec{p}_i and the line segment \vec{t} , with the coordinate of \mathbf{X}_i ,

$$\mathbf{X}_i = P_i + \delta_i \vec{p}_i \quad (13)$$

211 At the final optional fifth stage, the parameters (β_i) and vectors can be combined to obtain the exact
 212 intersection coordinates. T and $T + \vec{t}$ or \mathbf{X}_i and $T + \vec{t}$ shall be the required endpoints of the line segment.

213 3.3. Determination of forcing points and forcing terms

214 After the determination of the intersection line segment (such as cd shown in Fig. 1), the forcing points
 215 can be predicted. Here, take the uv plane as an example for the purpose of illustration. In Fig. 4(a), the
 216 red solid line is an intersection line segment that has been determined by the algorithm discussed in the last
 217 section, and \mathbf{n} is the unit normal vector pointing outwards. If the slope of the line segment is smaller than
 218 that of the diagonal line, the line segment is extended to a new grid in the y direction. However, if the line
 219 segment has already touched both two vertical sides of the present grid, the extension is not required. It is
 220 noted that the direction of extension of the line segment depends on the normal vector direction. The u and
 221 v positions closest to the line segment can be calculated by the linear equation of intersection line segment.
 222 According to the normal direction, u and v forcing points are located in the same direction of \mathbf{n} , shown as
 223 the red u and v arrows in Fig. 4(a). Otherwise, the forcing points are located at the black u and v positions
 224 if the normal direction is in the opposite way. On the other hand, if the slope of the line segment is larger
 225 than that of the diagonal line, the extension of line segment is in the horizontal direction, as shown in Fig.
 226 4(b). Similar to Fig. 4(a), the u and v positions closest to the line segment can be determined. Based on
 227 the normal direction, the u and v forcing points are located in the same direction of \mathbf{n} , shown as the red u
 228 and v arrows in Fig. 4(b). Otherwise, the forcing points are the black u and v if the normal vector direction
 229 is opposite. The following equation is aimed to represent such procedure,

$$\text{forcing point position} = \begin{cases} (x_i, y_{j+1}), & \text{if } n_j \geq n_i \\ (x_{i+1}, y_j), & \text{if } n_i \geq n_j \end{cases} \quad (14)$$

230 where n_i, n_j are the x, y - components of the normal vector \mathbf{n} .

231 It is the same to determine the information of u, w forcing points in the uw plane and v, w forcing points
 232 in the vw plane, respectively. Based on our developed algorithm for the 2D case in Yan et al. (2018), the
 233 forcing points closest to the body surface are selected and stored for the interpolation procedure.

234 Upon the location of forcing points is determined, the velocity at the forcing point v_f needs to be predicted
 235 by the interpolation (more details are also given in Yan et al., 2018). Once the velocity at the forcing point
 236 is available, the forcing term component at the forcing point is predicted based on the method described in
 237 Mohd-Yusof (1997),

$$f_i = \frac{v_f - v_i^n}{\Delta t} - RHS_i^n. \quad (15)$$

238 where RHS is the sum of the convective, viscous, pressure gradient and body force terms in the governing
 239 equations, and the superscript n denotes the values taken from the previous time step.

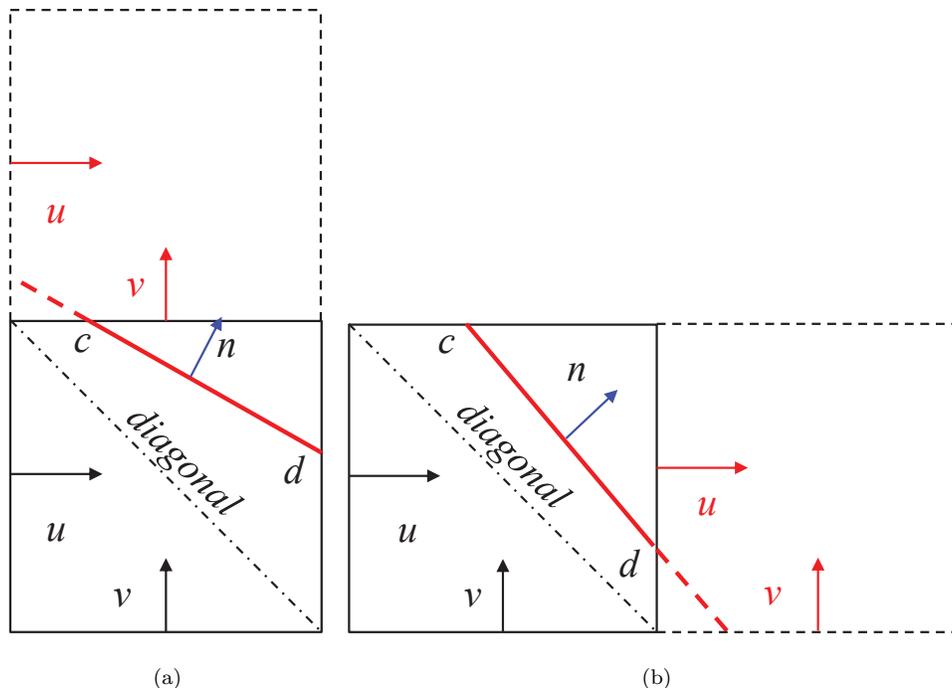


Figure 4: Sketch of determination of forcing points in a 2D plane: (a) slope is smaller than diagonal line; (b) slope is larger than diagonal line.

240 4. Validations and applications

241 To validate the effectiveness and accuracy of the proposed 3D immersed boundary method, three cases
 242 are conducted: 3D dam break over a cuboid, free fall of a 3D wedge, and free decay of a bobber. It aims to
 243 demonstrate the capability of our model to simulate fixed and moving 3D complex bodies in the fluid flow
 244 with complicated free surface.

245 4.1. 3D Dam break over a cuboid

246 In this case, the numerical model is adopted to simulate a physical experiment carried out in a short tank
 247 of $3.22\text{m} \times 1.0\text{m} \times 1.0\text{m}$, at the laboratory of Maritime Research Institute Netherlands (MARIN), which
 248 was reported in Kleefsman et al. (2005). To simulate the dam break, a door encloses a water column with
 249 1.22m width and 0.55m height on the right side of the tank. After the door is pulled up, the water column
 250 breaks and flows to the other side of the tank. A cuboid is fixed to resemble a scaled model of a deck-house
 251 on the deck of a vessel in the tank. Water heights and pressures were measured at the specified positions as
 252 shown in Fig. 5. As the behaviour in the middle of the z plane is more concerned, the pressure gauges are
 253 all set in the symmetrical plane in the z direction. The coordinates (x, z) of the water gauges H1 and H2
 254 are $(0.56, 0.5)$ and $(2.22, 0.5)$. The coordinates (x, y, z) of the pressure gauges P1 \sim P4 are $(2.4, 0.025, 0.5)$,
 255 $(2.4, 0.01, 0.5)$, $(2.425, 0.16, 0.5)$ and $(2.45, 0.16, 0.5)$. The unit of the coordinates is meter.

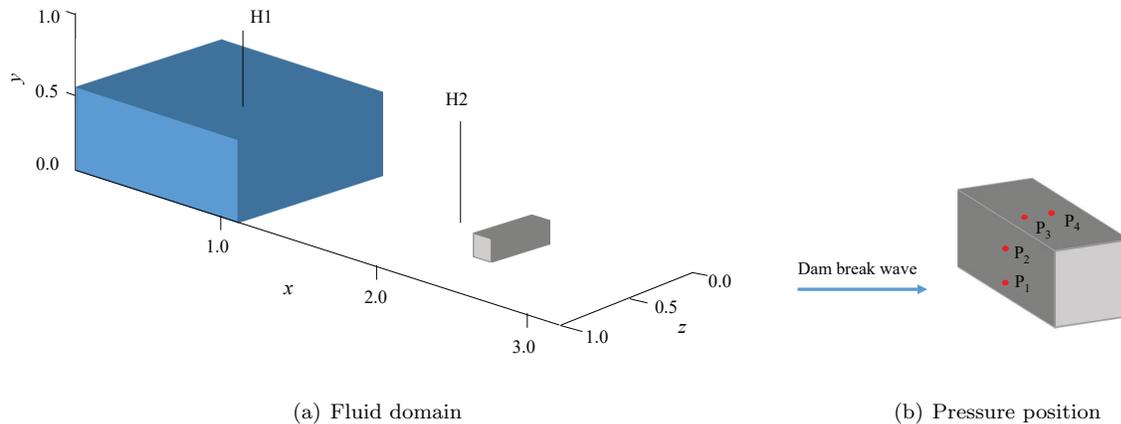


Figure 5: Sketch of 3D dam break over a cube and measured positions.

256 The surface of the cuboid needs to be described first. In the present numerical simulation, the cuboid
 257 is discretized with 12 vertices and 20 triangular surface elements, as shown in Fig. 6. To test the mesh
 258 convergence, three grids around the structure are examined with intervals of $0.04\text{m} \times 0.04\text{m} \times 0.04\text{m}$
 259 (coarse mesh), $0.02\text{m} \times 0.02\text{m} \times 0.02\text{m}$ (medium mesh) and $0.01\text{m} \times 0.01\text{m} \times 0.01\text{m}$ (fine mesh) in the x , y
 260 and z directions respectively. In the other area, the uniform grids of $0.04\text{m} \times 0.04\text{m} \times 0.04\text{m}$ are generated.
 261 The pressures at the gauges P1 and P3 are presented for those three grids in Fig. 7. From the figure, it can
 262 be observed that the results for the medium and fine meshes are close to each other, while the result for the
 263 coarse mesh deviates much from the other two results. Therefore, the convergent results are achieved when
 264 the medium mesh is used.

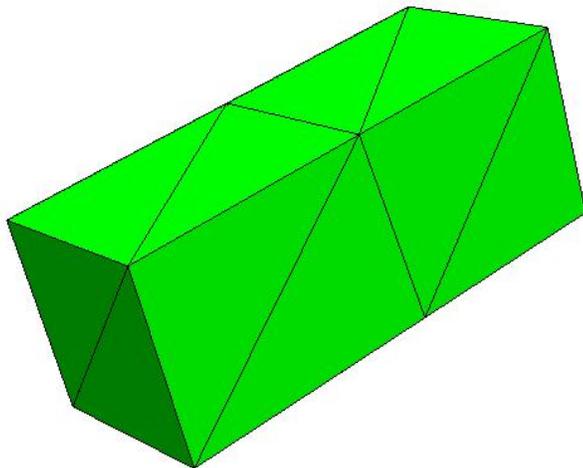
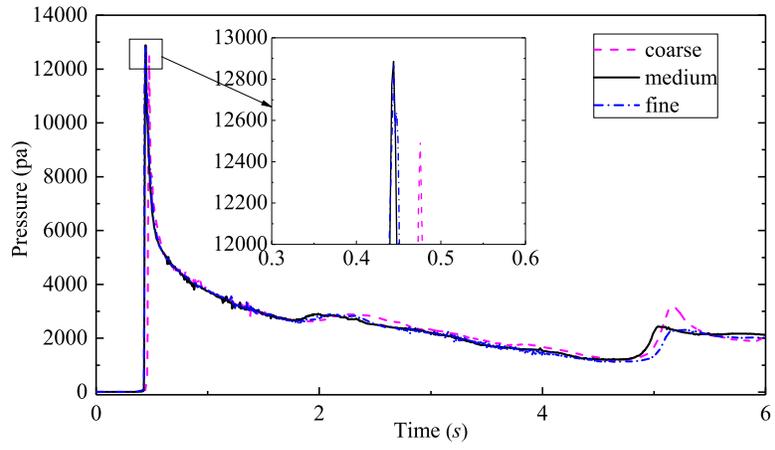
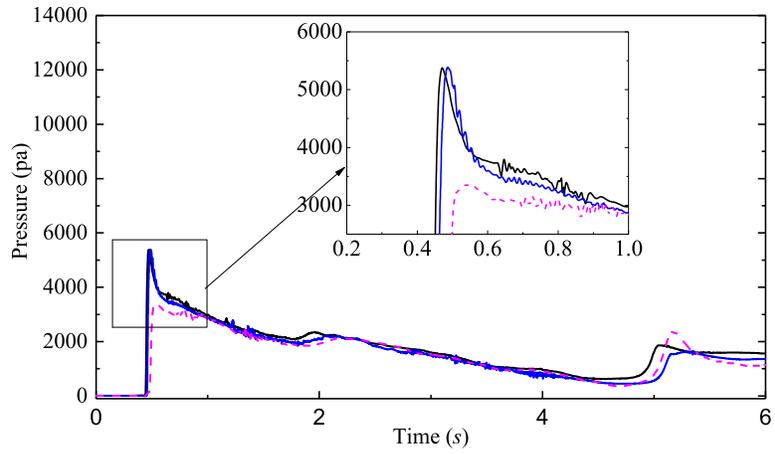


Figure 6: Mesh details of the surface of a cube (12 nodes and 20 surface elements).

265 Fig. 8 presents a comparison of time histories of water height with the experimental data and other
 266 numerical results. The results in Kleefsman et al. (2005) were computed using the model Comflow and Gu



(a) P1



(b) P3

Figure 7: Time history of pressures at P1 and P3 for the mesh convergence test.

267 et al. (2013) adopted the partial cell technique. In general, all the three numerical results deviate a little from
 268 the experimental one with a phase lag. However, the present results seem to capture the magnitude better,
 269 while the numerical results of both Kleefsman et al. (2005) and Gu et al. (2013) underestimate the peak
 270 value in Fig. 8(a) and overestimate the peak value in Fig. 8(b). At the wave gauge H1, the present result
 271 shows a delay in the peak value, which means the reflected water from both the cuboid and the end wall
 272 arrives the wave gauge slightly later. This is probably due to the fact that one disadvantage of the level set
 273 method adopted in the present study for the interface capturing lies in the poor guarantee of conservation
 274 of mass for the complex water surface with the fluid-structure interaction. The developed 3D immersed
 275 boundary method seems to perform well, as the free surface elevation at the gauge H2 which is just in front
 276 of the cuboid shows a good agreement with the experimental data.

277 To further demonstrate the capability, the time histories of pressures at four positions are presented in
 278 Fig. 9. It can be seen that there are suspicious spurs in the results of Kleefsman et al. (2005), while the
 279 present results and the results of Gu et al. (2013) shows the good feature of stability. Compared to the
 280 results of Gu et al. (2013), the present results agree better with the experimental data, as their results
 281 underestimate the first peak. Fig. 10 shows the water surface profiles at $t = 0.0s, 0.5s, 0.75s, 1.0s, 1.25s$ and
 282 $2.0s$ respectively, with the grid of $0.01m \times 0.01m \times 0.01m$. As the water impacts on the cuboid, the water
 283 surface exhibits the process of separation, breaking and mixing. The splashing water jets occur between the
 284 time instants $t = 0.75s$ and $t = 1.25s$. The complicated water surface is generated mainly when the water
 285 impacts on the front side of the obstacle and the end wall of the tank, as shown in Figs. 10(c) and 10(d).
 286 Fig. 10(e) captures the most chaotic and breaking features of the water surface.

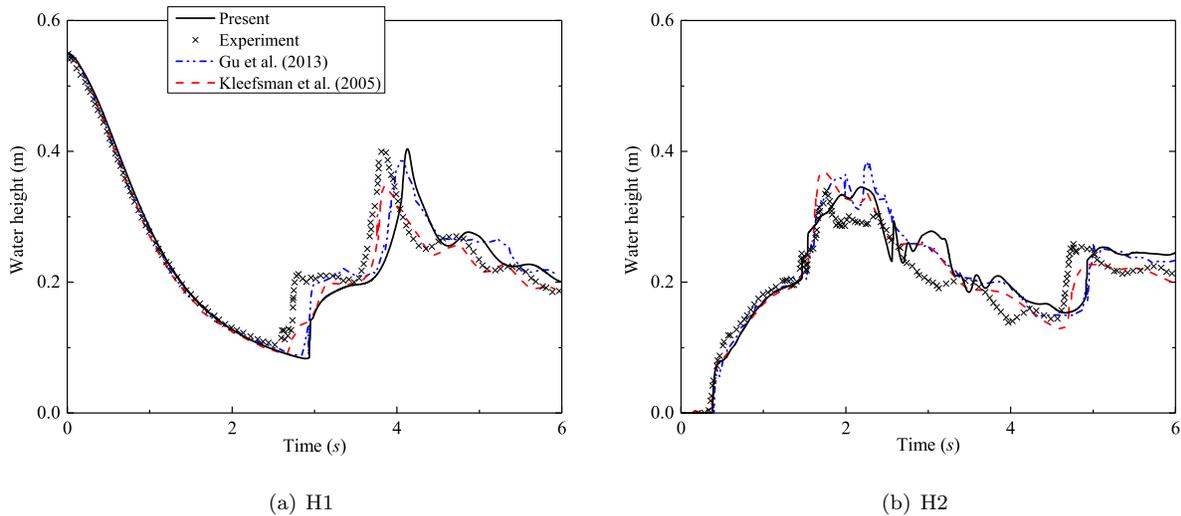


Figure 8: Comparison of water heights in the gauges H1 and H2 with the experimental and other numerical results.

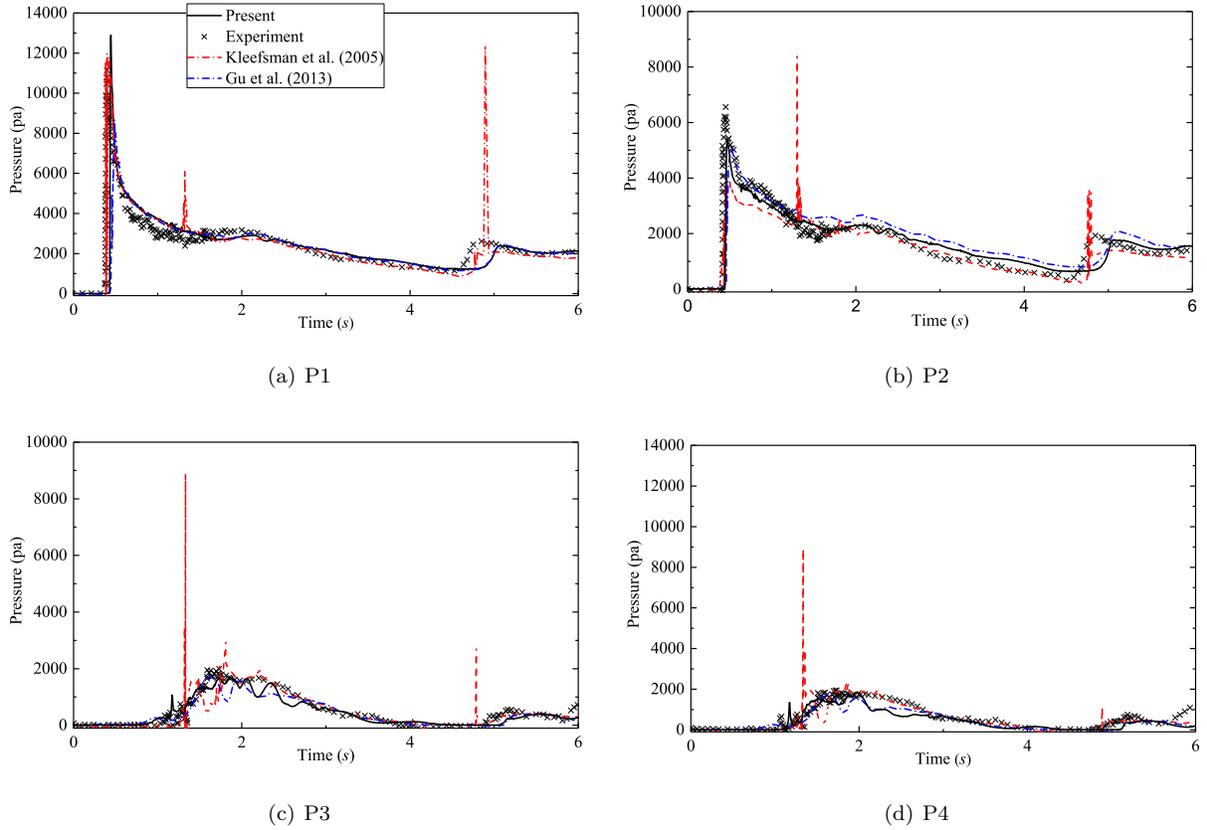


Figure 9: Comparison of time history of pressures at P1, P2, P3, and P4 with the experimental and other numerical results.

287 4.2. Free falling of a 3D wedge

288 In this section, a freely falling 3D wedge is investigated, which is more challenging than the dam break
 289 past a fixed body studied in the last section, as shown in Fig .11. Free falling of the 3D wedge was investigated
 290 experimentally in Yettou et al. (2006), where the position and velocity of the wedge were measured. Calderer
 291 et al. (2014) and Bihs and Kamath (2017) also worked on the same problem numerically using the immersed
 292 boundary method. The differences between the present and the previous immersed boundary methods mainly
 293 lie in the search of forcing points and the implementation of boundary condition.

294 This symmetric wedge with a 25 degree dead-rise angle weights 94kg, equivalent to a body with the
 295 density of 466 kg/m^3 . Initially, the wedge falls freely from the position 1.3m above the still water. For
 296 simplification, we set the velocity of the wedge as 5.0m/s ($\sqrt{2gs}$, where the distance $s = 1.3\text{m}$) at the initial
 297 instant of wedge penetrating the still water in the numerical simulation. The dimension of the wedge in the
 298 numerical simulations is taken as 1.2m in both the spanwise and longitudinal directions. Correspondingly,
 299 the length and width of the channel (fluid domain) are 4.0m and 2.0m. It ensures that there is a 0.4m gap
 300 between the wedge and each channel wall in the spanwise direction. The gap in the longitudinal direction is

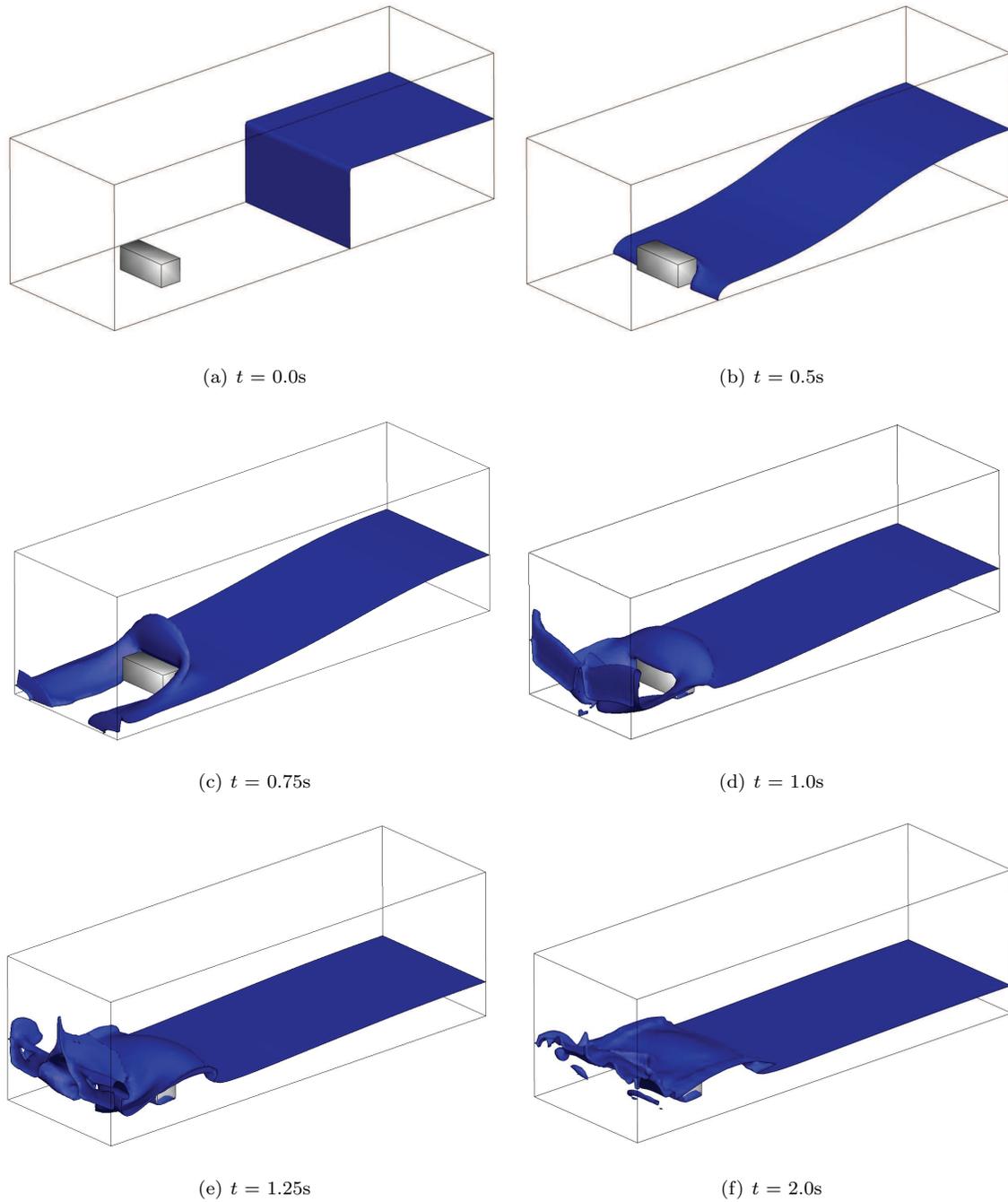


Figure 10: Water surface profiles at six different time instants for the dam break.

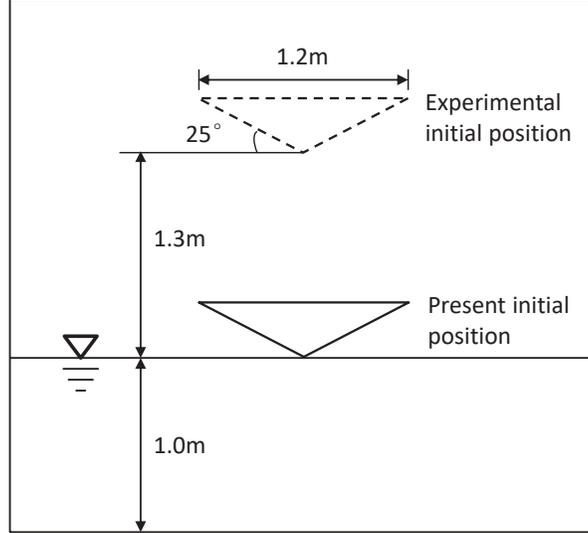


Figure 11: 2D sketch of water entry of the free falling wedge.

301 1.4m to each channel wall to avoid the influence of walls. The channel length in the present simulations is
 302 smaller compared to those in the experiment (Yettou et al., 2006) and other numerical simulations (Calderer
 303 et al., 2014, Bihs and Kamath, 2017). However, as the time duration is very short before we terminate the
 304 simulation, the wave reflection from the longitudinal walls cannot affect the central fluid area of interest.
 305 The uniform grid interval is 0.04m, and it is reduced to 0.01 around the structure in both the horizontal and
 306 vertical directions.

307 The wedge velocity at the initial stage is presented and compared with the experimental and the numerical
 308 results in Calderer et al. (2014), as shown in Fig. 12. In Calderer et al. (2014), the combination of a point-
 309 in-polyhedron algorithm for defining forcing points and the PPBC correction is adopted. Despite the PPBC
 310 improvement in Calderer et al. (2014) made to the standard method of Borazjani et al. (2008), the present
 311 numerical results of wedge velocity are still in a better agreement with the experimental data. At this stage,
 312 the present results and the experiment data show the evident inflection point around $t = 0.013s$, which is
 313 not well captured in the results of Calderer et al. (2014). Before $t = 0.013s$, the numerical results based
 314 on the proposed method agree much better with the experimental data than the other published results.
 315 In addition, Calderer et al. (2014) used much smaller grid spacings, which can further demonstrate the
 316 advantage of the present algebraic algorithm for the forcing point searching.

317 To extend the time duration in the comparison, both the wedge position and velocity in the whole process
 318 are compared with the experimental data and the numerical results in Calderer et al. (2014) and Bihs and
 319 Kamath (2017), as shown in Fig. 13. It is noted that the wedge position is recorded at the keel. In Bihs and
 320 Kamath (2017), a ray-tracing algorithm was adopted to locate forcing points. According to the comparison,
 321 it can be observed that the present numerical results almost override the results of Calderer et al. (2014),

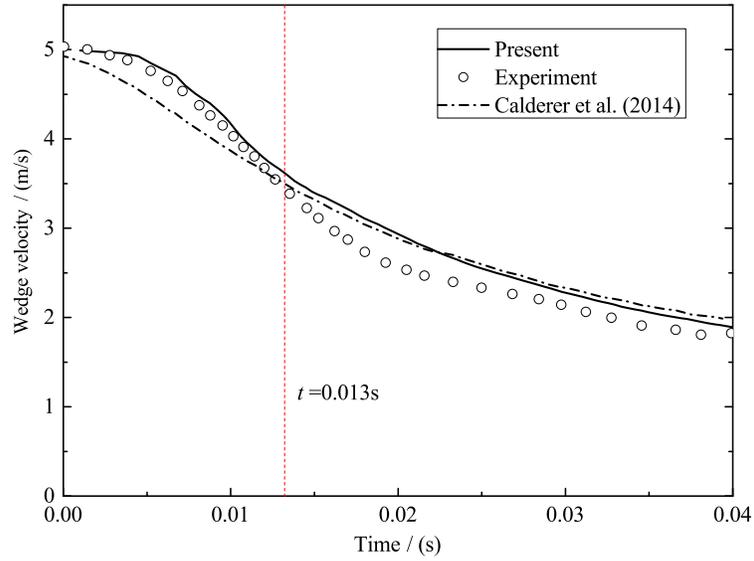


Figure 12: Comparison of the wedge velocity at the initial stage between the present, experimental, and other numerical results.

322 both of which agree well with the experimental data. However, one should note that much coarser grids
 323 are adopted in the present numerical simulations, compared to that in Calderer et al. (2014). Moreover, it
 324 is evident that the results for both the wedge position and velocity in Bihs and Kamath (2017) show more
 325 discrepancies with the experimental data, in the comparison with the present numerical simulations.

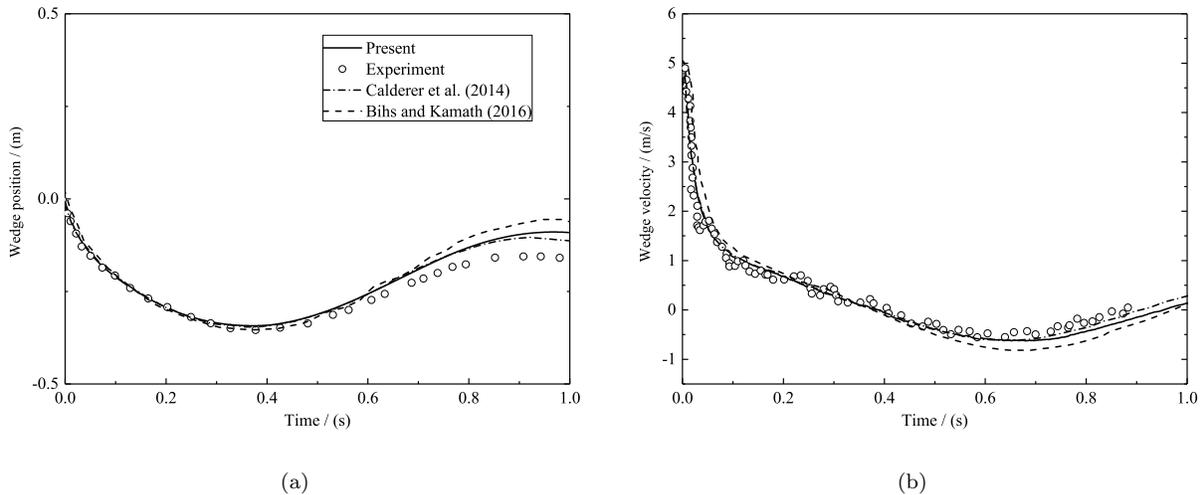


Figure 13: Comparison of the wedge position and velocity between the present, experimental, and other numerical results: (a) wedge position; (b) wedge velocity.

326 Fig.14 shows the time history of slamming force on the whole wedge. The peak value of the slamming

327 force is about 23525N, occurring at the time instant $t = 0.013s$. The largest slamming force corresponds
 328 to the largest acceleration according to the Newton's law. At this instant, the wedge velocity in Fig.13(b)
 329 decreases most sharply. In addition, the water surface profiles at several time instants are given in Fig. 15,
 330 where the position of the wedge and the water jet are shown. The wedge penetrates the water initially and
 331 floats upwards after $t = 0.5s$. A weak water jet around the wedge appears at $t = 0.7s$. The water surface
 332 looks similar to the "shipping-wave" with two wave crests. According to the comparison with the numerical
 333 results of the other two immersed boundary methods in Calderer et al. (2014) and Bihs and Kamath (2017),
 334 it can be seen that the surface profile is similar in all the studies when the 3D wedge is fully submerged.
 335 The surface profile from Bihs and Kamath (2017) looks more complicated than the other two, which seems
 336 unreal due to the errors in the predicted wedge velocity and position, as indicated in Fig. 13.

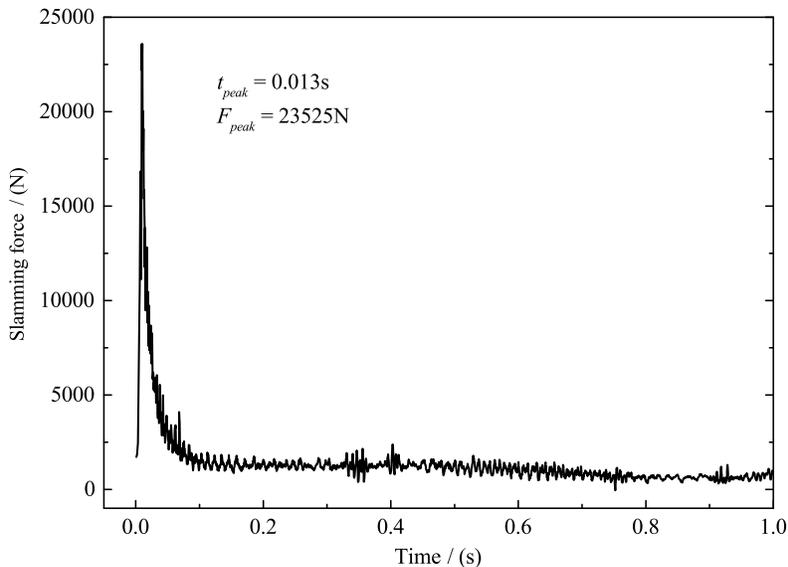


Figure 14: Time history of slamming force on the wedge.

337 Finally, numerical results of the vector field at the middle vertical section are shown in Fig. 16. It can
 338 be seen that around $t = 0.025s$, the velocity magnitude approaches the maximum at about 10m/s, which
 339 is consistent with the value reported in Bihs and Kamath (2017). The wedge penetrates the water sharply
 340 at $t = 0.2s$, as shown in Fig. 16(c). At the following time instants, the complicated water jet occurs and
 341 propagates to the walls. In addition, the vorticity appears and develops around the two tips of the wedge.

342 4.3. Decay of a bobber

343 To further demonstrate the capability of the developed 3D immersed boundary method in simulating a
 344 complex body geometry, a free decay bobber is considered in this section. In this test, a bobber undergoing
 345 oscillatory heavy motions is considered, which is controlled by its own weight m_f and a counterweight m_c . A
 346 pulley system is adopted to connect those two weighting systems, as shown in Fig. 17(a). According to the

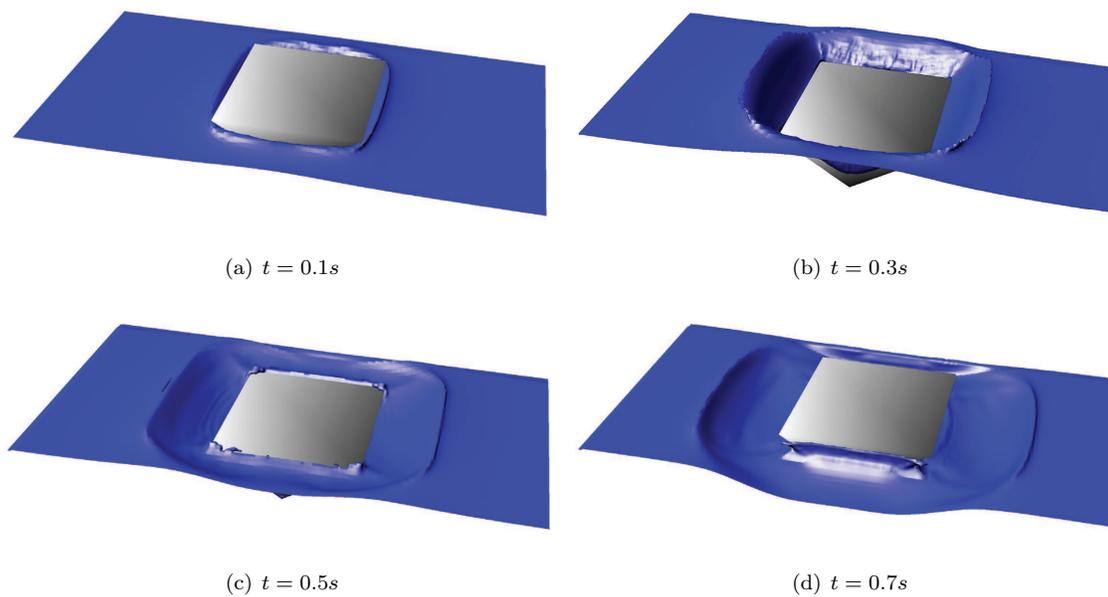


Figure 15: Water surface profiles at four different time instants for the free falling of the wedge.

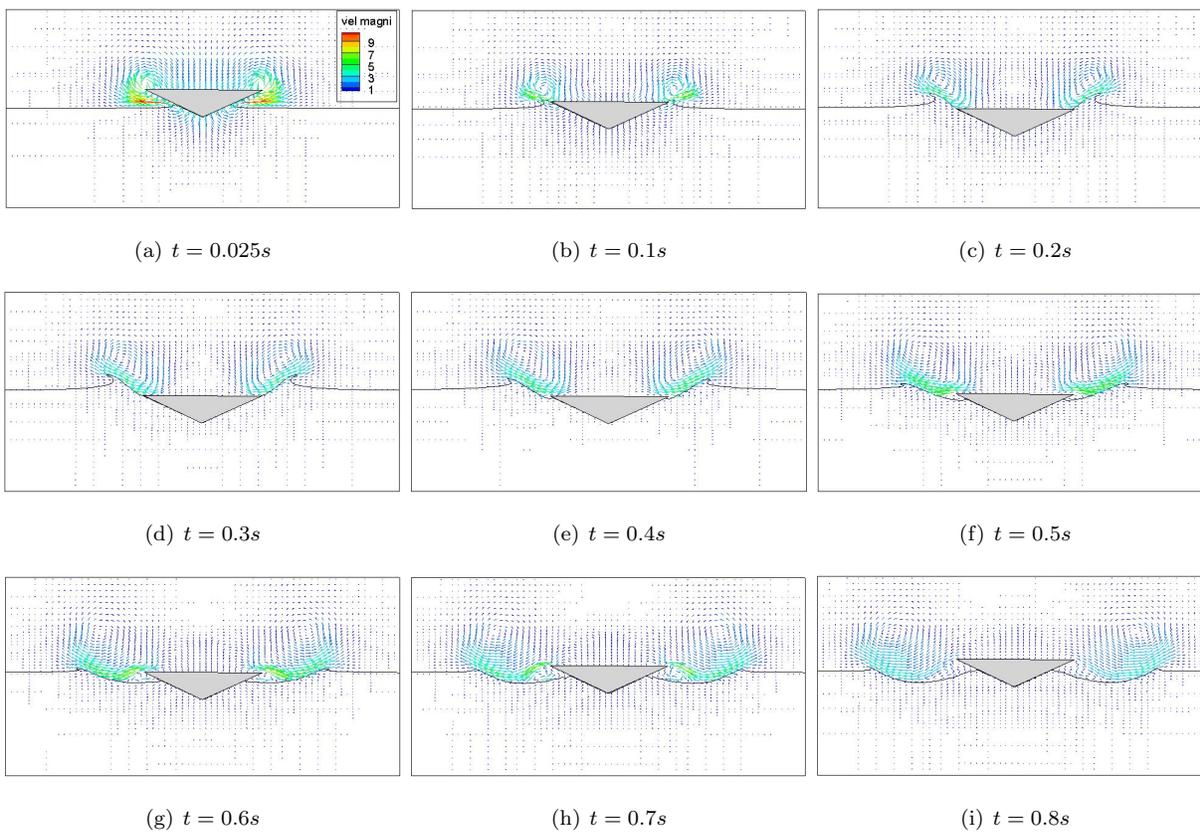


Figure 16: Vector field at different time instants for the free falling of the wedge.

347 Newton's second law, the motion of the system is described by the formula $(m_f + m_c)a_y = F_y + (m_f - m_c)g$,
 348 where a_y is the acceleration of the whole system in the vertical direction and F_y is the slamming force acting
 349 on the bobber. It is noted that the positive direction of force, displacement, acceleration is pointing upwards.

350 Based on Thomas et al. (2008) and Hu et al. (2013), the geometry of the bobber is the combination
 351 of a flat-bottomed cylinder of the radius 74mm, a 30° cone, and a cylinder with the radius 25mm. The
 352 flat-bottomed cylinder is unsharpened around the corner with the radius 33mm. The height of the bottom
 353 cylinder is 85mm, and the grids to describe the bobber surface are shown in Fig. 17(b). Initially, the bobber
 354 is placed 0.01m above the still water (0.5m water depth) before the free motion. The whole fluid domain is
 355 $2m \times 1m \times 2m$, with a grid of $0.01m \times 0.01m \times 0.01m$ around the bobber. To locate the forcing points,
 356 the bobber is discretized by 116 triangular surface elements.

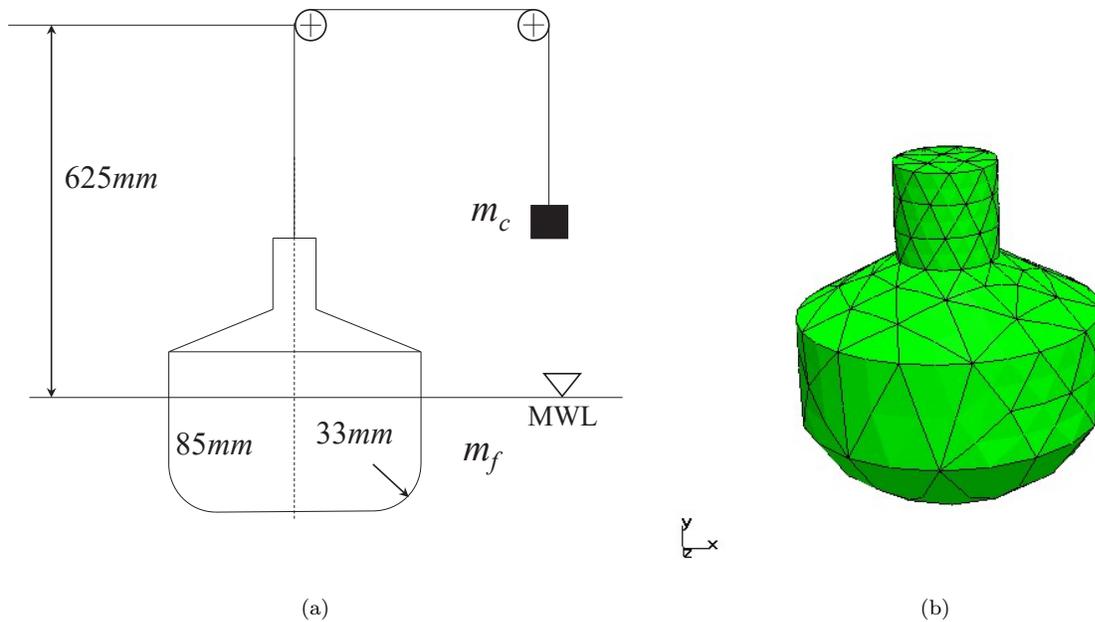


Figure 17: Sketch of bobber (a) and its surface mesh (b).

357 In the numerical simulation, m_f and m_c are set as 2.1kg and 1.2kg, respectively. Due to the difference in
 358 the weights of bobber and counterweight, the bobber could penetrate the water. In the water, the bobber is
 359 subjected to the hydrodynamic force, which leads to the oscillation of the bobber. In Calderer et al. (2014)
 360 and Bihs and Kamath (2017), the artificial damping was added into the motion equations for the heave
 361 motion by introducing the damping coefficient $C = 0.275$ in Eq. 5, in order to account for the friction of
 362 the experimental apparatus. However, as there is no friction of the experimental apparatus in the numerical
 363 simulation, the damping coefficient could be different. Thus, different values of damping coefficient are tested
 364 in the present study to evaluate the influence of damping, and the numerical results of vertical displacement
 365 of the bobber with different damping coefficients are compared in Fig. 18. The larger damping coefficient
 366 results in the smaller oscillation amplitude, and for the present case the damping coefficient $C = 1.5$ seems

367 to best fit with the experimental data.

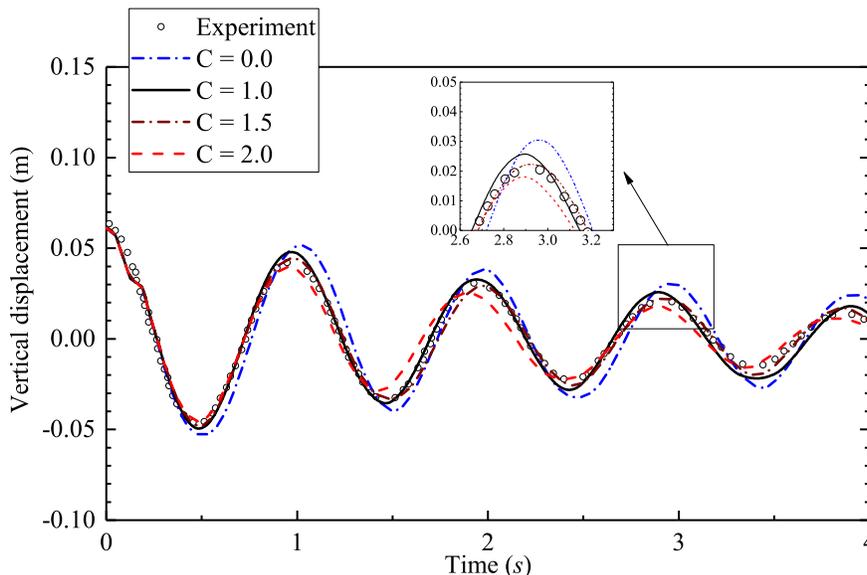


Figure 18: Numerical results of vertical displacement of the free decay bobber under different artificial damping coefficients.

368 The comparison of vertical displacement is shown in Fig. 19, in which the experimental data in Thomas
 369 et al. (2008), the numerical result from Hu et al. (2013) and the present results are included. In the first 4
 370 periods, it shows that the present numerical result obtains a obviously better agreement with the experiment
 371 than that in Hu et al. (2013). With further increase of time, both numerical results show some discrepancies
 372 with the experiment. However, the present results show no phase lag with the experimental data, while there
 373 is an evident delay in the numerical results from Hu et al. (2013). In addition, the numerical result from Hu
 374 et al. (2013) are severely over-predicted compared to the present numerical results, as the damping was not
 375 considered in Hu et al. (2013).

376 For a further comparison, the heave force and vertical velocity of the bobber are shown in Fig. 20(a) and
 377 Fig. 20(b). It can be seen that the agreement is not very satisfactory both for the phase and amplitude,
 378 especially after $t = 2.5s$. It seems that the present numerical results may be more reliable, as a better
 379 agreement has been achieved between the present numerical results and the experimental data for the
 380 displacement shown in Fig. 19. Finally, the snapshots of water surface closed to the bobber are presented in
 381 Fig. 21. The figure shows the oscillation of the bobber and the surface profile around the bobber at different
 382 time instants. The free surface is irregular, especially at $t = 0.9s$ when the discrete water volume appears.
 383 At $t = 0.5s$, there is a step-shape free surface profile due to the bobber geometry. From this case, the present
 384 new immersed boundary method is demonstrated to be robust and accurate even for a relatively complex
 385 body geometry.

386 Fig. 22 shows the pressure distribution at six different time instants at the central vertical section of the

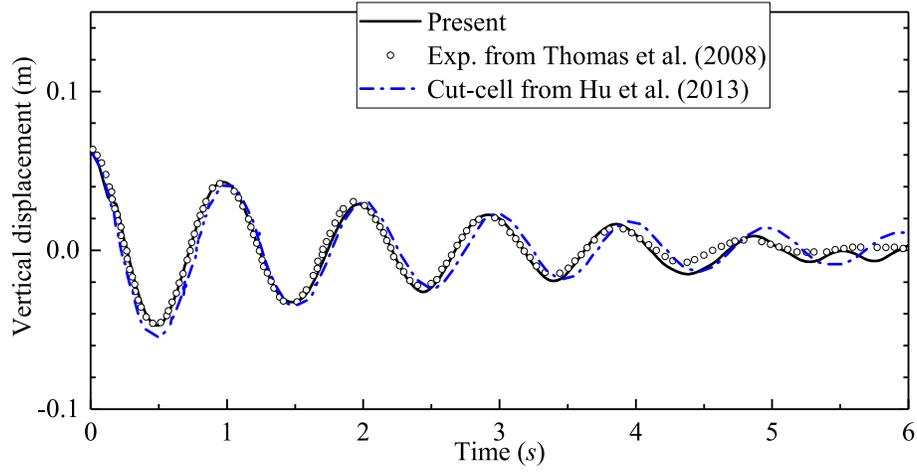
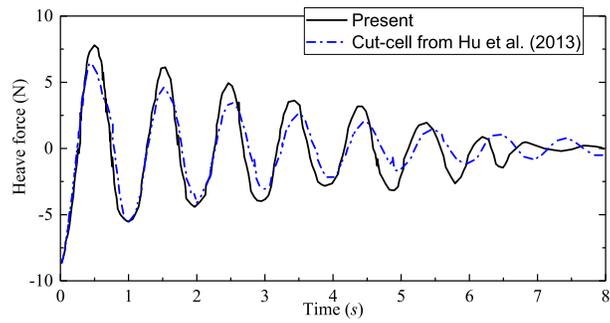
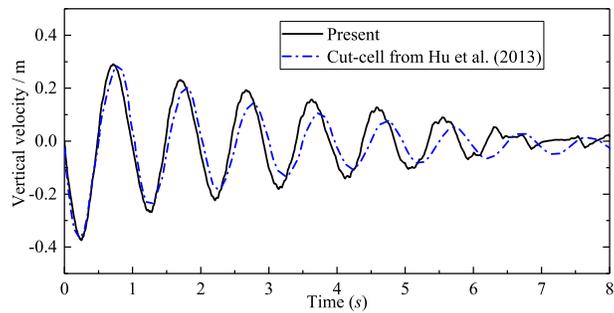


Figure 19: Comparison of time history of vertical displacement with the experimental data and other numerical results.



(a)



(b)

Figure 20: Time history of heave force (a) and vertical velocity (b) for the bobber undergoing heave motions.

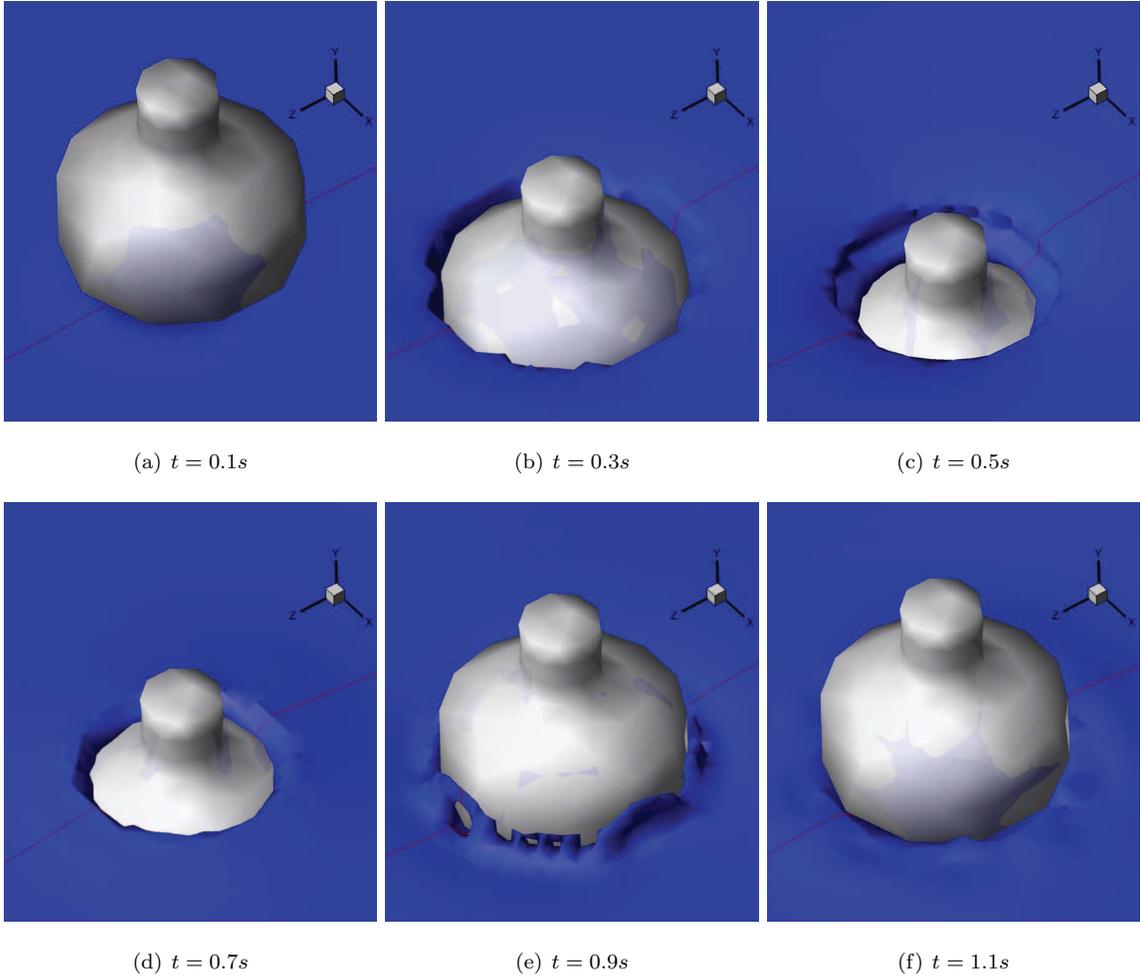


Figure 21: Water surface profiles at different time instants for the bobber decay.

387 domain. It can be seen that the hydrostatic pressure is dominant in most part of the domain. As the zero
 388 pressure is defined on the free surface, the free surface profile is represented by the zero-pressure contour
 389 approximately in the figure. Due to the geometry of the bobber, the splashing water jets are not very obvious
 390 during the water entry process. However, the water can run onto the shoulder of the bobber in Figs. 22(c)
 391 and 22(d). In Figs. 22(a), 22(e) and 22(f), the hydrostatic pressure field is not disturbed much by the bobber
 392 motion, while the dynamic pressure is visible around the bobber surface at $t = 0.3s$ in Fig. 22(b).

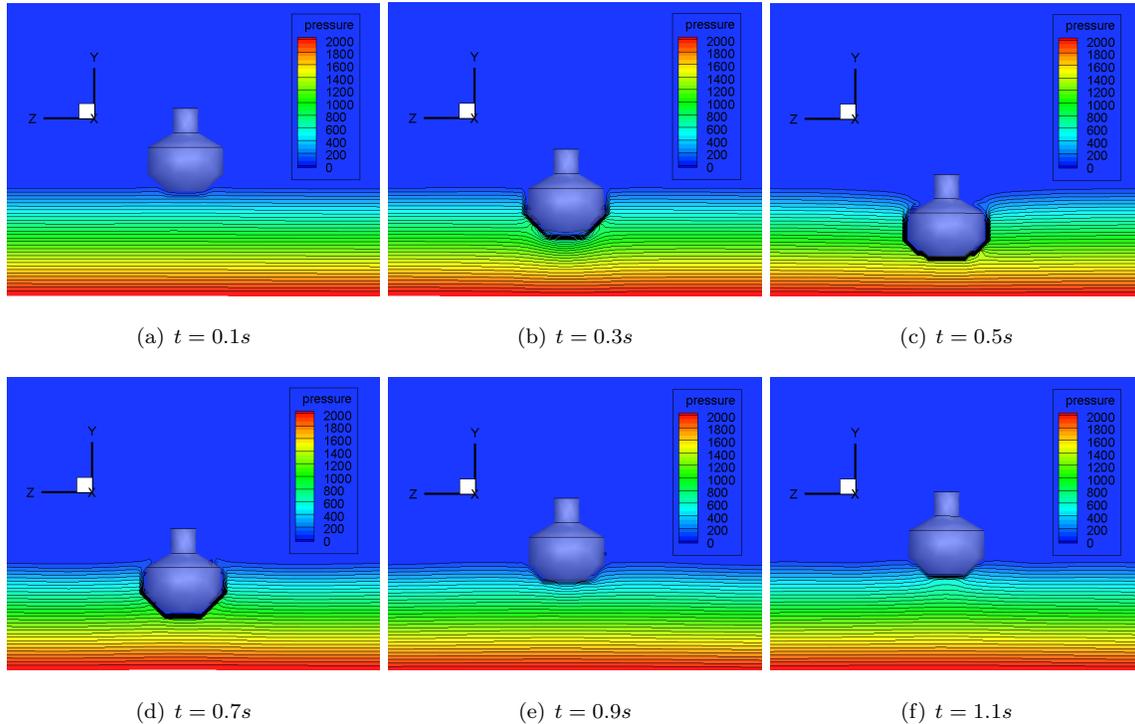


Figure 22: Pressure distribution at different time instants for the bobber decay.

393 5. Conclusions

394 In this paper, a new algebraic algorithm is incorporated to the forcing point searching scheme in the 3D
 395 immersed boundary method, which differs from the traditional geometrical approaches. The conventional
 396 way employs the concept of the geometrical method to locate the forcing points. However, the algebraic
 397 algorithm adopted to determine the triangle-triangle intersection is robust, easy to implement and efficient.
 398 For the purpose of validation of the developed 3D immersed boundary method, three test cases are carried
 399 out, including dam break over an obstacle, free falling of a 3D wedge and free decay of a bobber. In the dam
 400 break case, the present immersed boundary method converges fast and can capture the pressure magnitude
 401 better than other numerical methods. For the water entry of the wedge, the present results reveal that the
 402 overall wedge displacement and velocity obtained by the present immersed boundary method are superior to

403 other two immersed boundary methods. Lastly, after finding the most-fitted damping coefficient by matching
404 the experimental data, for the free decay bobber the developed immersed boundary method can provide the
405 better vertical displacement compared to other numerical method. All the numerical results suggest that
406 the present numerical model is robust and accurate for both fixed and moving bodies with irregular complex
407 geometries.

408 References

- 409 Archer, P. and Bai, W. (2015). A new non-overlapping concept to improve the hybrid particle level set
410 method in multi-phase fluid flows, *Journal of Computational Physics* **282**: 317–333.
- 411 Berthelsen, P. A. and Faltinsen, O. M. (2008). A local directional ghost cell approach for incompressible
412 viscous flow problems with irregular boundaries, *Journal of Computational Physics* **227**(9): 4354–4397.
- 413 Bihs, H. and Kamath, A. (2017). A combined level set/ghost cell immersed boundary representation for
414 floating body simulations, *International Journal for Numerical Methods in Fluids* **83**(12): 905–916.
- 415 Borazjani, I., Ge, L. and Sotiropoulos, F. (2008). Curvilinear immersed boundary method for simulating fluid
416 structure interaction with complex 3D rigid bodies, *Journal of Computational Physics* **227**(16): 7587–7620.
- 417 Calderer, A., Kang, S. and Sotiropoulos, F. (2014). Level set immersed boundary method for coupled
418 simulation of air/water interaction with complex floating structures, *Journal of Computational Physics*
419 **277**: 201–227.
- 420 Chen, H., Qian, L., Ma, Z. H., Bai, W., Li, Y., Causon, D. M. and Clive, C. G. (2019). Application of
421 an overset mesh based numerical wave tank for modelling realistic free-surface hydrodynamic problems,
422 *Ocean Engineering* **176**: 97–117.
- 423 Fadlun, E. A., Verzicco, R., Orlandi, P. and Mohd-Yusof, J. (2000). Combined immersed-boundary finite-
424 difference methods for three-dimensional complex flow simulations, *Journal of Computational Physics*
425 **61**: 35–60.
- 426 Gilmanov, A. and Sotiropoulos, F. (2005). A hybrid cartesian/immersed boundary method for simulating
427 flows with 3d, geometrically complex, moving bodies, *Journal of Computational Physics* **207**(2): 457–492.
- 428 Gu, H. B., Causon, D. M., Mingham, C. G. and Qian, L. (2013). Development of a free surface flow solver
429 for the simulation of wave/body interactions, *European Journal of Mechanics - B/Fluids* **38**: 1–17.
- 430 Guigue, P. and Devillers, O. (2003). Fast and robust triangle-triangle overlap test using orientation predi-
431 cates, *Journal of Graphics Tools* **8**(1): 25–32.
- 432 Held, M. (1997). ERIT - a collection of efficient and reliable intersection tests, *Journal of Graphics Tools*
433 **2**(24): 25–44.

- 434 Hu, Z. Z., Causon, D. M., Mingham, C. G. and Qian, L. (2013). A cartesian cut cell free surface cap-
435 turing method for 3d water impact problems, *International Journal for Numerical Methods in Fluids*
436 **71**(10): 1238–1259.
- 437 Kim, D. and Choi, H. (2006). Immersed boundary method for flow around an arbitrarily moving body,
438 *Journal of Computational Physics* **212**(2): 662–680.
- 439 Kim, J., Kim, D. and Choi, H. (2001). An immersed boundary finite-volume method for simulations of flow
440 in complex geometries, *Journal of Computational Physics* **171**: 132–150.
- 441 Kleefsman, K. M. T., Fekken, G., Veldman, A. E. P., Iwanowski, B. and Buchner, B. (2005). A Volume-of-
442 Fluid based simulation method for wave impact problems, *Journal of Computational Physics* **206**(1): 363–
443 393.
- 444 Lee, J. and You, D. (2013). An implicit ghost-cell immersed boundary method for simulations of moving
445 body problems with control of spurious force oscillations, *Journal of Computational Physics* **233**: 295–314.
- 446 Ma, Z. H., Qian, L., Ferrer, P. M., Causon, D. M., Clive, C. G. and Bai, W. (2018). An overset mesh based
447 multiphase flow solver for water entry problems, *Computers and Fluids* **172**: 689–705.
- 448 Mittal, R., Dong, H., Bozkurttas, M., Najjar, F. M., Vargas, A. and von Loebbecke, A. (2008). A versatile
449 sharp interface immersed boundary method for incompressible flows with complex boundaries, *Journal of*
450 *Computational Physics* **227**(10): 4825–4852.
- 451 Mohd-Yusof, J. (1997). Combined immersed boundary/B-spline method for simulations of flows in complex
452 geometries, *Technical report*, Center Annual Research Briefs, NASA Ames/Stanford University.
- 453 Moller, T. (1997). A fast triangle-triangle intersection test, *Journal of Graphics Tools* **2**(2): 25–44.
- 454 Nicolaou, L., Jung, S. Y. and Zaki, T. A. (2015). A robust direct-forcing immersed boundary method with
455 enhanced stability for moving body problems in curvilinear coordinates, *Computers and Fluids* **119**: 101–
456 114.
- 457 Peskin, C. S. (1972). Flow patterns around heart valves: a numerical method, *Journal of Computational*
458 *Physics* **10**: 252 – 271.
- 459 Roman, F., Napoli, E., Milici, B. and Armenio, V. (2009). An improved immersed boundary method for
460 curvilinear grids, *Computers and Fluids* **38**(8): 1510–1527.
- 461 Schöberl, J. (1997). NETGEN: An advancing front 2D/3D-mesh generator based on abstract rules, *Com-*
462 *puting and Visualization in Science* **1**(1): 41–52.
- 463 Seo, J. H. and Mittal, R. (2011). A sharp-interface immersed boundary method with improved mass conser-
464 vation and reduced spurious pressure oscillations, *Journal of Computational Physics* **230**(19): 7347–7363.

- 465 Thomas, S., Weller, S. and Stallard, T. (2008). Float response within an array: numerical and experimental
466 comparison, *Proceeding of 2nd International Conference on Ocean Energy*, Brest, France.
- 467 Tropp, O., Tal, A. and Shimshoni, I. (2006). A fast triangle to triangle intersection test for collision detection,
468 *Computer Animation and Virtual Worlds* **17**(5): 527–535.
- 469 Wu, C.-S. (2019). A modified volume-of-fluid/hybrid cartesian immersed boundary method for simulating
470 free-surface undulation over moving topographies, *Computers and Fluids* **179**: 91–111.
- 471 Wu, C.-S. and Young, D.-L. (2014). Simulations of free-surface flows with an embedded object by a coupling
472 partitioned approach, *Computers and Fluids* **89**: 66–77.
- 473 Wu, C.-S., Young, D.-L. and C.L, C. (2013). Simulation of wave-structure interaction by hybrid carte-
474 sian/immersed boundary and arbitrary lagrangian-eulerian finite-element method, *Journal of Computa-
475 tional Physics* **254**: 155–183.
- 476 Yan, B., Bai, W. and Quek, S. T. (2018). An improved immersed boundary method with new forcing point
477 searching scheme for simulation of bodies in free surface flows, *Communications in Computational Physics*
478 **24**(3): 830–859.
- 479 Yan, S. and Ma, Q. W. (2007). Numerical simulation of fully nonlinear interaction between steep waves and
480 2D floating bodies using the QALE-FEM method, *Journal of Computational Physics* **221**: 666 – 692.
- 481 Yang, J., Preidikman, S. and Balaras, E. (2008). A strongly coupled embedded-boundary method for fluid-
482 structure interactions of elastically mounted rigid bodies, *Journal of Fluids and Structures* **24**: 167 –
483 182.
- 484 Yettou, E. M., Desrochers, A. and Champoux, Y. (2006). Experimental study on the water impact of a
485 symmetrical wedge, *Fluid Dynamics Research* **38**(1): 47–66.
- 486 Zhang, Y., Zou, Q., Greaves, D., Reeve, D., Hunt-Raby, A., Graham, D., Phil, J. and Lv, X. (2010). A
487 level set immersed boundary method for water entry and exit, *Communications in Computational Physics*
488 **8**(2): 265–288.