



# Modified dynamic boundary conditions (mDBC) for general-purpose smoothed particle hydrodynamics (SPH): application to tank sloshing, dam break and fish pass problems

A. English<sup>1</sup> · J. M. Domínguez<sup>2</sup> · R. Vacondio<sup>3</sup> · A. J. C. Crespo<sup>2</sup> · P. K. Stansby<sup>1</sup> · S. J. Lind<sup>1</sup> · L. Chiapponi<sup>3</sup> · M. Gómez-Gesteira<sup>2</sup>

Received: 10 November 2020 / Revised: 28 February 2021 / Accepted: 11 March 2021  
© The Author(s) 2021

## Abstract

Dynamic boundary conditions (DBC) for solid surfaces are standard in the weakly compressible smoothed particle hydrodynamics (SPH) code DualSPHysics. A stationary solid is simply represented by fixed particles with pressure from the equation of state. Boundaries are easy to set up and computations are relatively stable and efficient, providing robust numerical simulation for complex geometries. However, a small unphysical gap between the fluid and solid boundaries can form, decreasing the accuracy of pressures measured on the boundary. A method is presented where the density of solid particles is obtained from ghost positions within the fluid domain by linear extrapolation. With this approach, the gap between fluid and boundary is reduced and pressures in still water converge to hydrostatic, including the case of a bed with a sharp corner. The violent free-surface cases of a sloshing tank and dam break impact on an obstacle show pressures measured directly on solid surfaces in close agreement with experiments. The complex 3-D flow in a fish pass, with baffles to divert the flow, is simulated showing close agreement with measured water levels with weirs open and gates closed, but less close with gates open and weirs closed. This indicates the method is suitable for rapidly varying free-surface flows, but development for complex turbulent flows is necessary. The code with the modified dynamic boundary condition (mDBC) is available in DualSPHysics to run on CPUs or GPUs.

**Keywords** SPH · Boundary conditions · DualSPHysics · Free-surface flows

## 1 Introduction

Smoothed particle hydrodynamics (SPH) is a meshless Lagrangian particle numerical method particularly adept at modelling complex, highly deforming interfacial and free-surface flows. In recent years, SPH numerical schemes have been successfully applied to a number of different phenomena, including sloshing tanks [1], sediment scour [2], debris flows [3], flow around bodies [4] and submarines

[5], landslides and flooding [6, 7], fish pass flows [8], wave action on breakwaters [9] with encouraging results. The SPH method is based on an integral approximation incorporating a kernel function (typically of compact support and characterised by a smoothing length,  $h$ ). The meshless nature of SPH and the issue of kernel truncation near boundaries can create difficulties in enforcing solid boundary conditions, and, accordingly, boundary conditions have been highlighted as one of the Grand Challenges of SPH [10]. A recent review addressing accuracy in SPH is provided by Lind et al. [11], and a review of applications in coastal and ocean engineering is available in Gotoh and Khayyar [12]

Many different approaches have been proposed in the literature to enforce solid boundary conditions, and they can be grouped into three main types: the first approach is the method of repulsive forces ([13, 14]), which enables the discretization of 2-D and 3-D irregular geometries. However, non-physical forces are added to prevent particle penetration and kernel truncation is not addressed and so the accuracy

✉ A. English  
aaron.english@manchester.ac.uk

<sup>1</sup> Department of Mechanical, Aerospace, and Civil Engineering, University of Manchester, Manchester, UK

<sup>2</sup> Environmental Physics Laboratory, Universidade de Vigo, Ourense, Spain

<sup>3</sup> Department of Civil Environmental Engineering, University of Parma, Parma, Italy

of SPH spatial interpolation near the walls is remarkably reduced. A second possible way to discretize the boundaries is the so-called semi-analytical formulation ([15–17]) where additional terms in the conservation equations are considered to compensate for the kernel truncation. At continuous level, there are no terms of SPH spatial interpolation, but when the particle discretization is introduced, surface integrals have to be approximated and this remains a challenge for complex 3-D geometries and/or multi-phase flows. The third class of boundary methods are based on fictitious particles to fill the space beyond the boundary interface to mimic the presence of a wall ([18–21]). A hybrid of the first and third classes has also been proposed [22]. The method herein proposed belongs to the third class and may be considered an extension of the dynamic boundary condition (DBC) currently used in the online version of DualSPHysics [23].

DualSPHysics [23] is an open source SPH code for simulating free-surface flows that is able to run on both CPUs and GPUs (graphics processing units) [24]. Using DBC, complex geometries can be created such as those in [3, 8] and [9]. This tackles one of the aspects of the Grand Challenge on boundary condition in SPH [10]. However, with DBC unphysical gaps form between the boundary and fluid particles. The work of [9] estimated the size of the gap to be of the order of the kernel smoothing length, but a realistic surface pressure may be obtained at this distance from the surface.

The main purpose of the present work is thus to avoid some of the limitations of DBC, while maintaining the capability for discretizing complex geometries, without affecting the efficiency of the GPU implementation. To reduce the gap and increase the accuracy of the pressure measured at the boundary particles, we follow the approach of [19] for updating physical quantities of the boundary particles by means of a first-order consistent SPH interpolation evaluated at ghost nodes, located inside the fluid domain. First-order consistency in the boundary is obtained by introducing the SPH corrected interpolation proposed in [25]. In this way, the flow properties may be extrapolated into the boundary and a fluid continuum is presented. Finally, the use of an adequate number of layers of boundary particles prevents inconsistencies due to kernel truncation effects for fluid particles located near the boundaries, as discussed in the early work of Vacondio et al. [26]. The corrected SPH interpolation herein adopted does not require an additional particle sweep, and thus the computational overheads of the proposed correction are low. The modified boundary conditions are named mDBC (modified DBC).

This paper is organised as follows: in Sect. 2 the SPH method used in this work is outlined; in Sect. 3 the DBC methodology is described and the advantages and issues with the method highlighted along with the new mDBC method. In Sect. 4 results for a number of 2D and 3D test

cases are presented comparing between the existing DBC and the new mDBC, finally in Sect. 5 conclusions of the work are provided along with suggestions for future work.

## 2 SPH methodology

The open source SPH solver DualSPHysics is described as follows. The conservation of mass is defined in compressible form, including the density diffusion term

$$\frac{d\rho_i}{dt} = \sum_j m_j \mathbf{v}_{ij} \cdot \nabla W_{ij} + \delta h c_0 \sum_j \frac{m_j}{\rho_j} \Psi_{ij} \cdot \nabla W_{ij} \quad (1)$$

where  $\rho_i$  is particle density,  $m_i$  its mass,  $c_0$  is the speed of sound in the fluid,  $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$  is the velocity difference between particles  $i$  and  $j$ , and  $\nabla W_{ij}$  is the kernel gradient.  $\delta = 0.1$  is generally applied. The SPH kernel used in this work is the quintic Wendland kernel [27] given by

$$W_{ij} = \alpha_D \left(1 - \frac{r_{ij}}{2h}\right)^4 \left(1 + \frac{2r_{ij}}{h}\right) \quad \text{for } 0 \leq r_{ij} \leq 2h \quad (2)$$

where  $r_{ij}$  is the distance between particles,  $h = \kappa dp$  is the smoothing length ( $\kappa = 1.3$  or  $2$  in this work) and the normalisation term  $\alpha_D = 7/4\pi h^2$  in 2D and  $\alpha_D = 21/16\pi h^3$  in 3D. Two options for the density diffusion function  $\Psi_{ij}$  are used, the first of Molteni and Colagrossi [28] is given by

$$\Psi_{ij}^{\text{MOL}} = 2(\rho_j - \rho_i) \frac{r_{ij}}{|r_{ij}|^2} \quad (3)$$

The second option is that of Fourtakas et al. [21] and is given by

$$\Psi_{ij}^{\text{FOUR}} = 2\left(\rho_{ji}^T - \rho_{ij}^H\right) \frac{r_{ij}}{|r_{ij}|^2} \quad (4)$$

where  $\rho^T$  and  $\rho^H$  are the total hydrostatic densities. The conservation of momentum is given by

$$\frac{d\mathbf{v}_i}{dt} = - \sum_j m_j \left( \frac{P_j + P_i}{\rho_i \rho_j} + \Pi_{ij} \right) \nabla W_{ij} + \mathbf{g} \quad (5)$$

where  $\mathbf{g}$  is the acceleration due to gravity,  $P_i$  is the particle pressure and  $\Pi_{ij}$  is the artificial viscosity given by

$$\Pi_{ij} = \begin{cases} -\alpha \frac{c_0}{\rho_i + \rho_j} \frac{h \mathbf{v}_{ij} \cdot \mathbf{r}_{ij}}{|r_{ij}|^2 + 0.01 h^2} & \text{for } \mathbf{v}_{ij} \cdot \mathbf{r}_{ij} < 0 \\ 0 & \text{for } \mathbf{v}_{ij} \cdot \mathbf{r}_{ij} \geq 0 \end{cases} \quad (6)$$

A standard value of  $\alpha = 0.01$  will be used unless otherwise stated.

Finally, the pressure is given by the Tait equation of state

$$P = \frac{c_0^2 \rho_0}{\gamma} \left[ \left( \frac{\rho}{\rho_0} \right)^\gamma - 1 \right] \quad (7)$$

where  $\rho_0 = 1000 \text{ kg/m}^3$  is the reference density of water and  $\gamma = 7$ .

Time stepping is by the Symplectic predictor–corrector option with time step given a CFL (Courant–Friedrichs–Lewy) value of 0.2. The speed of sound is usually given a value of  $10\times$  times the wave speed,  $\sqrt{gH}$ , where  $g$  is the modulus of the gravitational acceleration and  $H$  is initial water depth.

Note that the SPH formulation adopted in the present work is variationally consistent and therefore conserves mass, linear and angular momentum. In particular, the conservative form of the momentum equation (Eq. (5)) is used here and it is used both for fluid–fluid and fluid–boundary interactions. Mass is also conserved, as in the proposed scheme each particle carries its own mass during the simulation. This allows the continuity equation to be written in non-conservative form (Eq. (1)) without breaking conservation of relevant physical quantities and allows density to be assigned to boundary particles through a consistency correction (Eq. (12)) which improves remarkably the accuracy of the pressure field.

### 3 mDBC formulation

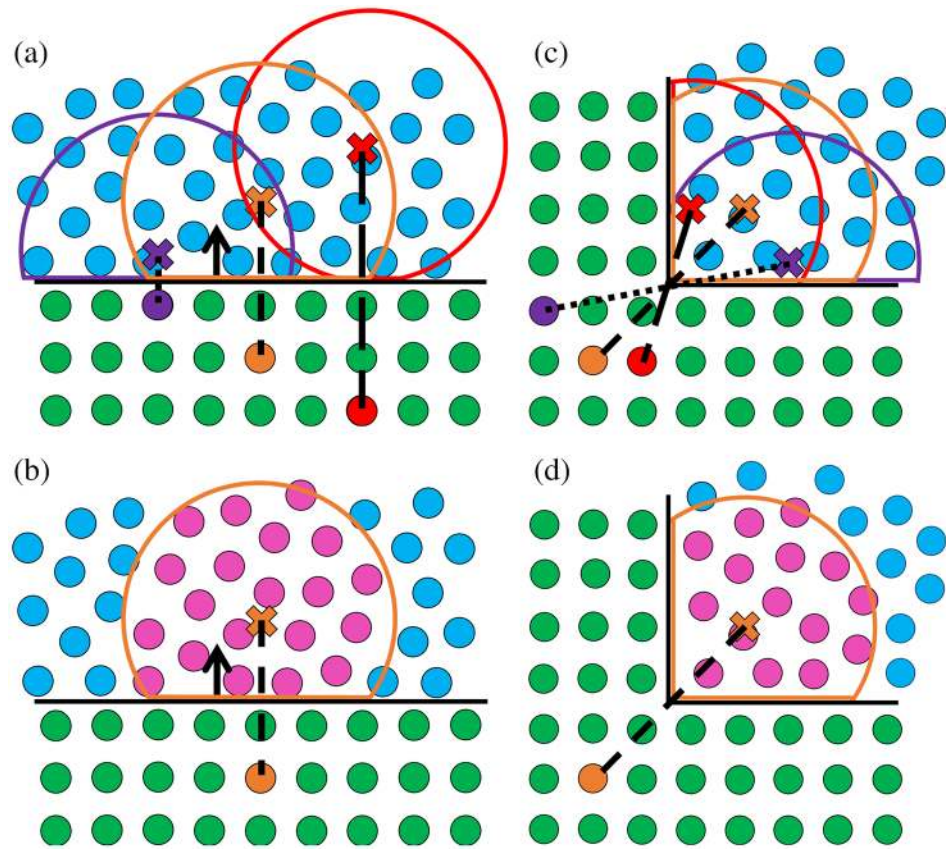
Dynamic boundary conditions were first introduced in [29] and further studied in [18]. The boundaries consist of a set of particles that satisfy the same continuity equation as the fluid particles. To approximate the no-slip condition at solid boundaries, the velocities of the boundary particles are set to zero. The repulsion mechanism generated by the dynamic boundary particles works in the following way: the incoming fluid particle increases the density locally according to the continuity equation (Eq. 1), which results in an increase in pressure following equation of state (Eq. 7) and in an increase in the pressure term  $((P_j + P_i)/\rho_i \rho_j)$ . In consequence, this increase in the pressure term in momentum equation (Eq. 5) will lead to an increase in the acceleration magnitude ( $dv/dt$ ) for the incoming fluid particle, which defines the repulsion force. The study of Domínguez et al. [30] showed how it is evident that as a repulsive force, this boundary keeps the particles at a certain distance from the boundary, and this “gap” was found to be of the order of the smoothing length ( $h$ ). Therefore, this approach presents a drawback in that the evolution of density and pressure of the fixed boundary particles leads to unphysical values at the surface, with unphysically large boundary layers in the flow. The boundary pressure has to be output at a distance of a smoothing

length from the actual surface to be representative and the dynamic boundary particles need to be initially created considering that “gap” in order to generate the repulsion force at the desired boundary limit. On the other hand, an advantage of these boundary particles is their computational simplicity, since density, and therefore pressure, can be calculated inside the same loops as fluid particles with a considerable saving of computational time. These conditions are able to represent complex shaped geometries. Validations with dam-break flows and wave flumes have been published and DBC has been successfully applied to coastal engineering problems, discretizing complex 3D geometries without the need for implementing complex mirroring techniques or semi-analytical wall boundary conditions. A good example is the work of Zhang et al. [9] where DualSPHysics was used to reproduce a laboratory test where a porous breakwater structure made of cubes was analysed. The SPH results were obtained for 52 numerical wave probes, and good agreement was obtained for this complex problem.

We propose here a new method to palliate the problems described for DBC that is named mDBC (modified dynamic boundary conditions). The boundary particles of mDBC are arranged in the same way as the boundary particles in the original DBC, with an additional boundary interface created between the fluid and the boundary particles. The boundary interface is located half a particle spacing ( $dp/2$ ) from the layer of boundary particles closest to the fluid. For each boundary particle, a ghost node is projected into the fluid across the boundary interface in a procedure similar to Marrone et al. [19]. For a flat surface, the ghost node is mirrored across the boundary interface along the direction of the boundary normal pointing into the fluid (Fig. 1a) and fluid properties are found at this ghost node through a first-order consistent SPH spatial interpolation over the surrounding fluid particles only (Fig. 1b). For a boundary particle located in a corner, the boundary normal is ill defined with more than one option. However, the boundary interfaces of each solid boundary meet to form an interface corner, and the ghost node is mirrored through the point of this corner into the fluid region (Fig. 1c). Again, fluid properties are found at the ghost node through a first-order consistent SPH spatial interpolation of the surrounding fluid (Fig. 1d).

The boundary particles receive fluid properties using the values calculated at the ghost node and an extrapolation method similar to the one used for open boundaries in [31]. For the density of the boundary particle  $\rho_b$  the ghost density  $\rho_g$  and its gradient  $[\partial_x \rho_g; \partial_y \rho_g; \partial_z \rho_g]$  are computed at the ghost node using the first-order consistent SPH interpolation proposed by Liu and Liu [25], which requires solving the following linear system for each ghost node:

**Fig. 1** Mirroring of ghost nodes (crosses) and the kernel radius around the ghost nodes for boundary particles in a flat surface (a) and a corner (c). Fluid particles (pink) included in the kernel sum around ghost nodes for boundary particles in a flat surface (b) and a corner (d)



$$\mathbf{A}_g \cdot \begin{bmatrix} \rho_g \\ \partial_x \rho_g \\ \partial_y \rho_g \\ \partial_z \rho_g \end{bmatrix} = \mathbf{b}_g \quad (8)$$

where:

$$\mathbf{A}_g = \begin{bmatrix} \sum_j W_{gj} V_j & \sum_j (x_j - x_g) W_{ij} V_j & \sum_j (y_j - y_g) W_{gj} V_j & \sum_j (z_j - z_g) W_{gj} V_j \\ \sum_j \partial_x W_{gj} V_j & \sum_j (x_j - x_g) \partial_x W_{ij} V_j & \sum_j (y_j - y_g) \partial_x W_{gj} V_j & \sum_j (z_j - z_g) \partial_x W_{gj} V_j \\ \sum_j \partial_y W_{gj} V_j & \sum_j (x_j - x_g) \partial_y W_{ij} V_j & \sum_j (y_j - y_g) \partial_y W_{gj} V_j & \sum_j (z_j - z_g) \partial_y W_{gj} V_j \\ \sum_j \partial_z W_{gj} V_j & \sum_j (x_j - x_g) \partial_z W_{ij} V_j & \sum_j (y_j - y_g) \partial_z W_{gj} V_j & \sum_j (z_j - z_g) \partial_z W_{gj} V_j \end{bmatrix} \quad (9)$$

$$\mathbf{b}_g^T = \left[ \sum_j W_{gj} m_j \quad \sum_j \partial_x W_{gj} m_j \quad \sum_j \partial_y W_{gj} m_j \quad \sum_j \partial_z W_{gj} m_j \right] \quad (10)$$

where the volume  $V_j$  is computed as  $V_j = m_j / \rho_j$ .

To evaluate the matrix  $\mathbf{A}_g$  and the vector  $\mathbf{b}_g$  at each ghost node  $g$ , a new particle interaction loop needs to be set up. The new interaction requires the inversion of the matrix  $\mathbf{A}_g$ ; however, this requires limited additional computational effort when compared to the original DBC formulation. Note also that the matrix  $\mathbf{A}_g$  is diagonally dominant and in most

cases remains well-conditioned (with condition numbers of  $O(10)$  even for disordered particle distributions) provided there is near complete support. Due to this property, the adopted formulation to compute  $\rho_g$  and  $[\partial_x \rho_g; \partial_y \rho_g; \partial_z \rho_g]$  is still suitable for real engineering applications with complex surfaces. If ghost node neighbours decrease to very low

numbers (e.g. less than 3 or 4 fluid particles), the  $\mathbf{A}_g$  matrix does become ill-conditioned and the matrix is not inverted. The density at the ghost node is then found through a Shepard filter evaluated at the ghost node according to:

$$\rho_g = \frac{\sum_j \rho_j W_{gj} V_j}{\sum_j W_{gj} V_j} \quad (11)$$

Once the density and density gradient are computed at the ghost node, then the density of the boundary particle  $\rho_b$  is

obtained by means of a linear extrapolation with the values found using the above relations through:

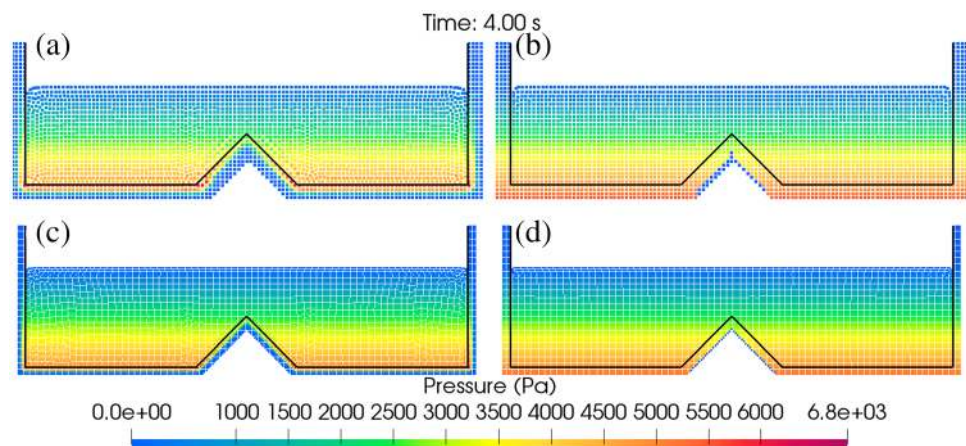
$$\rho_b = \rho_g + (\mathbf{r}_b - \mathbf{r}_g) \cdot [\partial_x \rho_g; \partial_y \rho_g; \partial_z \rho_g] \quad (12)$$

where  $\mathbf{r}_b$  and  $\mathbf{r}_g$  are the position of the boundary particle and associated ghost node, respectively. In this way, the boundary density is presented as part of a fluid continuum and pressure from the equation of state gives smoother and more physical pressure fields, avoiding the non-physical gap between the boundary and the fluid observed when using DBC. If the matrix is ill-conditioned, the boundary particle is given the Shephard filtered ghost node density found in Eq. (11). The boundary particles have zero velocity as before. As the boundary velocity is set to zero for all boundary particles; by definition,  $\mathbf{u} \cdot \mathbf{n} = 0$  (as  $\mathbf{u} = 0$ ) is guaranteed on the boundary. It is noted, however, that this approach is at best first order for velocity and velocity gradients on and near the boundary are in general not as well approximated as they are in other velocity boundary condition approaches (see [19], for example). Neumann boundary conditions on the pressure, however, are approximated at higher order following from the formulation above, and this is sufficient to significantly improve results over the standard DBC.

## 4 Results

This section will show the capabilities of mDBC and the improvements comparing with original DBC. Initially 2D simulations are investigated including still water with a triangular corner (a wedge) and a sloshing tank, and the 3D cases of a dam break impacting a cuboid obstacle, and flow through a fish pass. In each test case the performance of the boundary conditions will be assessed.

**Fig. 2** Pressure in a 2D still water tank: last instant of the simulation for: **a** DBC  $dp=0.02$  m; **b** mDBC  $dp=0.02$  m; **c** DBC  $dp=0.01$  m, and **d** mDBC  $dp=0.01$  m



### 4.1 Still water on bed with a wedge (sharp corner)

A 2D still water tank with dimensions of  $2.4 \text{ m} \times 1.2 \text{ m}$  encloses a trigonal wedge in the bottom centre of the tank with a height of  $0.24 \text{ m}$ . The initial water height is  $H = 0.5 \text{ m}$ . Results are analysed during  $4 \text{ s}$  of physical time. The ratio of smoothing length  $h$  to initial particle spacing  $dp$ ,  $h/dp = 2$ . Two simulations with different resolutions were executed ( $dp=0.02 \text{ m}$  and  $dp=0.01 \text{ m}$ ).

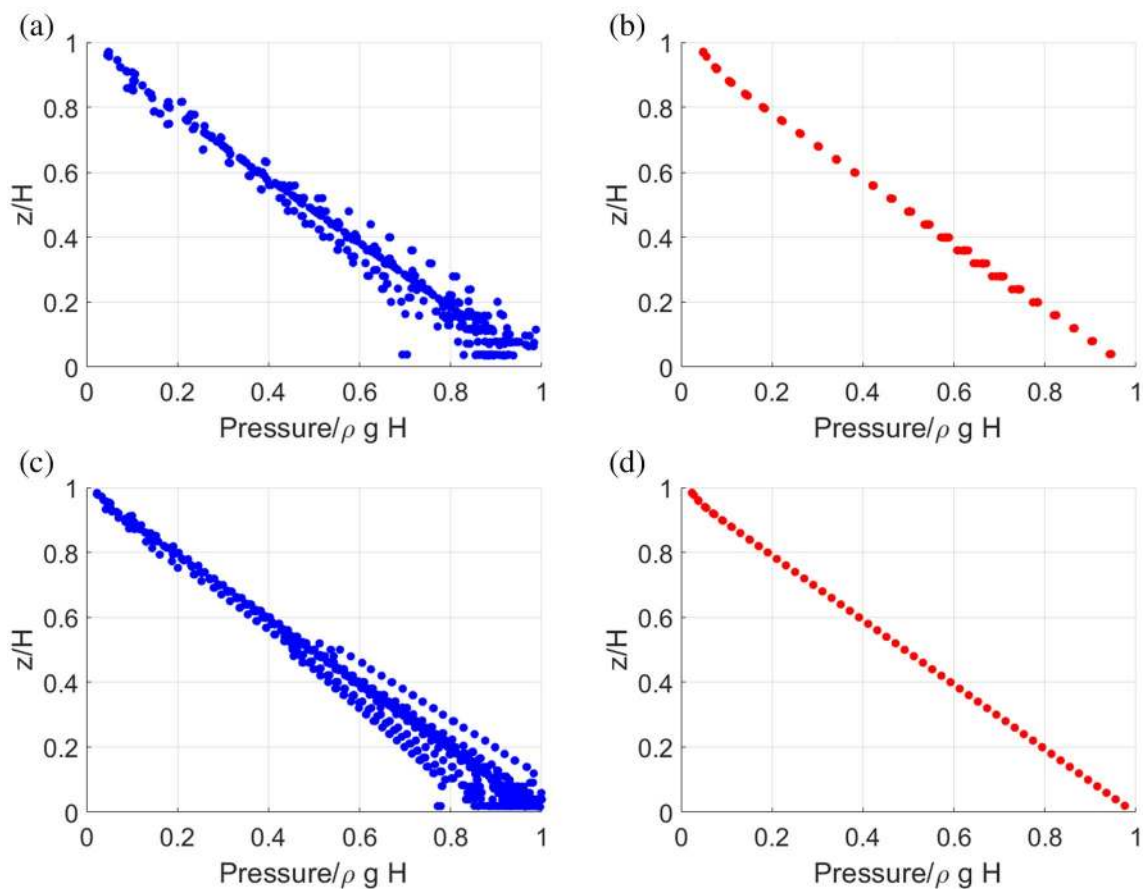
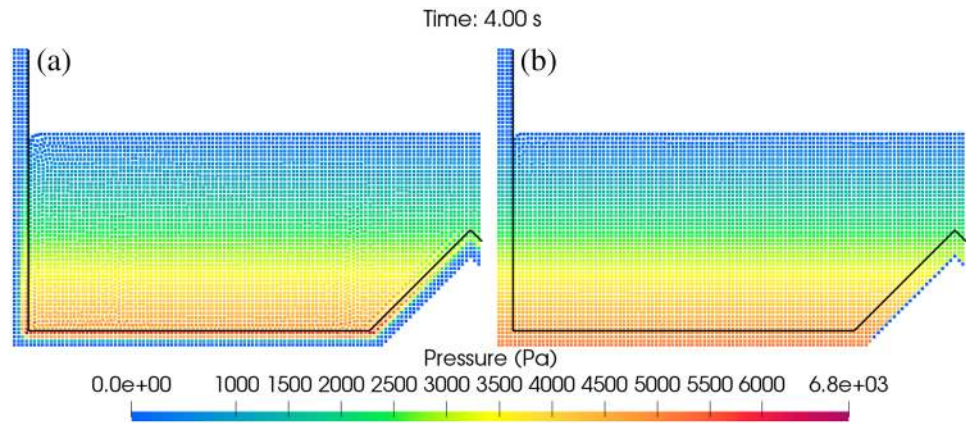
The final instant ( $t=4 \text{ s}$ ) is depicted in Fig. 2 for the two different resolutions and using DBC (left) and mDBC (right). The first row corresponds to  $dp=0.02 \text{ m}$  and the second row to  $dp=0.01 \text{ m}$ . Good results and improvements can be observed using mDBC with smoother and more physical pressure values now being obtained for the boundary particles, not only in the flat surface but also in the corners. Figure 3 shows more detail at the corners for the case with  $dp=0.01 \text{ m}$ , where this improvement becomes more apparent.

The vertical distribution of pressure is plotted to show the accuracy of the hydrostatic pressure distribution. Values of depth ( $z/H$ ) against pressure ( $p/\rho gH$ ) are plotted in Fig. 4 (at  $4 \text{ s}$ ) for each fluid particle. For both resolutions, much improved hydrostatic pressure behaviour is obtained using mDBC, accurate down to the solid surface. In addition, the kinetic energy of the particles is measured. The time series for the summation of kinetic energy of all the fluid particles is shown in Fig. 5 (note a logarithmic scale is used). It is clear that mDBC generates much smaller particle movement than DBC. Runs up to  $20 \text{ s}$  showed negligible noise in the pressure, but a small amount of noise became apparent around  $200 \text{ s}$ .

### 4.2 Sloshing tank (moving boundaries)

The second 2D test case reproduces an experiment with moving boundaries. This is the SPHERIC Benchmark Test

**Fig. 3** Last instant of the still water simulation: **a** DBC  $dp=0.01$  m, and **b** mDBC  $dp=0.01$  m (close up of the pressure field in the corners)

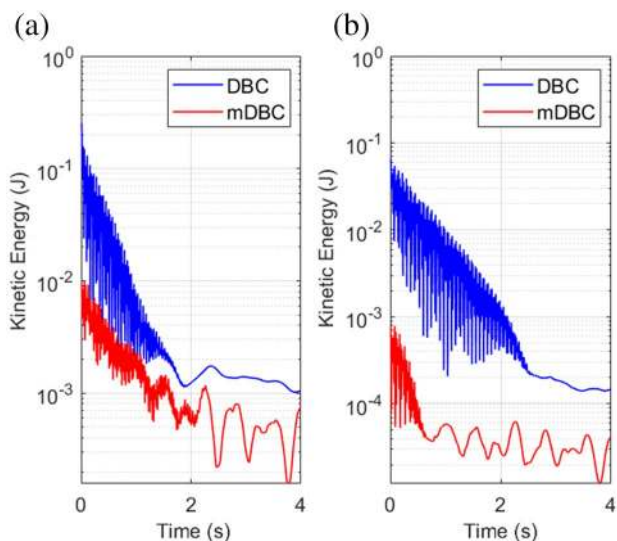


**Fig. 4** Particle pressure values versus depth at last instant of the still water simulation for: **a** DBC  $dp=0.02$  m; **b** mDBC  $dp=0.02$  m; **c** DBC  $dp=0.01$  m, and **d** mDBC  $dp=0.01$  m

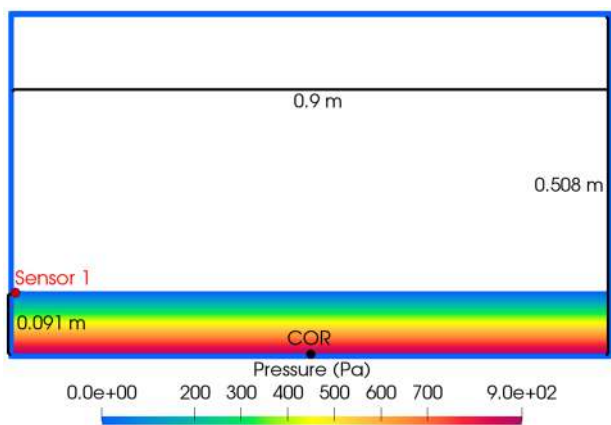
Case 10 (<https://spheric-sph.org/tests/test-10>), consisting of a sloshing tank of  $0.9 \text{ m} \times 0.508 \text{ m}$  with an initial water level  $H = 0.093 \text{ m}$  (Fig. 6). The numerical pressures are obtained using DBC and mDBC and compared with the experimental values detected at Sensor 1 (Fig. 6). Two simulations

with different resolutions are executed ( $dp = 0.004 \text{ m}$  and  $dp = 0.002 \text{ m}$ ), again with  $h/dp = 2$ .

The instant of the simulation at time  $t = 2.47 \text{ s}$  seconds is shown in Fig. 7, where the first impact of the fluid with left wall (Sensor 1) has just occurred. The colour of the particles corresponds to their pressure values.



**Fig. 5** Time series of fluid particles kinetic energy during the still water simulation for initial partial spacings **a**  $dp=0.02$  m and **b**  $dp=0.01$  m



**Fig. 6** Initial setup of the sloshing tank including location of pressure sensor 1 and the centre of rotation denoted as COR

Figure 8 shows in more detail the pressure field of the particles with DBC (left) and mDBC (right) with  $dp = 0.002$  m. Two improvements can be observed using mDBC: (i) values of pressure for the boundary particles in the walls using mDBC are less noisy than the ones shown for DBC and (ii) the unphysical gap between the fluid and the boundary is negligible when mDBC is used. This is an important improvement since the computation of pressure values at Sensor 1 (Fig. 6) with DBC provides representative results only if the numerical pressure gauge is moved to take into account the size of the gap between the fluid and the boundary. The work of [32] estimated the size of the gap to be the smoothing length ( $h$ ). There is a small high pressure region seen towards the top of the boundary of Fig. 8(b), which

is a spurious but ineffectual pressure carried from a previous time step. This may occur when there are no nearby fluid particles: the correction matrix (Eq. (9)) does not meet the criteria to be solved due to its poor conditioning and the Shepard filter (Eq. 11) cannot be used as there are no fluid particles near a ghost node. This spurious pressure is not a concern in practice as it only results when there are no nearby fluid particles (within the support length of the ghost node) and so can have no effect on the fluid dynamics. As soon as there is fluid near the boundary, the correction matrix is able to solve and so provides correct densities (and pressures) in the boundary.

The experimental pressure (black line) is compared with numerical results in Fig. 9. The blue line, in the first row, corresponds to numerical pressure computed using DBC with pressure gauge at the original location; however the results are erroneous due to the large gap (in the order of  $h$ ) without fluid particles. The second row shows the numerical pressure (green line) using DBC with pressure gauge moved  $+h$  into the fluid, much improving agreement with experimental data. However the use of mDBC, avoiding the gap and creating a realistic boundary layer, allows the computation of pressure at the exact location of the gauge obtaining very good results; the red line in the bottom panel of Fig. 9 shows again a good agreement with the experimental data. Note that negative pressures are observed in the experimental data and the mDBC numerical data around the time of the most violent impacts, coinciding with highly transient pressure shocks rebounding after high velocity water impact. The fact these transients are captured provides a further demonstration of the benefit of the mDBC approach over DBC. While persistent negative pressures can be an issue in SPH (leading potentially to the tensile instability), no evidence of this instability was observed here as the negative pressure event is so short-lived (existing for only  $O(1)$  time steps).

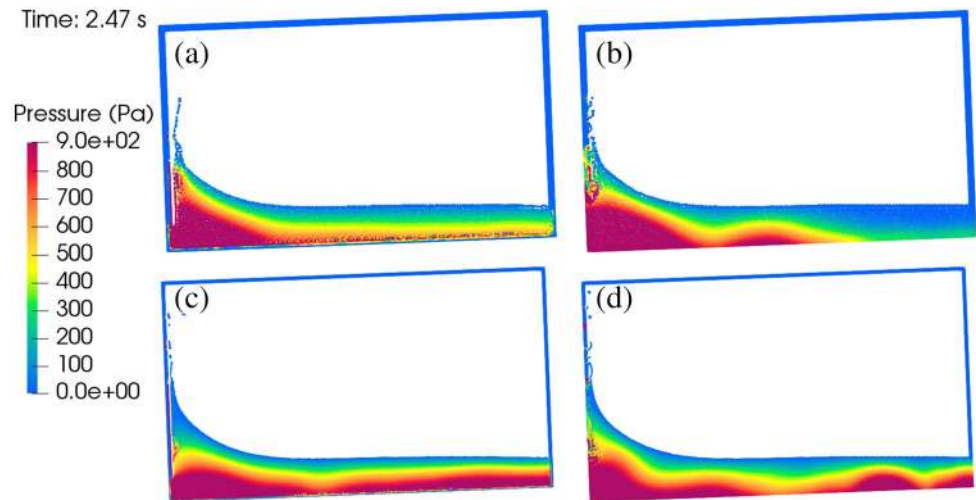
### 4.3 3D Dam break over cuboid obstacle

This is the SPHERIC Benchmark Test Case 2 (<https://spheric-sph.org/tests/test-2>) shown in Fig. 10; surface elevation was measured at positions  $H1$ ,  $H2$ ,  $H3$  and  $H4$  and pressures on the centreline plane of the cuboid obstacle surface [33]. Particle spacings  $dp = 0.01$  m and  $0.02$  m were applied giving a particle number close to 1 million and 170,000, respectively. In this case  $h/dp = 1.3$ .

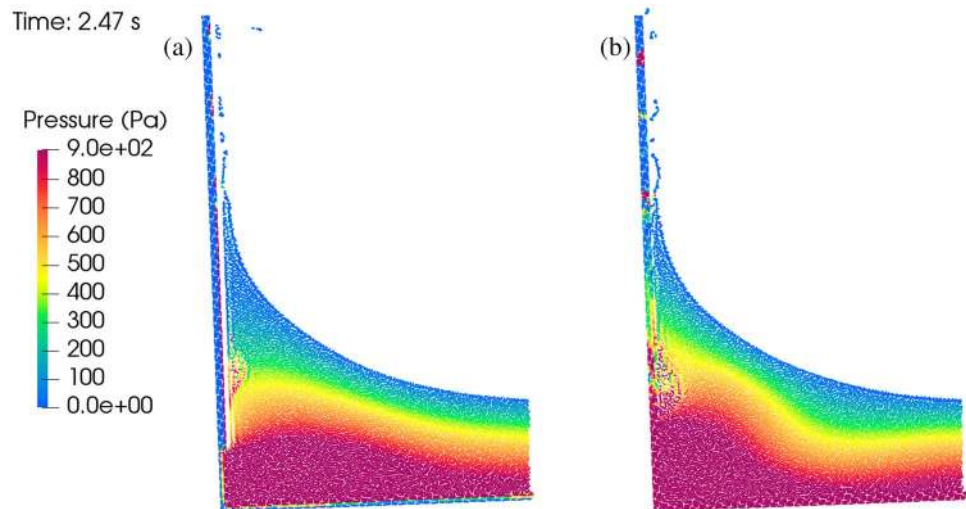
Results of surface elevation at  $H4$ ,  $H3$ ,  $H2$  are shown in Fig. 11 to be in close agreement with experiment, closer than DBC which shows a spurious value for  $H2$  for small time.

Figure 12 shows the particle distribution at  $t = 0.8$  s with jet height close to maximum, and the spurious DBC value at  $H2$  is due to jet formation too far upstream, in front of the obstacle.

**Fig. 7** Sloshing tank simulation at time  $t=2.47$  s, with: **a** DBC  $dp=0.004$  m; **b** mDBC  $dp=0.004$  m; **c** DBC  $dp=0.002$  m, and **d** mDBC  $dp=0.002$  m



**Fig. 8** Sloshing tank simulation at the time  $t=2.47$  s, close up of the pressure field in the corner and the lateral wall for DBC boundary (left) and mDBC boundaries (right)



Pressures are shown at  $P2$  (front face) and  $P5$  (top corner of upper surface) in Fig. 13. For  $P2$  agreement of mDBC with experiment is again close, generally closer than DBC. The peak pressures for  $P2$  are in close agreement with experiment. Note again that DBC requires measurement at  $+h$  from the surface. For  $P5$  mDBC with finer resolution is generally close to experiment as is DBC. Now the coarser mDBC shows some spurious fluctuations not present with the finer resolution.

This test case demonstrates how SPH with mDBC predicts highly transient free surface flow and pressures quite accurately.

#### 4.4 Fish pass

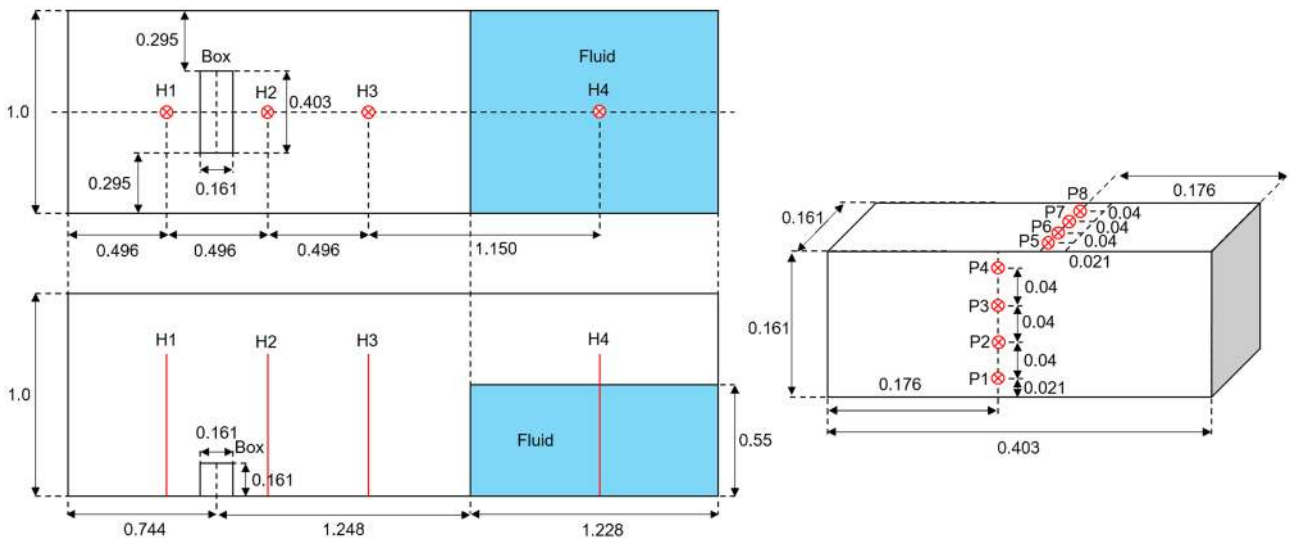
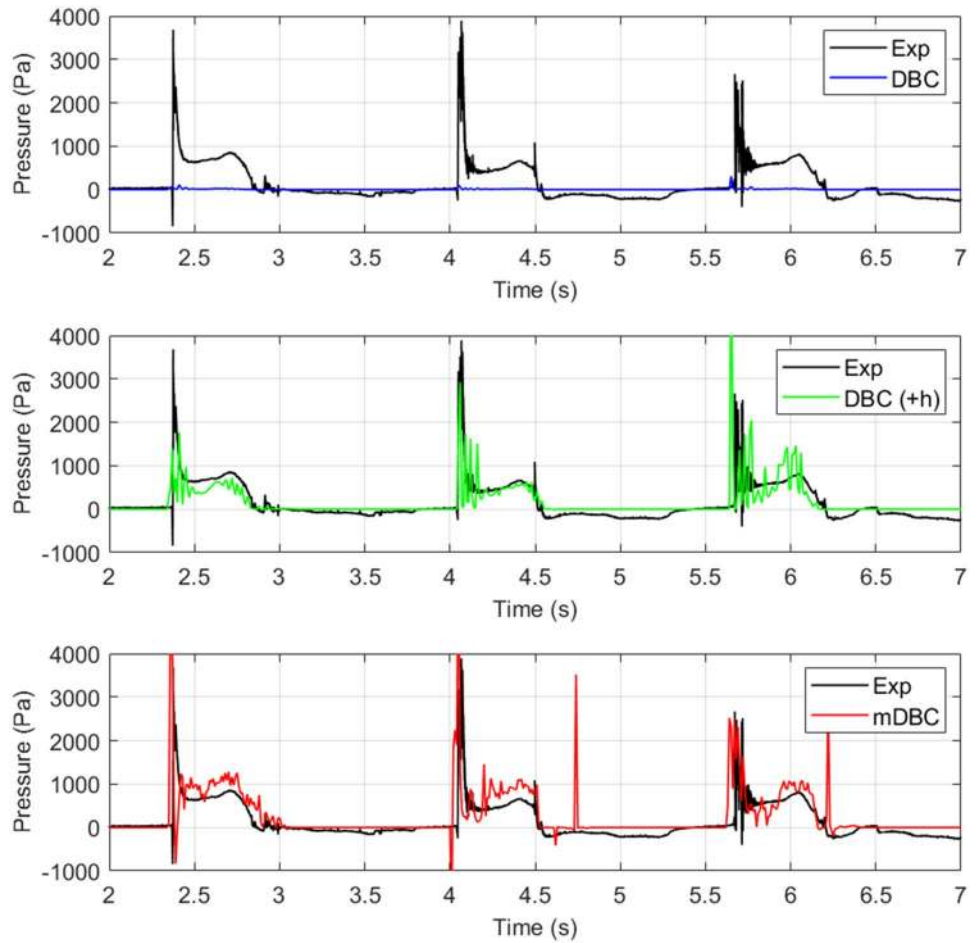
The complex 3D test case of a fish pass, as shown in Fig. 14, is now simulated. A fish pass is a structure that facilitates the natural migration of some species of fish on or around artificial and natural barriers (such as dams,

locks and waterfalls). Technical fish passes include: pool and weir fish passes, which is the type treated in the present work; vertical slot fish passes, and Denil fish passes. The principle of a pool and weir fish pass is to divide a channel by installing cross-walls, in order to form a succession of stepped pools from the headwater to the tailwater. The discharge usually passes through openings in the cross-walls and fishes migrate from one pool to the next through the submerged orifices (or gates) or through the notches (or weirs). Previous examples of fish pass flows being simulated using DualSPHysics with DBC include the work of Novak et al. [8] for a vertical slot type fish pass.

The fish pass considered here comprises a long channel inclined at an angle of  $4.5^\circ$  to the horizontal with vertical cross baffles restricting the flow, dividing the pass into three pools. Each of the baffles has a gate in one of the bottom corners and a weir in the opposite top corner, the orientation of the gates and weirs alternates between pairs



**Fig. 9** Comparison of experimental and numerical ( $d_p = 0.002$  m) pressures measured at probe location  $P1$  with: (top) DBC and numerical probe in the correct location; (middle) DBC and numerical probe placed a distance of  $h$  from the wall, (bottom) mDBC with numerical probe in the correct location



**Fig. 10** The geometric configuration of initial conditions for the dam break and obstacle (left) and pressure measurement points (right)

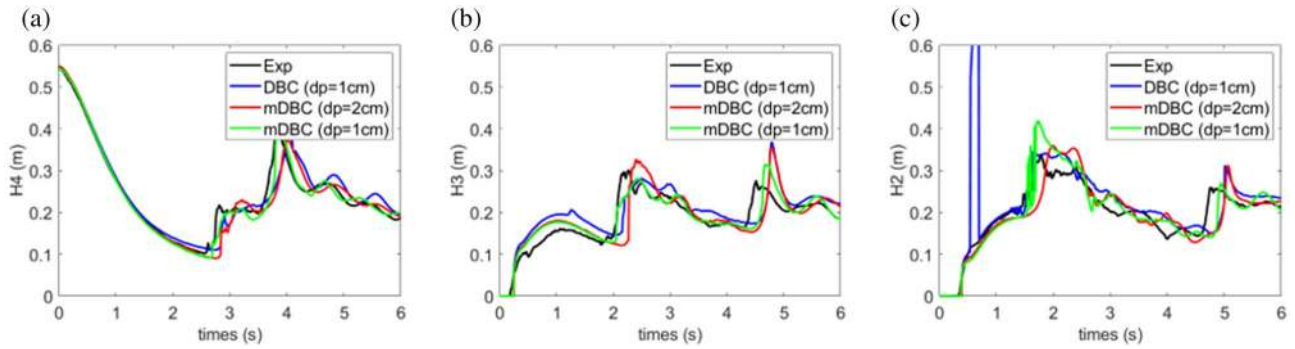


Fig. 11 3D Dam Break over cuboid obstacle: surface elevations at measured at points  $H4$  (left),  $H3$  (middle) and  $H2$  (right)

Fig. 12 Dam break at  $t=0.8$  s from DBC and mDBC showing extent of jet formation

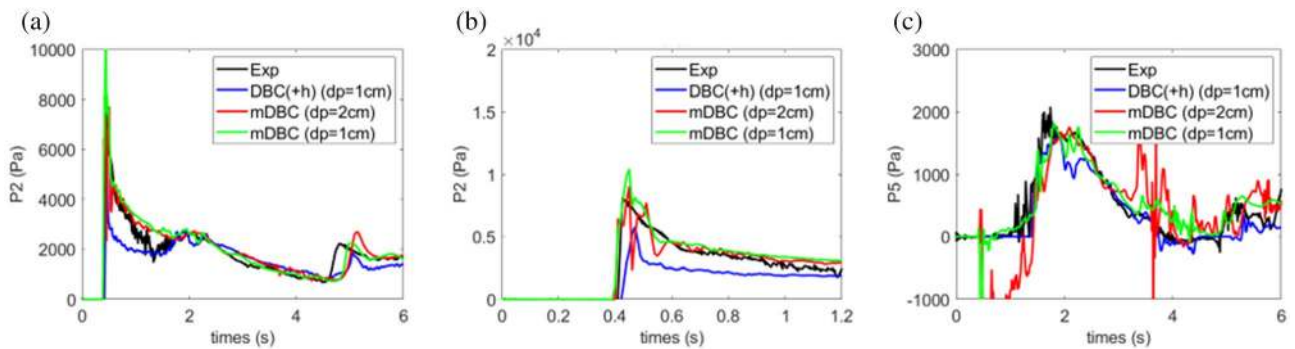
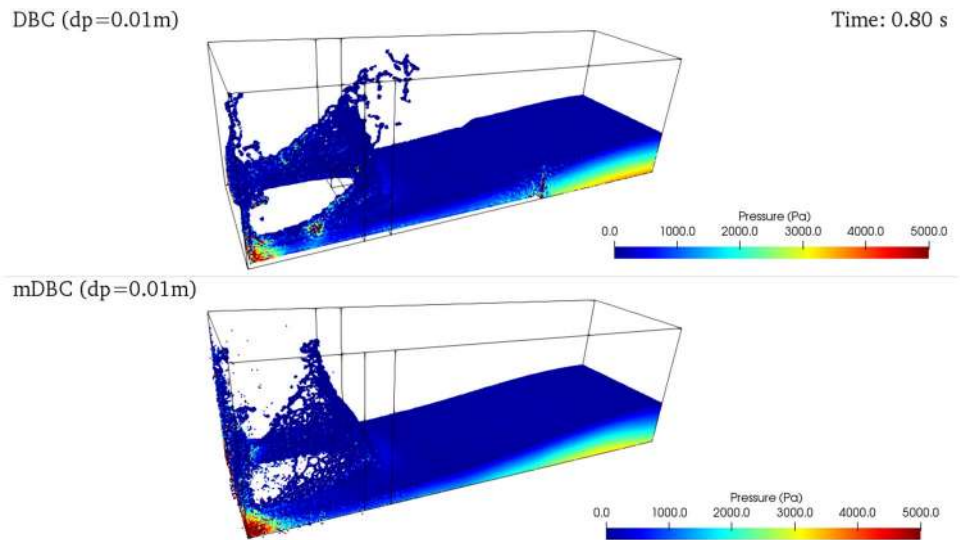


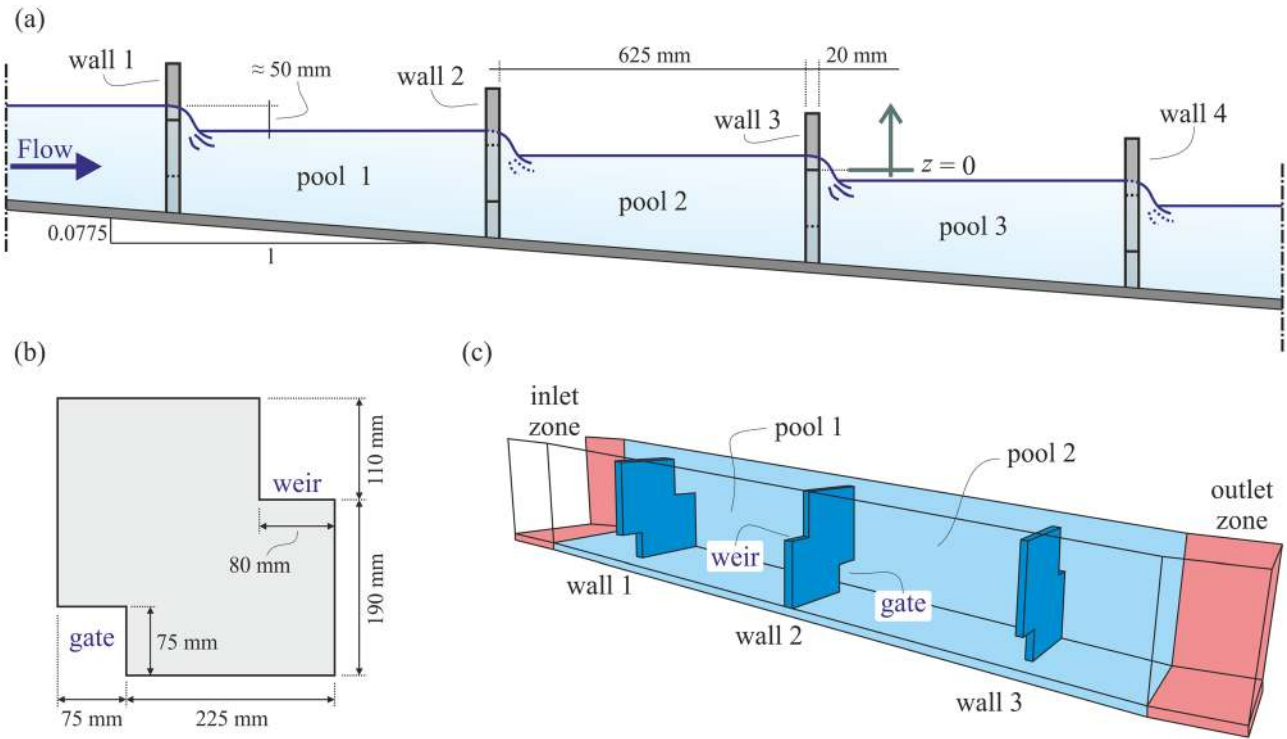
Fig. 13 Time variation of pressure at probe  $P2$  (left), probe  $P2$  focussing on the first peak (middle) and probe  $P5$  (right)

of baffles, as shown in Fig. 15. For this case  $h/dp = 1.3$ ,  $\alpha$  in the range 0.01 (standard) to 0.001 was tested and particle shifting based of Fick's equation [34] was applied to regularise the particle distributions. The particles are shifted a distance  $\delta r_s$  according to the equation

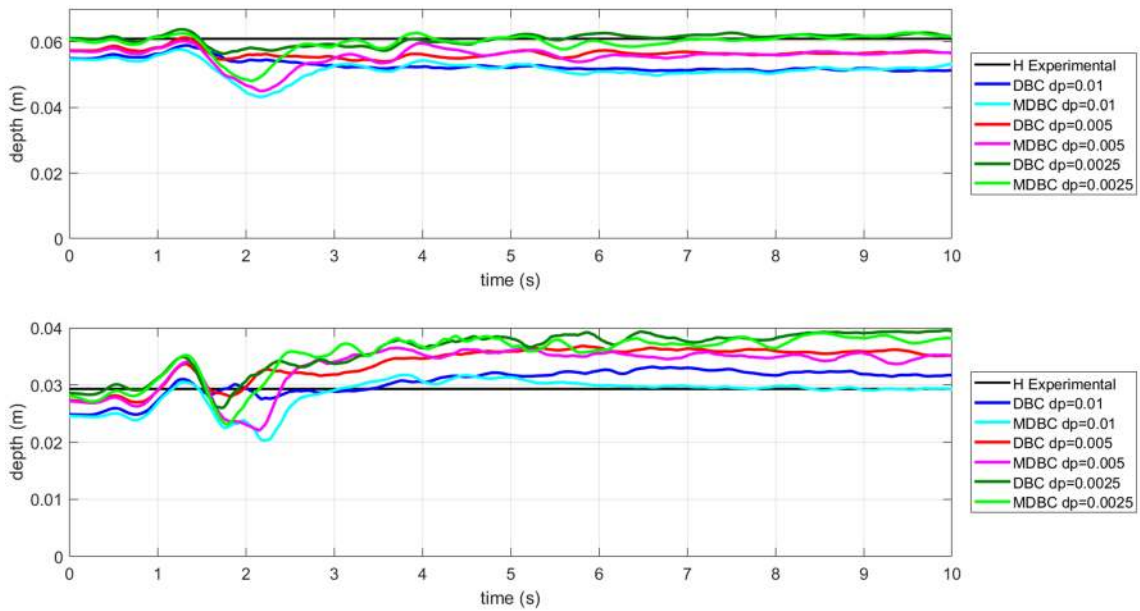
$$\delta r_s = -D \nabla C_i \tag{13}$$

where  $C$  is the particle concentration and  $D$  is the diffusion coefficient given by Skillen et al. [34] as

$$D = Ahu_i dt \tag{14}$$



**Fig. 14** Fish pass: **a** scheme of the experimental setup, note reference  $z=0$  on wall 3; **b** detail of a vertical cross baffle; **c** 3-D view of the flow domain, showing pools, gates and weirs



**Fig. 15** Water levels with gates open in pools 1 (top) and 2 (bottom) for flow rate  $Q_2$ , results averaged using a moving mean over 0.3 s

where  $A$  is a dimensionless constant which takes the value  $A = 2$  for this case.

Experiments conducted at the University of Parma are used for comparison. Five test cases were measured: two

involving flow only through the gates; three involving flow over only the weirs. The flow rates and water depths are shown in Table 1 where all the water depths are measured with respect to the base of the weir in wall 3.

**Table 1** Flow rates and water depths measured with respect to weir 3 for fish pass test cases

	$Q$ (l/s)	$H_{in}$ (mm)	$H1$ (mm)	$H2$ (mm)	$H3$ (mm)	Note
$Q1$	1.70	3.9	-10.4	-24.9	-37.2	Gate only
$Q2$	2.45	46.2	15.9	-15.9	-44.9	Gate only
$Q3$	2.267	169	115	63	13	Weir only
$Q4$	1.963	160	109	57	7	Weir only
$Q5$	1.634	154	103	51	1	Weir only

The numerical fish pass uses the inflow-outflow boundary conditions described in [31] to control the fluid particles entering and leaving the channel. The inlet is created in the centre of the upstream pool and the outlet in the centre of pool 3. To set up steady conditions, the experimental levels in each pool are first input and velocities are increased to provide the experimental flow rate. The correct flow rates are then maintained at inlet and outlet and water levels may be compared with experiment.  $H_{in}$  is the depth in the inlet zone above the base of the weir in Gate 3 ( $z = 0$  in Fig. 14).  $H1$ ,  $H2$ ,  $H3$  are depths in the centre of pools 1, 2, 3.

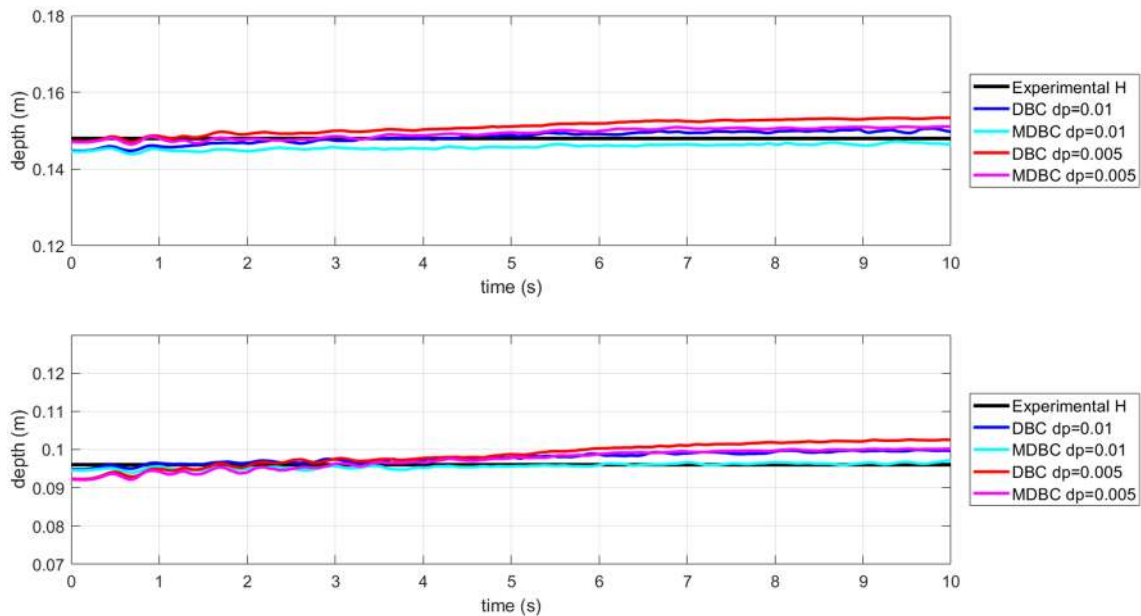
#### 4.4.1 Gate only cases

We consider  $Q2$  with an initial particle spacing of  $dp = 0.01$  m (150,827 particles),  $dp = 0.005$  m (914,467 particles) and  $dp = 0.0025$  m (6,234,131 particles); results for  $Q1$  were similar. Water level results are shown in Fig. 15. The water level in pool 1 with the smallest spacing is in close agreement with experiment although in pool 2 it is 0.0075 m, or three particle spacings, different.

#### 4.4.2 Weir only cases

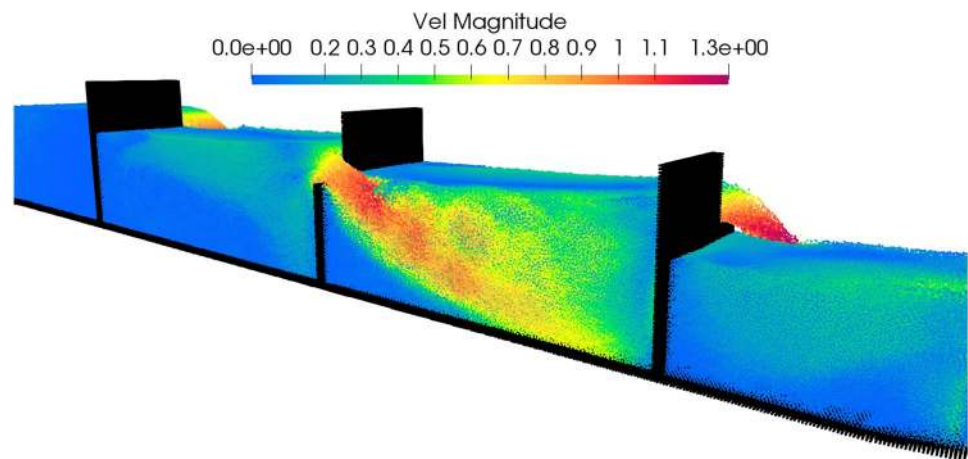
For the cases  $Q3$ – $5$  flow is only allowed to pass over the weirs, as the gates have been blocked off. Two particle spacings of  $dp = 0.01$  m (207,784 particles) and  $dp = 0.005$  m (1,372,059 particles) are used. We show results for  $Q5$  as all cases are quite similar and the water levels are in close agreement with experiment, within one particle spacing, shown in Fig. 16.

These results show that flow through a very complex geometry may be modelled with mDBC. The results with weir only show very close agreement with experiment while those with gate only were less close. This is probably because rapidly varying free surface flows are well suited to SPH, corresponding with accelerated flow over the weirs. With accelerated flow through the gates and relatively tranquil free surface flow, mixing due to turbulence will be significant and the SPH model does not have a turbulence model. With the flow over the gates mixing due to the jets over the gate is gravity dominated. Flow visualisation is shown in Fig. 17 for flow rate  $Q3$  with particle spacing  $dp = 0.005$  m.



**Fig. 16** Water levels with weirs open in pools 1 (top) and 2 (bottom) for flow rate  $Q5$ , results averaged using a moving mean over 0.3 s

**Fig. 17** Flow over weirs from SPH simulation with contours showing velocity magnitude, the set up shown if for flow rate  $Q_3$  with particle spacing  $dp = 0.005$  m (1,372,059 particles)



**Table 2** GPU specifications

	Tesla V100	RTX 2080 Ti	Tesla K40
GPU microarchitecture	Volta	Turing	Kepler
Compute capability	7.0	7.5	3.5
Global memory (MB)	16,128	10,989	12,207
Cuda cores	5120	4352	2880
Multiprocessors	80	68	15
GPU Max clock rate (MHz)	1530	1545	745
Memory clock rate (MHz)	877	7000	3004
Memory bus width (bits)	4096	352	384

**Table 3** Runtimes of the 2-D sloshing tank simulation using  $dp = 0.002$  m with different hardware devices

Hardware	DBC (s)	mDBC (s)	mDBC/DBC
CPU i7-6700 K	7350	8191	1.11
Tesla K40	811	967	1.19
GeForce RTX 2080 Ti	625	681	1.09
Tesla V100	376	430	1.14

## 4.5 Computational performance

The computational performance of DualSPHysics on GPUs with mDBC and DBC is of practical importance as real problems generally require high resolution. This section compares the runtimes of DBC and mDBC for the same resolution, although mDBC achieves a given level of accuracy with lower resolution than mDBC. The performance of both boundary conditions is presented for two cases on a CPU for reference and on different generations of GPUs. The CPU device used here is an Intel Core i7-6700 K with a clock speed of 4.0 GHz and 8 execution threads (4 physical cores). The specifications of the GPU cards (commonly used for numerical computing) are shown in the Table 2.

**Table 4** Runtimes of the 3-D dam break simulation with different hardware devices

Hardware	$dp$ (m)	DBC (s)	mDBC (s)	mDBC/DBC
CPU i7-6700 K	0.02	9998	12,442	1.24
Tesla K40		624	768	1.23
GeForce RTX 2080 Ti		224	252	1.12
Tesla V100		171	190	1.11
Tesla K40	0.01	7366	8700	1.18
GeForce RTX 2080 Ti		1406	1586	1.13
Tesla V100		1050	1180	1.12

Two SPHERIC benchmark cases are selected to analyse the performance with the different boundary conditions; a 2-D case with moving boundaries and a 3-D problem. The 2-D sloshing tank case shown in Sect. 4.2 was simulated for 7 physical seconds using  $dp = 0.002$  m and 26,791 particles. This simulation was executed on the devices in Table 2, and the runtimes are shown in Table 3 with the ratio mDBC to DBC runtime. The 3-D dam break case was simulated for 6 physical seconds using  $dp = 0.02$  m (172,422 particles) and  $dp = 0.01$  m (1,015,809 particles). The execution times of the 3-D dam break case described in Sect. 4.3 are included in Table 4.

The use of mDBC thus results in a 10–20% increase in execution time over DBC for 2-D and 3-D simulations for the same resolution. This increase depends on the number and distribution of boundary and fluid particles of the simulation case, since extra calculation time is required for the interpolation carried on the ghost nodes projected from the boundary particles into the fluid domain. In practice a given level of accuracy may be achieved with mDBC with a lower level of resolution than DBC, reducing execution time below that for DBC.

## 5 Conclusions

The dynamic boundary condition has been improved by providing solid boundary particles with density extrapolated from mirror positions within the fluid without sacrificing any of the robustness of the original formulation. Pressure on a solid surface is predicted accurately. This has been demonstrated for the still water case with a sharp corner by recovering hydrostatic pressure; for the dynamic 2D SPHERIC test case 10 of the sloshing tank; for the 3D SPHERIC test case 2 of the dam break impacting a cuboid obstacle; and for the new complex test case of fish pass flow with several gates and weirs. Agreement with experiment is good for the weirs only case where the flow is gravity dominated and less good for the gates only case with turbulent mixing and a tranquil free surface. The computational overhead on the original DBC is less than 25% and depends on application and the choice of GPU. However, this is more than compensated by mDBC enabling realistic simulation with lower resolution and a smaller number of particles. DualSPHysics is now available with this functionality. This capability should be particularly beneficial for complex problems requiring high resolution such as the rubble mound breakwater [32] and marine vehicles [5] studied previously with DBC.

Further work will allow fluid velocity at mirror points to be extrapolated to solid boundary particles. This will allow for the creation of more accurate no-slip boundaries as well as slip and partial slip boundaries for some non-Newtonian applications.

**Acknowledgements** This work was partially financed by Xunta de Galicia (Spain) under project ED431C 2017/64 “Programa de Consolidación e Estructuración de Unidades de Investigación Competitivas (Grupos de Referencia Competitiva)” co-funded by European Regional Development Fund (FEDER). The work is funded by the Ministry of Economy and Competitiveness of the Government of Spain under project “WELCOME ENE2016-75074-C2-1-R”. We are grateful for funding from the European Union Horizon 2020 programme under the ENERXICO Project, Grant Agreement No. 828947 and the Mexican CONACYT- SENER Hidrocarburos Grant Agreement No. B-S-69926. The work is also funded by the Italian Ministry of Education, Universities and Research under the Scientific Independence of young Researchers project, grant number RBSI14R1GP, CUP code D92I15000190001. Aaron English PhD programme is supported by Unilever and EPSRC, funding code EP/P510579/1. Dr. J. M. Domínguez acknowledges funding from Spanish government under the program “Juan de la Cierva-incorporación 2017” (IJCI-2017-32592).

**Availability of data, material and code** All data, material and code are available from the download package of the DualSPHysics code found through <https://dual.sphysics.org/>.

## Declarations

**Conflicts of interest** The authors declare no conflicts of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Souto-Iglesias A, Delorme L, Pérez-Rojas L, Abril-Pérez S (2006) Liquid moment amplitude assessment in sloshing type problems with smooth particle hydrodynamics. *Ocean Eng*. <https://doi.org/10.1016/j.oceaneng.2005.10.011>
2. Fourtakas G, Rogers BD (2016) Modelling multi-phase liquid-sediment scour and resuspension induced by rapid flows using smoothed particle hydrodynamics (SPH) accelerated with a graphics processing unit (GPU). *Adv Water Resour* 92:186–199. <https://doi.org/10.1016/j.advwatres.2016.04.009>
3. Canelas RB, Domínguez JM, Crespo AJC et al (2017) Resolved Simulation of a granular-fluid flow with a coupled SPH-DCDEM Model. *J Hydraul Eng* 143:06017012. [https://doi.org/10.1061/\(ASCE\)HY.1943-7900.0001331](https://doi.org/10.1061/(ASCE)HY.1943-7900.0001331)
4. Colagrossi A, Nikolov G, Durante D et al (2019) Viscous flow past a cylinder close to a free surface: benchmarks with steady, periodic and metastable responses, solved by meshfree and mesh-based schemes. *Comput Fluids*. <https://doi.org/10.1016/j.compfluid.2019.01.007>
5. Mogan SRC, Chen D, Hartwig JW et al (2018) Hydrodynamic analysis and optimization of the Titan submarine via the SPH and finite-volume methods. *Comput Fluids* 174:271–282. <https://doi.org/10.1016/j.compfluid.2018.08.014>
6. Manenti W, Domínguez, et al (2019) SPH modeling of water-related natural hazards. *Water* 11:1875. <https://doi.org/10.3390/w11091875>
7. Barreiro A, Domínguez JM, Crespo AJC et al (2014) Integration of UAV photogrammetry and SPH modelling of fluids to study runoff on real terrains. *PLoS ONE*. <https://doi.org/10.1371/journal.pone.0111031>
8. Novak G, Tafuni A, Domínguez JM et al (2019) A numerical study of fluid flow in a vertical slot fishway with the smoothed particle hydrodynamics method. *Water* 11:1928. <https://doi.org/10.3390/w11091928>
9. Zhang F, Crespo A, Altomare C et al (2018) DualSPHysics: a numerical tool to simulate real breakwaters. *J Hydrodyn* 30:95–105. <https://doi.org/10.1007/s42241-018-0010-0>
10. Vacondio R, Altomare C, De Leffe M et al (2020) Grand challenges for Smoothed Particle Hydrodynamics numerical schemes. *Comp Part Mech*. <https://doi.org/10.1007/s40571-020-00354-1>
11. Lind SJ, Rogers BD, Stansby PK (2020) Review of smoothed particle hydrodynamics: towards converged Lagrangian flow modelling. *Proc R Soc A* 476: 20190801. <https://doi.org/10.1098/rspa.2019.0801>
12. Gotoh H, Khayyer A (2018) On the state-of-the-art of particle methods for coastal and ocean engineering. *Coast Eng J* 60:79–103. <https://doi.org/10.1080/21664250.2018.1436243>

13. Monaghan JJ (1994) Simulating free surface flows with SPH. *J Comput Phys* 110:399–406. <https://doi.org/10.1006/jcph.1994.1034>
14. Monaghan JJ, Kajtar JB (2009) SPH particle boundary forces for arbitrary boundaries. *Comput Phys Commun.* <https://doi.org/10.1016/j.cpc.2009.05.008>
15. Kulasegaram S, Bonet J, Lewis RW, Profit M (2004) A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications. *Comput Mech.* <https://doi.org/10.1007/s00466-003-0534-0>
16. Leroy A, Violeau D, Ferrand M, Kassiotis C (2014) Unified semi-analytical wall boundary conditions applied to 2-D incompressible SPH. *J Comput Phys.* <https://doi.org/10.1016/j.jcp.2013.12.035>
17. Mayrhofer A, Rogers BD, Violeau D, Ferrand M (2013) Investigation of wall bounded flows using SPH and the unified semi-analytical wall boundary conditions. *Comput Phys Commun.* <https://doi.org/10.1016/j.cpc.2013.07.004>
18. Crespo AJC, Gómez-Gesteira M, Dalrymple RA (2007) Boundary conditions generated by dynamic particles in SPH methods. *Comput Mater Contin* 5:173–184. <https://doi.org/10.3970/cm.2007.005.173>
19. Marrone S, Antuono M, Colagrossi A et al (2011)  $\delta$ -SPH model for simulating violent impact flows. *Comput Methods Appl Mech Eng* 200:1526–1542. <https://doi.org/10.1016/j.cma.2010.12.016>
20. Adami S, Hu XY, Adams NA (2012) A generalized wall boundary condition for smoothed particle hydrodynamics. *J Comput Phys* 231:7057–7075. <https://doi.org/10.1016/j.jcp.2012.05.005>
21. Fourtakas G, Domínguez JM, Vacondio R, Rogers BD (2019) Local uniform stencil (LUST) boundary condition for arbitrary 3-D boundaries in parallel smoothed particle hydrodynamics (SPH) models. *Comput Fluids* 190:346–361. <https://doi.org/10.1016/j.compfluid.2019.06.009>
22. Ren B, He M, Dong P, Wen H (2015) Nonlinear simulations of wave-induced motions of a freely floating body using WCSPH method. *Appl Ocean Res* 50:1–12. <https://doi.org/10.1016/j.apor.2014.12.003>
23. Domínguez JM, Fourtakas G, Altomare C, Canelas RB, Tafuni A, García-Feal O, Martínez-Estévez I, Mokus A, Vacondio R, Crespo AJC, Rogers BD, Stansby PK, Gómez-Gesteira M (2021) Dual-SPHysics: from fluid dynamics to multiphysics problems. *Comput Part Mech.* <https://doi.org/10.1007/s40571-021-00404-2>
24. Domínguez JM, Crespo AJC, Gómez-Gesteira M (2013) Optimization strategies for CPU and GPU implementations of a smoothed particle hydrodynamics method. *Comput Phys Commun* 184:617–627. <https://doi.org/10.1016/j.cpc.2012.10.015>
25. Liu MB, Liu GR (2006) Restoring particle consistency in smoothed particle hydrodynamics. *Appl Numer Math* 56:19–36. <https://doi.org/10.1016/j.apnum.2005.02.012>
26. Vacondio R, Rogers BD, Stansby PK (2012) Smoothed particle hydrodynamics: approximate zero-consistent 2-D boundary conditions and still shallow-water tests. *Int J Numer Methods Fluids.* <https://doi.org/10.1002/flid.2559>
27. Wendland H (1995) Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv Comput Math.* <https://doi.org/10.1007/BF02123482>
28. Molteni D, Colagrossi A (2009) A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH. *Comput Phys Commun* 180:861–872. <https://doi.org/10.1016/j.cpc.2008.12.004>
29. Dalrymple RA, Knio O (2001) SPH modelling of water waves. In: *Coastal dynamics, American society of civil engineers (ASCE)* pp 779–787. [https://doi.org/10.1061/40566\(260\)80](https://doi.org/10.1061/40566(260)80)
30. Domínguez JM, Crespo AJC, Fourtakas G, et al (2015) Evaluation of reliability and efficiency of different boundary conditions in an SPH code. In: *Proceedings of the 10th international SPHERIC workshop.* pp 341–348
31. Tafuni A, Domínguez JM, Vacondio R, Crespo AJC (2018) A versatile algorithm for the treatment of open boundary conditions in smoothed particle hydrodynamics GPU models. *Comput Methods Appl Mech Eng* 342:604–624. <https://doi.org/10.1016/j.cma.2018.08.004>
32. Altomare C, Crespo AJC, Rogers BD et al (2014) Numerical modelling of armour block sea breakwater with smoothed particle hydrodynamics. *Comput Struct* 130:34–45. <https://doi.org/10.1016/j.compstruc.2013.10.011>
33. Kleefsman KMT, Fekken G, Veldman AEP et al (2005) A Volume-of-Fluid based simulation method for wave impact problems. *J Comput Phys* 206:363–393. <https://doi.org/10.1016/j.jcp.2004.12.007>
34. Skillen A, Lind S, Stansby PK, Rogers BD (2013) Incompressible smoothed particle hydrodynamics (SPH) with reduced temporal noise and generalised Fickian smoothing applied to body–water slam and efficient wave–body interaction. *Comput Methods Appl Mech Eng* 265:163–173. <https://doi.org/10.1016/j.cma.2013.05.017>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.