




# On the use of multibody dynamics techniques to simulate fluid dynamics and fluid–solid interaction problems

M. Rakhsha<sup>1</sup> · L. Yang<sup>1</sup> · W. Hu<sup>1</sup> · D. Negrut<sup>1</sup> 

Received: 1 July 2020 / Accepted: 15 February 2021

© The Author(s), under exclusive licence to Springer Nature B.V. part of Springer Nature 2021

**Abstract** A multibody dynamics-based solution to the fluid dynamics problem is compared herein to two established Lagrangian-based techniques used by the computational fluid dynamics (CFD) community. The multibody dynamics-based solution has two salient attributes: it enforces the incompressibility condition through bilateral kinematic constraints, and it treats the coupling with the solid phase via unilateral kinematic constraints. The multibody dynamics-based solution, called herein the Kinematically Constrained Smoothed Particle Hydrodynamics (KCSPH) method, is a Lagrangian approach to solving the CFD problem. It relies on the Smoothed Particle Hydrodynamics (SPH) method to discretize the spatial differential operators in the Navier–Stokes equations, and on the Newton–Euler equations of multibody dynamics to convect the SPH particles forward in time. We show that the multibody dynamics-based approach is efficient and accurate by comparing its performance with the two most commonly used SPH algorithms in the CFD community: the weakly compressible SPH (WCSPH), and the implicit SPH (ISPH) methods. The comparison is carried out in conjunction with four tests: an incompressibility benchmark test, dam break, floating cylinder, and sloshing tank. We conclude that KCSPH is a robust alternative to conventional CFD approaches for fluid–solid interaction (FSI) problems with complex/moving boundaries. The solvers and models used herein are publicly available in an open-source software called Chrono; the implementations use GPU (for WCSPH and ISPH), and multicore CPU (for KCSPH) parallel computing.

**Keywords** Multibody dynamics solution techniques · Kinematic constraints · Parallel computing · Smoothed Particle Hydrodynamics · Fluid-solid interaction

---

✉ D. Negrut  
[negrut@wisc.edu](mailto:negrut@wisc.edu)

M. Rakhsha  
[rakhsha@wisc.edu](mailto:rakhsha@wisc.edu)

L. Yang  
[lyang296@wisc.edu](mailto:lyang296@wisc.edu)

W. Hu  
[whu59@wisc.edu](mailto:whu59@wisc.edu)

<sup>1</sup> Dept. of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA

# 1 Introduction

In this contribution we assess a recently developed CFD method [44], which draws on multi-body dynamics solution techniques, against two classical approaches used in the Lagrangian CFD field: the weakly compressible SPH (WCSPH) method, and the implicit SPH (ISPH) method. Smoothed Particle Hydrodynamics [22, 41] is a meshless CFD approach to spatially discretize the mass and momentum balance equations [45]. SPH is attractive since: (i) it conveniently handles free-surface problems because unlike Eulerian methods, it eschews the interface tracking task that would require the solution of an extra transport equation, e.g., volume of fluid and/or level set [14, 32]; (ii) it interfaces well with the numerical solution of the solid mechanics and multibody dynamics problems, which almost always are also approached in a Lagrangian framework [6, 54, 59, 78]; (iii) owing to its Lagrangian nature, SPH avoids difficulties encountered in Eulerian CFD methods, such as treatment of the nonlinear advection term or mass conservation; (iv) it can handle highly transient FSI problems, where impulsive and nonsteady state dynamics come into play, potentially in a coupled-dynamics framework that involves the motion of solids; and (v) it shows promise for multiphase flows at high density ratios [37, 50, 69, 75]. On the downside, SPH: (a) loses second-order accuracy [58] unless a consistency-enforcing step is taken in its numerical solution [21, 29, 61, 73]; (b) poses challenges with, and requires careful treatment of boundary conditions [2, 27, 38].

KCSPH is inspired by work reported in [44], where the incompressibility is enforced via holonomic kinematic constraint equations asserting that the density assumes a reference value  $\rho_0$  at the location of each SPH marker. This leads to large index three differential-algebraic problems that have as many kinematic constraints as particles in the SPH solution, i.e., from tens of thousands to millions of holonomic constraints. In addition to the presence of these bilateral (equality) kinematic constraints induced by the incompressibility attribute of the flow, the presence of solid boundary conditions and the solid–fluid coupling calls for the use of unilateral (inequality) kinematic constraints that are equivalently posed as complementarity conditions [66]. Owing to the presence of both equality and inequality constraints, KCSPH relies on the solution a quadratic optimization problem typically encountered in multibody dynamics with friction and contact [51]. Of the three SPH methods discussed herein, WCSPH is the only one using a state equation to relate the pressure to density, see, for instance, [3]. The time integration is typically explicit and carried out via Runge–Kutta or predictor–corrector methods, see, for instance, [40]. The combined use of an explicit time integration scheme and a stiff incompressibility-enforcing state equation limits the size of the time step. In fact, the more one insists on incompressibility, the shorter the time step and thus the longer the simulation time. This drawback prompted the development of ISPH formulations, see, for instance, [16]. In ISPH, the pressure may be regarded as a *mechanical* property (Lagrange multiplier) thus shedding its *thermodynamical* nature (equation of state) manifest in the WCSPH approaches. Moreover, as far as the derivation of the pressure equation is concerned, many ISPH methods are at their core projection methods [13], i.e., the pressure is obtained from a Poisson equation at every time step, see, for instance, [64]. Solving a Poisson equation calls for the solution of a linear system, which poses both a computational bottleneck and a parallel implementation challenge for ISPH. However, what one gains with ISPH, when compared to WCSPH methods, is larger integration time steps and a more effective mechanism to enforce incompressibility.

The manuscript is organized as follows. Section 2 describes the governing equations for fluid and multibody dynamics systems, the latter anchoring the KCSPH numerical solution. Section 3 outlines the three SPH-based, CFD numerical methods used in this study. To that

end, it starts with a brief overview of the SPH method in which we express the SPH space discretization via field-independent discretization matrices. This allows for separating the space discretization from the implementation of the time integration scheme. An overview of the KCSPH method and its derivation from the Newton–Euler equations is provided in Sect. 3.1. The classical WSPH method is described in Sect. 3.2. Section 3.3 focuses on the ISPH method. Section 3.4 concentrates on handling boundary conditions. Section 4 reports numerical results in relation to four test problems: an incompressibility benchmark test, dam break, floating cylinder, and fluid sloshing. In Sect. 5 we discuss perceived strengths and weaknesses of KCSPH relative to WSPH and ISPH, and conclude with directions of future work.

## 2 Governing equations

### 2.1 Multibody dynamics: the equations of motion

We follow the notation used in [25] to define the configuration of the system of bodies by a set of generalized coordinates for the position and orientation of a rigid body in the 3D Euclidean space as  $\mathbf{r}_j \in \mathbb{R}^3$  and  $\boldsymbol{\epsilon}_j \in \mathbb{R}^4$ , i.e., the absolute position of the center of mass, and the Euler parameters associated with orientation of body  $j$ . The Euler parameters satisfy the normalization constraint  $\boldsymbol{\epsilon}_j^T \cdot \boldsymbol{\epsilon}_j = 1$ . Combining the set of generalized coordinates of different bodies for a system of  $n_b$  bodies, one can write the set of generalized coordinates describing the system at position level as  $\mathbf{q} = [\mathbf{r}_1^T, \boldsymbol{\epsilon}_1^T, \dots, \mathbf{r}_{n_b}^T, \boldsymbol{\epsilon}_{n_b}^T]^T \in \mathbb{R}^{7n_b}$ , and at velocity level as  $\dot{\mathbf{q}} = [\dot{\mathbf{r}}_1^T, \dot{\boldsymbol{\epsilon}}_1^T, \dots, \dot{\mathbf{r}}_{n_b}^T, \dot{\boldsymbol{\epsilon}}_{n_b}^T]^T \in \mathbb{R}^{7n_b}$ . Instead of using the time derivative of the Euler parameters, one may choose to use angular velocities to describe the state of the system at the velocity level by  $\mathbf{v} = [\dot{\mathbf{r}}_1^T, \bar{\boldsymbol{\omega}}_1^T, \dots, \dot{\mathbf{r}}_{n_b}^T, \bar{\boldsymbol{\omega}}_{n_b}^T]^T \in \mathbb{R}^{6n_b}$ , which reduces the problem size. The transformation from the derivatives of Euler parameters,  $\dot{\boldsymbol{\epsilon}}_B$ , to angular velocities represented in the body-fixed frame,  $\bar{\boldsymbol{\omega}}_B$ , for each body is governed by

$$\dot{\boldsymbol{\epsilon}}_B = \frac{1}{2} \mathbf{G}^T(\boldsymbol{\epsilon}_B) \bar{\boldsymbol{\omega}}_B, \quad (1)$$

where matrix  $\mathbf{G} \in \mathbb{R}^{3 \times 4}$  depends linearly on the Euler parameters  $\boldsymbol{\epsilon}_B$ . Therefore, a block diagonal matrix  $\mathbf{L}(\mathbf{q}) \equiv \text{diag}[\mathbf{I}_{3 \times 3}, \frac{1}{2} \mathbf{G}^T(\boldsymbol{\epsilon}_1), \dots, \mathbf{I}_{3 \times 3}, \frac{1}{2} \mathbf{G}^T(\boldsymbol{\epsilon}_{n_b})] \in \mathbb{R}^{7n_b \times 6n_b}$  is used to express via  $\dot{\mathbf{q}} = \mathbf{L}(\mathbf{q}) \mathbf{v}$ , the relationship between  $\dot{\mathbf{q}}$  and  $\mathbf{v}$ , where  $\mathbf{I}_{3 \times 3}$  is the identity matrix [25].

The Newton–Euler equations of motion that describe the motion of a system of bodies interacting through friction, contact, and bilateral constraints, assume the following form of a differential variational inequality (DVI) problem, see, for instance, [51, 70]:

$$\dot{\mathbf{q}} = \mathbf{L}(\mathbf{q}) \mathbf{v}, \quad (2a)$$

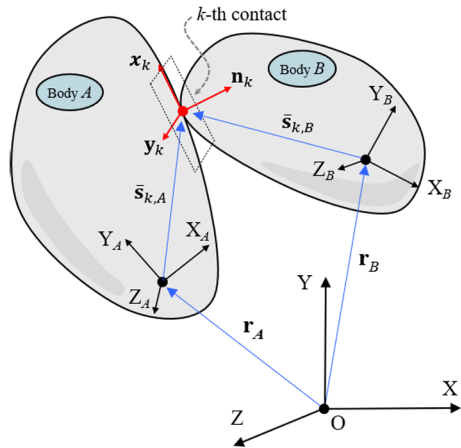
$$\mathbf{M} \dot{\mathbf{v}} = \mathbf{f}(t, \mathbf{q}, \mathbf{v}) + \nabla_{\mathbf{q}} \mathbf{g}(\mathbf{q}, t) \Delta + \sum_{i \in \mathcal{A}(\mathbf{q}, \delta)} (\hat{\gamma}_{i,n} \mathbf{D}_{i,n} + \hat{\gamma}_{i,x} \mathbf{D}_{i,x} + \hat{\gamma}_{i,y} \mathbf{D}_{i,y}), \quad (2b)$$

$$0 = \mathbf{g}(\mathbf{q}, t), \quad (2c)$$

$$i \in \mathcal{A}(\mathbf{q}(t), \delta) : 0 \leq \hat{\gamma}_{i,n} \perp \Phi_i(\mathbf{q}) \geq 0, \quad (2d)$$

$$(\hat{\gamma}_{i,x}, \hat{\gamma}_{i,y}) = \underset{\sqrt{(\hat{\gamma}_{i,x})^2 + (\hat{\gamma}_{i,y})^2} \leq \mu_i^f \hat{\gamma}_{i,n}}{\text{argmin}} \quad \mathbf{v}^T (\hat{\gamma}_{i,x} \mathbf{D}_{i,x} + \hat{\gamma}_{i,y} \mathbf{D}_{i,y}), \quad (2e)$$

**Fig. 1** Schematic of contact between two bodies



where  $\mathbf{f}(t, \mathbf{q}, \mathbf{v})$  is the array of external forces;  $\mathbf{M}$  is the constant system mass matrix;  $\mathbf{g}(\mathbf{q}, t) = 0$  defines the holonomic constraints between the generalized coordinates  $\mathbf{q}$ ;  $\Lambda$  represents the set of Lagrange multipliers associated with these holonomic constraints;  $\nabla_{\mathbf{q}}\mathbf{g}(\mathbf{q}, t)\Lambda$  describes the reaction forces associated with the bilateral constraints in the force balance equation; and,  $\mathcal{A}(\mathbf{q}, \delta)$  is the set of active and potential unilateral constraints based on the bodies that are mutually less than a gap  $\delta$  apart. For contact  $i$ , the tangent space generator  $\mathbf{D}_i \equiv [\mathbf{D}_{i,n}, \mathbf{D}_{i,x}, \mathbf{D}_{i,y}] \in \mathbb{R}^{6n_b \times 3}$  is defined as [51]

$$\mathbf{D}_i^T = \begin{bmatrix} \mathbf{0} & \dots & -\mathbf{A}_i^T & \mathbf{A}_i^T \mathbf{A}_A \tilde{\mathbf{s}}_{i,A} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{A}_i^T & -\mathbf{A}_i^T \mathbf{A}_B \tilde{\mathbf{s}}_{i,B} & \dots & \mathbf{0} \end{bmatrix},$$

where  $\mathbf{A}_i = [\mathbf{n}_i, \mathbf{x}_i, \mathbf{y}_i] \in \mathbb{R}^{3 \times 3}$  is the orientation matrix associated with contact  $i$ ,  $\mathbf{A}_A = \mathbf{A}(\boldsymbol{\epsilon}_A)$  and  $\mathbf{A}_B = \mathbf{A}(\boldsymbol{\epsilon}_B)$  are the rotation matrices, and  $\boldsymbol{\epsilon}_A$  and  $\boldsymbol{\epsilon}_B \in \mathbb{R}^4$  are the Euler parameters associated with orientation of body  $A$  and  $B$ , respectively; the vectors  $\tilde{\mathbf{s}}_{i,A}$  and  $\tilde{\mathbf{s}}_{i,B} \in \mathbb{R}^3$  represent the contact point positions in body-relative coordinates as shown in Fig. 1. Above, the operator “tilde” applied to a three dimensional vector  $\mathbf{a}$  produces a matrix  $\tilde{\mathbf{a}} \in \mathbb{R}^{3 \times 3}$  such that  $\mathbf{a} \times \mathbf{b} = \tilde{\mathbf{a}} \mathbf{b}$  for all  $\mathbf{b} \in \mathbb{R}^3$ .

Equation (2d) expresses a complementarity condition between  $\Phi$ , the gap (distance) between bodies  $A$  and  $B$ , and  $\hat{\gamma}_{i,n}$ , the Lagrange multiplier for the normal (contact) force associated with the contact  $i$ . The complementarity condition states that of  $\hat{\gamma}_{i,n}$  and  $\Phi$ , at least one is zero and the other one is nonnegative. Indeed, when the gap function is zero (contact is present), the normal contact force is nonnegative; and, conversely, when the normal contact force is zero the gap function is nonnegative. The forces associated with contact  $i$  can be expressed as  $\mathbf{f}_{i,N} = \hat{\gamma}_{i,n} \mathbf{n}_i$ , and  $\mathbf{f}_{i,T} = \hat{\gamma}_{i,x} \mathbf{x}_i + \hat{\gamma}_{i,y} \mathbf{y}_i$ , which are the contact and friction forces, respectively; and,  $\hat{\gamma}_{i,x}$  and  $\hat{\gamma}_{i,y}$  are the components of the friction force in the tangent plane. The friction forces are determined based on the Coulomb dry-friction model [66]

$$\sqrt{(\hat{\gamma}_{i,x})^2 + (\hat{\gamma}_{i,y})^2} \leq \mu_i^f \hat{\gamma}_{i,n}, \quad (3a)$$

$$\|\mathbf{v}_{i,T}\| \left( \sqrt{(\hat{\gamma}_{i,x})^2 + (\hat{\gamma}_{i,y})^2} - \mu_i^f \hat{\gamma}_{i,n} \right) = 0, \quad (3b)$$

$$\mathbf{f}_{i,T} \cdot \mathbf{v}_{i,T} = -\|\mathbf{f}_{i,T}\| \|\mathbf{v}_{i,T}\|, \quad (3c)$$

where  $\mathbf{v}_{i,T}$  denotes the relative tangential velocity between bodies  $A$  and  $B$  at the contact point. More specifically, Eq. (3a) states that the friction force is less than the normal force scaled by the friction coefficient. Equation (3b) states a complementarity condition. Lastly, Eq. (3c), which involves the dot/inner product of two vectors, states that the friction force is in the opposite direction of  $\mathbf{v}_{i,T}$ . Note that the constraint minimization problem in Eq. (2e), whose first order Karush–Kuhn–Tucker optimality conditions are Eqs. (3a)–(3c), represents one way of stating that friction is governed by the Coulomb model, see, for instance, [51]. The DVI problem stated in Eqs. (2a)–(2e) can be solved with a variety of techniques, see [1, 4, 5, 10, 26, 35, 36, 43, 65, 67].

## 2.2 Fluid dynamics: the equations of motion

The mass and momentum balance, i.e., the continuity and Navier–Stokes equations are formulated for the fluid phase as [23]

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{u}, \quad (4)$$

$$\frac{d\mathbf{u}}{dt} = \frac{1}{\rho} \nabla \cdot \boldsymbol{\sigma} + \mathbf{f}^b = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \boldsymbol{\tau} + \mathbf{f}^b, \quad (5)$$

where

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix} = -p\mathbf{I} + \boldsymbol{\tau} = -\begin{bmatrix} p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{bmatrix} + \begin{bmatrix} \sigma_{xx} + p & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} + p & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} + p \end{bmatrix}$$

is the Cauchy stress tensor; and  $p$  and  $\boldsymbol{\tau}$  provide the volumetric and deviatoric decomposition of the stress tensor. The pressure  $p$ , a mechanical property of the system, is tied to the trace of the stress tensor,  $p = -\frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz})$ . Upon adopting a Newtonian constitutive model to express  $\boldsymbol{\tau} = \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$  and accounting for the incompressible flow assumption ( $\nabla \cdot \mathbf{u} = 0$ ), Eq. (5) leads to the following form of the Navier–Stokes equations:

$$\frac{d\mathbf{u}}{dt} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}^b, \quad (6)$$

where  $\nu = \mu/\rho$  and  $\rho$  are the fluid kinematic viscosity and density, respectively;  $\mathbf{f}^b$  is the volumetric force density; and  $\mathbf{u}$  is the flow velocity.

## 3 Numerical solution methods

KCSPH, WCSPH, and ISPH are succinctly presented below at a level of detail sufficient to recreate the algorithms used to generate the numerical results reported in Sect. 4. For a more thorough discussion, particularly of a theoretical (as opposed to algorithmic) nature, the interested reader is referred to other sources. Specifically, approaches similar to KCSPH are discussed in [11, 19, 44]. For the class of methods that WCSPH belongs to, i.e., solutions that resort to an equation of state for enforcing incompressibility, see, for instance, [3, 7, 12, 15, 46, 48, 54, 68]. For the class of methods ISPH belongs to, the reader is referred to [16, 28, 33, 59, 73, 77].

We employ SPH for the spatial discretization of Eqs. (4) and (6) to approximate a function  $f$  at location  $\mathbf{r}_i$  as [47]

$$f(\mathbf{r}_i) \approx \langle f \rangle_i = \sum_{j \in \mathcal{S}(i)} \frac{m_j}{\rho_j} f(\mathbf{r}_j) W_{ij}, \quad (7)$$

where  $\langle f \rangle_i$  indicates the SPH approximation of  $f$  at the location of SPH particle, or marker,  $i$ ;  $\mathcal{S}(i)$  represents the collection of SPH particles found in the support domain associated with particle  $i$ ;  $\rho_j$  is the density  $\rho(\mathbf{r}_j)$  at location  $\mathbf{r}_j$  of particle  $j$ ;  $m_j = \rho_j V_j$  and  $V_j = (\sum_{k \in \mathcal{S}(j)} W_{jk})^{-1}$  are the mass and volume associated with marker  $j$ , respectively;  $W_{ij} \equiv W(|\mathbf{r}_i - \mathbf{r}_j|, h)$ , where  $|\mathbf{r}|$  is the length of  $\mathbf{r}$ . The kernel function  $W$  can assume various expressions, e.g., a cubic spline kernel for 3D problems:

$$W(|\mathbf{r}|, h) = \frac{5}{14\pi h^3} \times \begin{cases} (2-q)^3 - 4(1-q)^3, & 0 \leq q < 1, \\ (2-q)^3, & 1 \leq q < 2, \\ 0, & q \geq 2, \end{cases} \quad (8)$$

where, if the kernel function is located at the origin,  $q \equiv |\mathbf{r}|/h$ . The radius of the support domain,  $\kappa h$ , is proportional to the characteristic length  $h$  through the parameter  $\kappa$ , the latter commonly set to 2 for the cubic spline kernel.

The standard SPH gradient and Laplacian approximations assume the following expressions, respectively [47]:

$$\nabla f(\mathbf{r}_i) \approx \langle \nabla f \rangle_i = \sum_{j \in \mathcal{S}(i)} V_j (f_j - f_i) \nabla_i W_{ij}, \quad (9)$$

$$\nabla^2 f(\mathbf{r}_i) \approx \langle \nabla^2 f \rangle_i = 2 \sum_{j \in \mathcal{S}(i)} V_j (\mathbf{e}_{ij} \cdot \nabla_i W_{ij}) \frac{f_i - f_j}{|\mathbf{r}_{ij}|}, \quad (10)$$

where  $\mathbf{e}_{ij} = \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|}$  and  $\nabla_i$  denotes differentiation in space with respect to the coordinates of SPH particle  $i$ , i.e.,

$$\nabla_i W_{ij} = \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} \frac{\partial W}{\partial q} \frac{\partial q}{\partial |\mathbf{r}_{ij}|} \Big|_{i,j} = \frac{-15\mathbf{r}_{ij}}{14\pi h^5 q} \times \begin{cases} (2-q)^2 - 4(1-q)^2, & 0 \leq q < 1, \\ (2-q)^2, & 1 \leq q < 2, \\ 0, & q \geq 2. \end{cases} \quad (11)$$

Elaborating on the concept of convergence and accuracy, if a numerical discretization matches the first  $m$  terms of the Taylor expansion of the solution, then the numerical approximation is said to be  $(m+1)$ th-order accurate and  $\mathcal{C}^m$  consistent. The standard SPH discretizations have  $\mathcal{C}^1$  consistency (exact approximation of linear functions) in the interior of the domain provided a regular particle distribution is maintained. If the particle regularity is lost over time, the standard discretization is no longer  $\mathcal{C}^1$  consistent and corrections are necessary to maintain the consistency order. However, once the kernel function is altered to retain consistency, the SPH discretization will forfeit its symmetry attributes, thus losing its conservation trait. Whereas conservative schemes are essential for discretization of the pressure gradient term and pressure driven flows, consistent schemes play a more important role in viscous flows. One side effect of using a consistent discretization is that it requires smaller

kernel support [34, 72]. Reducing this support size reduces the number of SPH neighbors for a given particle, which is critical in implicit solvers since linear system fill-in is dictated by the number of SPH particle neighbors. This is the rationale for using a consistent formulation for the implicit SPH method discussed herein, see Sect. 3.3, whenever handling an interior flow scenario.

The “consistent” flavor of the discretization already defined for the conservative case in Eqs. (9) and (10) assumes the expression [21, 61]:

$$\nabla f(\mathbf{r}_i) \approx \langle \nabla f \rangle_i = \sum_{j \in \mathcal{S}(i)} V_j (f_j - f_i) \mathbf{G}_i \nabla_i W_{ij}, \quad (12)$$

$$\nabla^2 f(\mathbf{r}_i) \approx \langle \nabla^2 f \rangle_i = 2 \sum_{j \in \mathcal{S}(i)} [\mathbf{L}_i : (\mathbf{e}_{ij} \otimes \nabla_i W_{ij})] \left( \frac{f_i - f_j}{|\mathbf{r}_{ij}|} - \mathbf{e}_{ij} \cdot \nabla f_i \right) V_j, \quad (13)$$

where “ $\otimes$ ” represents the dyadic product of the two vectors; “ $:$ ” represents the double dot-product of two matrices;  $\mathbf{G}_i$  and  $\mathbf{L}_i$ , defined differently from  $\mathbf{G}(\epsilon_B)$  and  $\mathbf{L}(\mathbf{q})$  in Eqs. (1) and (2a)–(2e), are second-order and symmetric correction tensors for SPH marker  $i$ . The  $(m, n)$  element of the inverse of  $\mathbf{G}_i$  is expressed as [21, 39, 61]:

$$(\mathbf{G}_i^{-1})^{mn} = - \sum_j r_{ij}^m \nabla_{i,n} W_{ij} V_j. \quad (14)$$

The  $3 \times 3$  matrix  $\mathbf{L}_i$  is symmetric and has six unknowns obtained as the solution of a linear system [21]. The required six independent equations may be obtained by expanding the following equation for the upper/lower triangular elements of a  $3 \times 3$  matrix, e.g.,  $m = 1, n = 1, 2, 3, m = 2, n = 2, 3$ , and  $m = 3, n = 3$ ,

$$-\delta^{mn} = \sum_j (A_i^{kmn} e_{ij}^k + r_{ij}^m e_{ij}^n) (L_i^{op} e_{ij}^o \nabla_{i,p} W_{ij} V_j), \quad (15)$$

where  $\delta^{mn}$  is the Kronecker symbol, and the elements of the third-order tensor  $A_i$  are obtained as

$$A_i^{kmn} = \sum_j r_{ij}^m r_{ij}^n G_i^{kq} \nabla_{i,q} W_{ij} V_j. \quad (16)$$

A detailed account of obtaining the elements  $\mathbf{L}_i$  is provided in [29].

The outcome of the SPH discretization steps described above can be conveniently represented in matrix form. To that end, at the beginning of a time step one computes and stores the discretization matrices  $\mathbf{A}^G$  and  $\mathbf{A}^L$ , which are defined differently from the ones used in Sect. 2.1;  $\mathbf{A}^G$  and  $\mathbf{A}^L$  arise from either the standard discretization Eqs. (9)–(10), or the consistent discretization of Eqs. (12)–(13). For instance, working with Eq. (10),  $\langle \nabla^2 f \rangle_i$  can be expressed as follows:

$$\begin{aligned} \langle \nabla^2 f \rangle_i &= \mathbf{A}_i^L \mathbf{f}, \\ \mathbf{f} &= [f_1, \quad f_2, \quad \dots, \quad f_P]^T, \\ \mathbf{A}_i^L &= \left[ \dots, \quad \underbrace{2 \sum_{j \in \mathcal{S}(i)} V_j (\mathbf{e}_{ij} \cdot \nabla_i W_{ij}) \frac{1}{|\mathbf{r}_{ij}|}}_{i\text{th element}}, \quad \dots, \quad \underbrace{-2 V_j (\mathbf{e}_{ij} \cdot \nabla_i W_{ij}) \frac{1}{|\mathbf{r}_{ij}|}}_{j\text{th element s.t. } j \in \mathcal{S}(i)}, \quad \dots \right], \end{aligned}$$

where the subscript  $P$  denotes the number of SPH particles in the domain. Similarly, the gradient of a scalar field  $\langle \nabla f \rangle_i$  and divergence of a vector field  $\langle \nabla \cdot \mathbf{u} \rangle_i$  may be computed from Eq. (9) as follows:

$$\langle \nabla f \rangle_i = \begin{bmatrix} \mathbf{A}_i^{Gx} \\ \mathbf{A}_i^{Gy} \\ \mathbf{A}_i^{Gz} \end{bmatrix} \mathbf{f}, \quad (17)$$

$$\langle \nabla \cdot \mathbf{u} \rangle_i = \mathbf{A}_i^{Gx} \mathbf{u}_x + \mathbf{A}_i^{Gy} \mathbf{u}_y + \mathbf{A}_i^{Gz} \mathbf{u}_z, \quad (18)$$

$$\mathbf{f} = [f_1, f_2, \dots, f_P]^T, \quad (19)$$

$$\mathbf{u}_x = [(u_x)_1, (u_x)_2, \dots, (u_x)_P]^T, \quad (20)$$

$$\mathbf{u}_y = [(u_y)_1, (u_y)_2, \dots, (u_y)_P]^T, \quad (21)$$

$$\mathbf{u}_z = [(u_z)_1, (u_z)_2, \dots, (u_z)_P]^T, \quad (22)$$

where

$$\mathbf{A}_i^{Gx} = [\dots, -\sum_{j \in S(i)} V_j \nabla_{i,1} W_{ij}, \dots, V_j \nabla_{i,1} W_{ij}, \dots], \quad (23)$$

$$\mathbf{A}_i^{Gy} = [\dots, -\sum_{j \in S(i)} V_j \nabla_{i,2} W_{ij}, \dots, V_j \nabla_{i,2} W_{ij}, \dots], \quad (24)$$

$$\mathbf{A}_i^{Gz} = [\dots, -\sum_{j \in S(i)} V_j \nabla_{i,3} W_{ij}, \dots, V_j \nabla_{i,3} W_{ij}, \dots], \quad (25)$$

$$\mathbf{A}_i^G = [\dots, \underbrace{-\sum_{j \in S(i)} V_j \nabla_i W_{ij}}_{i\text{th element}}, \dots, \underbrace{V_j \nabla_i W_{ij}}_{j\text{th element s.t. } j \in S(i)}, \dots]. \quad (26)$$

The same approach can be used to obtain the discretization matrices for the consistent discretization of Eqs. (12)–(13). The system level matrices  $\mathbf{A}^G$  and  $\mathbf{A}^L$  are obtained as:

$$\langle \nabla f \rangle^x = \begin{bmatrix} \langle \nabla f \rangle_1^x \\ \langle \nabla f \rangle_2^x \\ \vdots \\ \langle \nabla f \rangle_P^x \end{bmatrix} = \mathbf{A}^{Gx} \mathbf{f}, \quad \langle \nabla f \rangle^y = \begin{bmatrix} \langle \nabla f \rangle_1^y \\ \langle \nabla f \rangle_2^y \\ \vdots \\ \langle \nabla f \rangle_P^y \end{bmatrix} = \mathbf{A}^{Gy} \mathbf{f}, \quad \langle \nabla f \rangle^z = \begin{bmatrix} \langle \nabla f \rangle_1^z \\ \langle \nabla f \rangle_2^z \\ \vdots \\ \langle \nabla f \rangle_P^z \end{bmatrix} = \mathbf{A}^{Gz} \mathbf{f}, \quad (27)$$

$$\langle \nabla \cdot \mathbf{u} \rangle = [\langle \nabla \cdot \mathbf{u} \rangle_1, \langle \nabla \cdot \mathbf{u} \rangle_2, \dots, \langle \nabla \cdot \mathbf{u} \rangle_P]^T = \mathbf{A}^{Gx} \mathbf{u}_x + \mathbf{A}^{Gy} \mathbf{u}_y + \mathbf{A}^{Gz} \mathbf{u}_z, \quad (28)$$

$$\mathbf{A}^{Gx} = \begin{bmatrix} \mathbf{A}_1^{Gx} \\ \mathbf{A}_2^{Gx} \\ \vdots \\ \mathbf{A}_P^{Gx} \end{bmatrix}, \quad \mathbf{A}^{Gy} = \begin{bmatrix} \mathbf{A}_1^{Gy} \\ \mathbf{A}_2^{Gy} \\ \vdots \\ \mathbf{A}_P^{Gy} \end{bmatrix}, \quad \mathbf{A}^{Gz} = \begin{bmatrix} \mathbf{A}_1^{Gz} \\ \mathbf{A}_2^{Gz} \\ \vdots \\ \mathbf{A}_P^{Gz} \end{bmatrix}, \quad (29)$$

$$\langle \nabla^2 f \rangle = \mathbf{A}^L \mathbf{f}, \quad (30)$$



$$\langle \nabla^2 f \rangle = [\langle \nabla^2 f \rangle_1, \quad \langle \nabla^2 f \rangle_2, \quad \dots, \quad \langle \nabla^2 f \rangle_P]^T, \quad (31)$$

$$\mathbf{A}^L = \begin{bmatrix} \mathbf{A}_1^L \\ \mathbf{A}_2^L \\ \vdots \\ \mathbf{A}_P^L \end{bmatrix}. \quad (32)$$

This allows for the space discretization of the Navier–Stokes equations in the  $x$ ,  $y$ , and  $z$  directions, per Eq. (6),

$$\begin{cases} \frac{d\mathbf{u}_x}{dt} \approx -\frac{1}{\rho} \mathbf{A}^{Gx} \mathbf{p} + \nu \mathbf{A}^L \mathbf{u}_x + f_x^b, \\ \frac{d\mathbf{u}_y}{dt} \approx -\frac{1}{\rho} \mathbf{A}^{Gy} \mathbf{p} + \nu \mathbf{A}^L \mathbf{u}_y + f_y^b, \\ \frac{d\mathbf{u}_z}{dt} \approx -\frac{1}{\rho} \mathbf{A}^{Gz} \mathbf{p} + \nu \mathbf{A}^L \mathbf{u}_z + f_z^b, \end{cases} \quad (33)$$

where

$$\mathbf{p} = [p_1, \quad p_2, \quad \dots, \quad p_P]^T, \quad (34)$$

is the vector of pressures, and  $\mathbf{u}_x$ ,  $\mathbf{u}_y$ , and  $\mathbf{u}_z$  were defined in Eq. (22).

### 3.1 The KCSPH algorithm

The KCSPH draws on a thermodynamically consistent SPH discretization [20] that leads to an index-3 set of differential algebraic equations (see Sect. 2.1) describing the motion of the fluid markers. The SPH particles can be regarded as 3 degree-of-freedom point-masses constrained in their motion. Collectively, these kinematic constraints capture the incompressibility of the fluid and couple the relative motion of the SPH markers. Notably, the effect of these constraints comes into the momentum equations as well. Indeed, the Lagrange multiplier forcing term that arises from the compressibility constraint acts as the pressure gradient term in the momentum balance equations [11, 19, 20, 44].

The cornerstones of WSPH and ISPH were, respectively, the use of a stiff state equation for recovering the pressure, and the use of a Poisson equation to produce a pressure field that enforces incompressibility. In KCSPH, the defining aspect is the use of holonomic kinematic constraints to enforce incompressibility; at the location of SPH marker  $i$ , the density should assume a reference value  $\rho_0$ :

$$C_i^f = \frac{\rho_i - \rho_0}{\rho_0} = 0. \quad (35)$$

If the time-derivative of the constraints in Eq. (35) is satisfied at the velocity level, i.e.,  $\dot{C}_i^f = d\rho/dt = 0$ , the following will emerge after applying Eq. (7), i.e., invoking the SPH machinery:

$$\begin{aligned} \dot{C}_i^f &= \frac{d}{dt} \left( \frac{\rho_i}{\rho_0} - 1 \right) \approx \sum_j \frac{m_j}{\rho_0} \frac{dW_{ij}}{dt} = \sum_j \frac{m_j}{\rho_0} \frac{dW_{ij}}{d\mathbf{r}_{ij}} \frac{d\mathbf{r}_{ij}}{dt} \\ &= \sum_j \frac{m_j}{\rho_0} \nabla_i W_{ij} (\mathbf{u}_i - \mathbf{u}_j) \\ &= - \sum_j \frac{m_j}{\rho_0} \nabla_i W_{ij} (\mathbf{u}_j - \mathbf{u}_i). \end{aligned} \quad (36)$$

Above, if  $\rho_j \approx \rho_0$ , the last term mimics  $-(\nabla \cdot \mathbf{u})_i$ , see Eq. (9). Elements in row  $i$  of the constraint Jacobian matrix  $\nabla_{\mathbf{q}} \mathbf{g}(\mathbf{q}, t) \equiv \mathbf{G}$  (see Eq. (2b)) are associated with the constraint  $g_i = \dot{C}_i^f = 0$ . These entries in  $\mathbf{G}$  may be obtained via Eq. (36) as follows:

$$\begin{aligned} \dot{C}_i^f &= \sum_j \frac{m_j}{\rho_0} \nabla_i W_{ij} \mathbf{u}_i - \sum_j \frac{m_j}{\rho_0} \nabla_i W_{ij} \mathbf{u}_j \quad \Rightarrow \\ G_{ii} &= \frac{1}{\rho_0} \sum_{k \neq i} m_k \nabla W_{ik} \quad \text{and} \quad G_{ij} = -\frac{m_j}{\rho_0} \nabla_i W_{ij}. \end{aligned} \quad (37)$$

Each density constraint on a marker contributes to a single row in the full Jacobian matrix which has  $3P$  columns. This matrix is sparse and its rows have three values at the columns corresponding to the current marker  $i$  and three values for each marker  $j$  within the support domain of  $i$ . More specifically, for row  $i$ ,

$$\mathbf{G}_i = \left[ \dots, \underbrace{\frac{1}{\rho_0} \sum_{j \in \mathcal{S}(i)} m_k \nabla_i W_{ik}^T}_{i\text{th element}}, \dots, \underbrace{-\frac{m_j}{\rho_0} \nabla_i W_{ij}^T}_{j\text{th element s.t. } j \in \mathcal{S}(i)}, \dots \right]_{1 \times 3P}. \quad (38)$$

It is informative to compare Eqs. (38) and (26) to underline similarities between the constraint Jacobian row  $\mathbf{G}_i$  and the discretized gradient operator  $\mathbf{A}_i^G$ . Qualitatively, the density constraint Jacobian matrix is similar to the discretized gradient operator – satisfying  $\mathbf{G}\mathbf{u} = 0$  is the analog of  $\nabla \cdot \mathbf{u} = \mathbf{A}^{Gx} \mathbf{u}_x + \mathbf{A}^{Gy} \mathbf{u}_y + \mathbf{A}^{Gz} \mathbf{u}_z = 0$  (see Eq. (18)). This connection between  $\mathbf{G}$  and  $\mathbf{A}^G$  comes further into play when one considers how the pressure factors into the Newton–Euler equations of motion. Imposing the kinematic constraint  $\dot{C}^f = \mathbf{G}\dot{\mathbf{q}} = 0$  leads to the presence of a Lagrange multiplier. The multipliers in Eq. (2b) play a role similar to that of the pressures. This becomes clearer when the force term associated with the Lagrange multipliers in Eq. (2b), i.e.,  $(\nabla_{\mathbf{q}} \mathbf{g}(\mathbf{q}, t))^T \Lambda$  is written as  $\mathbf{G}^T \Lambda$ , where the connection between the  $\mathbf{G}$  matrix and the discretized gradient matrix  $\mathbf{A}^G$  is considered. It follows that KCSPH replaces the pressure gradient term  $-\frac{1}{\rho} \mathbf{A}^G \mathbf{p}$  in the space-discretized Navier–Stokes (Eq. (33)) with  $\mathbf{G}^T \Lambda$ . Considering the resemblance of  $\mathbf{A}^G$  and the  $-\mathbf{G}$  matrices, we conclude that the Lagrange multipliers scaled (element-wise) by the particles' volume ( $\Lambda/V$ ) provide the mechanical pressure  $\mathbf{p}$  in the Navier–Stokes equations.

Note also that the derivation of the constraint Jacobian in the KCSPH formulation (Eq. (36)) leads to the conservative discretization discussed in Eq. (9), which as discussed before lacks consistency. Up to this point, the KCSPH discussion focused exclusively on obtaining the SPH equations of motion in the form of Sect. 2.1's differential-algebraic equations. In other words, the spatial discretization step has been accomplished. How to discretize and solve in time these differential algebraic equations goes beyond the scope of this manuscript and the interested reader is referred to [44].

### 3.2 The WSPH algorithm

The cornerstone of WSPH is its use of a state equation to obtain the pressure from density: at particle/marker  $i$ ,

$$p_i = (k|\mathbf{u}|_{\max})^2 \left( \frac{\rho_i}{\rho_0} - 1 \right) + p_0, \quad (39)$$

where  $k|\mathbf{u}|_{\max}$  is a sound speed proxy;  $|\mathbf{u}|_{\max}$  is the magnitude of the maximum velocity in the domain;  $k = 10$  is an empirical scaling factor; and  $\rho_0$  and  $p_0$  are reference values. The

density update may be obtained from the time integration of the continuity equation Eq. (4) using the space discretization of the velocity divergence from Eq. (28)

$$\frac{\rho^{n+1} - \rho^n}{\Delta t} = - \langle \rho^n \cdot (\mathbf{A}^{Gx} \mathbf{u}_x + \mathbf{A}^{Gy} \mathbf{u}_y + \mathbf{A}^{Gz} \mathbf{u}_z) \rangle, \quad (40)$$

where  $\rho^n \equiv [\rho_0, \rho_1, \dots, \rho_P]^T$  is the system level vector of markers' density at the current time step, and  $\mathbf{c} = \langle \mathbf{a} \cdot \mathbf{b} \rangle$  indicates the element-wise product of vector  $\mathbf{a}$  and  $\mathbf{b}$ . Shepard filtering [15] may be periodically applied when dealing with free surface problems for density reinitialization. The time integration used is of predictor–corrector type:

$$\begin{aligned} \text{(predictor stage)} \quad & \begin{cases} \bar{\mathbf{u}}^{n+1/2} = \mathbf{u}^n + \frac{\Delta t}{2} \mathbf{a}^n, \\ \bar{\mathbf{r}}^{n+1/2} = \mathbf{r}^n + \frac{\Delta t}{2} \mathbf{u}^n, \end{cases} \\ \text{(corrector stage)} \quad & \begin{cases} \mathbf{u}^{n+1/2} = \mathbf{u}^n + \frac{\Delta t}{2} \bar{\mathbf{a}}^{n+1/2}, \\ \mathbf{r}^{n+1/2} = \mathbf{r}^n + \frac{\Delta t}{2} \mathbf{u}^{n+1/2}. \end{cases} \end{aligned}$$

Finally,

$$\text{(update stage)} \quad \begin{cases} \mathbf{u}^{n+1} = 2\mathbf{u}^{n+1/2} - \mathbf{u}^n, \\ \mathbf{r}^{n+1} = 2\mathbf{r}^{n+1/2} - \mathbf{r}^n. \end{cases}$$

Above,  $\mathbf{a}^n$  is the system-level vector of accelerations at time step  $n$ ; its components in the  $x$ ,  $y$ ,  $z$  directions are obtained from the space discretization of the Navier–Stokes as:

$$\begin{cases} (\mathbf{a}^x)^n = -\frac{1}{\rho} (\mathbf{A}^{Gx})^n \mathbf{p}^n + \nu (\mathbf{A}^L)^n \mathbf{u}_x^n + f_x^b, \\ (\mathbf{a}^y)^n = -\frac{1}{\rho} (\mathbf{A}^{Gy})^n \mathbf{p}^n + \nu (\mathbf{A}^L)^n \mathbf{u}_y^n + f_y^b, \\ (\mathbf{a}^z)^n = -\frac{1}{\rho} (\mathbf{A}^{Gz})^n \mathbf{p}^n + \nu (\mathbf{A}^L)^n \mathbf{u}_z^n + f_z^b. \end{cases}$$

Likewise,  $\bar{\mathbf{a}}^{n+1/2}$  in the corrector stage is obtained using  $\bar{\mathbf{x}}^{n+1/2}$ ,  $\bar{\mathbf{u}}^{n+1/2}$ , and the associated discrete representation of the gradient and Laplacian operators, i.e.,  $(\bar{\mathbf{A}}^G)^{n+1/2}$  and  $(\bar{\mathbf{A}}^L)^{n+1/2}$ . As far as the time step  $\Delta t$  is concerned, its size is constrained on numerical stability grounds by the following condition [40]:

$$\Delta t \leq \min \left\{ 0.25 \frac{h}{k|\mathbf{u}|_{\max}}, \quad 0.125 \frac{h^2}{\nu}, \quad 0.25 \sqrt{\frac{h}{|\mathbf{f}_b|}} \right\}. \quad (41)$$

Above, the first term corresponds to the CFL condition and is the place where the “numerical stiffness” of the state equation in Eq. (39) comes into play. Specifically, the higher the  $k$  value, the lower the amount of allowed compressibility, and, at the same time, the smaller the time step. The second restriction in Eq. (41) appears due to the explicit treatment of the viscous term and restricts the time step by a factor that is inversely proportional to the viscosity – the higher the viscosity, the lower the time step. More importantly, the second restriction is also proportional to  $h^2$ , which significantly and adversely impacts the time step when a finer particle distribution is employed. The last restriction is due to the explicit treatment of the external (body) forces. Ultimately, these relatively stringent bounds on the time step  $\Delta t$  prompted the search for alternative SPH-based approaches, e.g., ISPH and KCSPH.

### 3.3 The ISPH algorithm

ISPH alleviates the time step constraints hindering WCSPH at the price of solving a linear system of equations at each time step. It draws on the Helmholtz–Hodge decomposition and Chorin’s projection method [13] to integrate the continuity and the Navier–Stokes equation as:

$$\text{(prediction)} \quad \begin{cases} \frac{(\mathbf{u}^* - \mathbf{u}^n)}{\Delta t} = \frac{\nu}{2}(\nabla^2 \mathbf{u}^* + \nabla^2 \mathbf{u}^n) + \mathbf{f}^b, & x \in \Omega, \\ \mathbf{u}^* = 0, & x \in \partial\Omega, \end{cases} \quad (42)$$

$$\text{(correction)} \quad \begin{cases} \frac{(\mathbf{u}^{n+1} - \mathbf{u}^*)}{\Delta t} = -\frac{1}{\rho} \nabla p^{n+1}, & x \in \Omega, \\ \nabla \cdot \mathbf{u}^{n+1} = 0. \end{cases} \quad (43)$$

Equation (42) is the predictor step used to find the intermediate velocity  $\mathbf{u}^*$ . Taking divergence of Eq. (43), the Poisson equation for pressure may be obtained as follows:

$$\frac{\nabla \cdot \mathbf{u}^{n+1} - \nabla \cdot \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho} \nabla^2 p^{n+1}. \quad (44)$$

The incompressible flow equation,  $\nabla \cdot \mathbf{u}^{n+1} = 0$ , can be used to simplify Eq. (44) to

$$\begin{cases} \frac{1}{\rho} \nabla^2 p^{n+1} = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t}, \\ \nabla p^{n+1} \cdot \mathbf{n}|_{\partial\Omega} = 0. \end{cases} \quad (45)$$

Once the above Poisson problem is solved for pressure, Eq. (43) may be used to find  $\mathbf{u}^{n+1}$  as

$$\mathbf{u}^{n+1} = -\Delta t \frac{1}{\rho} \nabla p^{n+1} + \mathbf{u}^*.$$

The algorithm described above is known as the *velocity-based projection* version. However, when working with free-surface flows, it is beneficial to use a *density-based projection* method [8, 64] to account for the density variation experienced by the SPH particles in the proximity of the free surface. The continuity equation is then used to replace the velocity divergence term  $\frac{\nabla \cdot \mathbf{u}^*}{\Delta t}$  in Eq. (45). The semidiscrete continuity equation corresponding to the prediction state is

$$\frac{\rho^* - \rho^n}{\Delta t} = -\rho^n \nabla \cdot \mathbf{u}^*. \quad (46)$$

Using the right-hand side of Eq. (46), one may write Eq. (45) as

$$\begin{cases} \frac{1}{\rho} \nabla^2 p^{n+1} = -\frac{1}{\rho^n} \frac{\rho^* - \rho^n}{\Delta t^2}, \\ \nabla p^{n+1} \cdot \mathbf{n}|_{\partial\Omega} = 0, \end{cases} \quad (47)$$

which takes into consideration the density variation as a source term in the Poisson equation. Following an approach similar to that introduced in [8], we use a stabilization of the source

term in the Poisson equation to linearly combine the right-hand sides of Eqs. (45) and (47):

$$\text{(pressure equation)} \quad \begin{cases} \frac{1}{\rho} \nabla^2 p^{n+1} = \alpha \frac{1}{\rho^n} \frac{\rho^n - \rho^*}{\Delta t^2} + (1 - \alpha) \frac{\nabla \cdot \mathbf{u}^*}{\Delta t}, \\ \nabla p^{n+1} \cdot \mathbf{n}|_{\partial\Omega} = 0. \end{cases} \quad (48)$$

This stabilization technique turned out to be critical in simulations where the difference between the actual  $\rho$  and reference density  $\rho_0$  is large. Typical  $\alpha$  values are small, similar to the order of magnitude of the kernel characteristic length.

The above time-discretized equations may be combined with the space-discretization of Eqs. (27) and (30). The space-time discretized ISPH equations are as follows:

$$\begin{cases} \left( \frac{1}{\Delta t} \mathbf{I} - \frac{\nu}{2} \mathbf{A}^L \right) \mathbf{u}_x^* = \left( \frac{1}{\Delta t} \mathbf{I} + \frac{\nu}{2} \mathbf{A}^L \right) \mathbf{u}_x^n + f_x^b, & \text{particles} \in \Omega, \\ \left( \frac{1}{\Delta t} \mathbf{I} - \frac{\nu}{2} \mathbf{A}^L \right) \mathbf{u}_y^* = \left( \frac{1}{\Delta t} \mathbf{I} + \frac{\nu}{2} \mathbf{A}^L \right) \mathbf{u}_y^n + f_y^b, & \text{particles} \in \Omega, \\ \left( \frac{1}{\Delta t} \mathbf{I} - \frac{\nu}{2} \mathbf{A}^L \right) \mathbf{u}_z^* = \left( \frac{1}{\Delta t} \mathbf{I} + \frac{\nu}{2} \mathbf{A}^L \right) \mathbf{u}_z^n + f_z^b, & \text{particles} \in \Omega, \\ \mathbf{u}^* = 0 & \text{on } \partial\Omega, \end{cases} \quad (49)$$

$$\begin{cases} \frac{1}{\rho} \mathbf{A}^L \mathbf{p}^{n+1} = \alpha \frac{1}{\rho^n} \frac{\rho^n - \rho^*}{\Delta t^2} + (1 - \alpha) \frac{\mathbf{A}^{Gx} \mathbf{u}_x^* + \mathbf{A}^{Gy} \mathbf{u}_y^* + \mathbf{A}^{Gz} \mathbf{u}_z^*}{\Delta t}, \\ \nabla p^{n+1} \cdot \mathbf{n}|_{\partial\Omega} = 0, \end{cases} \quad (50)$$

$$\begin{cases} \frac{(\mathbf{u}_x^{n+1} - \mathbf{u}_x^*)}{\Delta t} = -\frac{1}{\rho} \mathbf{A}^{Gx} \mathbf{p}^{n+1}, \\ \frac{(\mathbf{u}_y^{n+1} - \mathbf{u}_y^*)}{\Delta t} = -\frac{1}{\rho} \mathbf{A}^{Gy} \mathbf{p}^{n+1}, \\ \frac{(\mathbf{u}_z^{n+1} - \mathbf{u}_z^*)}{\Delta t} = -\frac{1}{\rho} \mathbf{A}^{Gz} \mathbf{p}^{n+1}. \end{cases} \quad (51)$$

Several key observations pertaining the ISPH method implemented are summarized as follows:

- The Crank–Nicolson discretization of the viscous term in Eq. (42) leads to a nondiagonal coefficient matrix  $(\frac{1}{\Delta t} \mathbf{I} - \frac{\nu}{2} \mathbf{A}^L)$  in Eq. (49). Had one chosen to treat the viscous term in Eq. (42) explicitly, i.e.,  $\nu \nabla^2 \mathbf{u}^n$  instead of  $\frac{\nu}{2} (\nabla^2 \mathbf{u}^* + \nabla^2 \mathbf{u}^n)$ , the coefficient matrix of the linear system in Eq. (49) would have become  $\frac{1}{\Delta t} \mathbf{I}$ , i.e., a diagonal matrix. Yet, this choice that makes the linear solving trivial, would constrain the time step  $\Delta t$  owing to the explicit treatment of the viscous term, which imposes the time-step restriction of  $\Delta t < 0.125 \frac{h^2}{\nu}$ , see Eq. (41).
- The  $x$ ,  $y$ , and  $z$  directions in Eq. (49) use the same coefficient matrix, which simplifies the implementation.
- A modification that proved particularly useful at small  $\Delta t$  pertains a scaling of the pressure by a factor  $\Delta t^2$  in the Poisson equation, and by  $1/\Delta t^2$  in the correction step Eq. (51). In other words, we compute  $p \Delta t^2$  when solving the Poisson equation, and subsequently scale the pressure in the correction step Eq. (51) by a factor of  $1/\Delta t^2$ .

### 3.4 Handling of boundary conditions

Herein, imposing boundary conditions (BCs) falls back on the use of so-called *boundary conditions enforcing* (BCE) markers. These are fictitious markers, rigidly attached to the boundary in a buffer zone that runs several layers of BCE markers deep. These markers are used to enforce the no-slip and no-penetration conditions, i.e., to satisfy the  $\mathbf{u}^* = \mathbf{0}$  condition on the boundary. This condition is implemented differently in WCSPH, ISPH and KCSPH.

In one widely used approach [2], the *expected* kinematic attributes of the markers, calculated from the motion of the solid phase at the location occupied by the markers, are different from their *assigned* values. The latter are calculated such that the no-slip and no-penetration boundary conditions are implicitly enforced at the fluid–solid interface. The no-slip condition states that the velocity of the BCE markers should oppose the velocity of the fluid particles such that the average *relative* fluid–solid velocity at the interface is zero, i.e., the average velocity at the interface is the *expected* interface velocity. Herein, the *induced* velocity  $\tilde{\mathbf{u}}_a$  at the position of marker  $a$  is computed from the velocity of the fluid markers as

$$\tilde{\mathbf{u}}_a = \frac{\sum_{b \in \mathbf{F}} \mathbf{u}_b W_{ab}}{\sum_{b \in \mathbf{F}} W_{ab}}, \quad (52)$$

where  $\mathbf{F}$  denotes a set of fluid markers that are within the compact support of the BCE marker  $a$ . The no-slip condition holds if  $(\tilde{\mathbf{u}}_a + \mathbf{u}_a)/2 = \mathbf{u}_a^p$ ; in other words, the *assigned* velocity of marker  $a$  is [2]:

$$\mathbf{u}_a = 2\mathbf{u}_a^p - \tilde{\mathbf{u}}_a, \quad (53)$$

where  $\mathbf{u}_a^p$  is the *expected* wall velocity at the position of the marker  $a$ , and  $\tilde{\mathbf{u}}_a$  is an extrapolation of the smoothed velocity field of the fluid phase to the BCE markers.

The pressure of a BCE marker may be calculated via a force balance condition at the wall interface, which leads to [2]

$$p_a = \frac{\sum_{b \in \mathbf{F}} p_b W_{ab} + (\mathbf{g} - \mathbf{a}_a^p) \cdot \sum_{b \in \mathbf{F}} \rho_b \mathbf{r}_{ab} W_{ab}}{\sum_{b \in \mathbf{F}} W_{ab}}, \quad (54)$$

where  $\mathbf{g}$  is the gravitational acceleration and  $\mathbf{a}_a^p$  is the acceleration of the boundary/solid at the location of BCE marker  $a$ .

Note that the boundary condition treatment discussed above is based on a force balance argument and is conservative but not consistent. One may rely on interpolation methods discussed in [30] to retain consistency close to the boundaries. This approach is fairly straightforward to implement in WCSPH – one can modify the velocity and pressure of the BCE markers according to Eqs. (53) and (54), respectively. In regard to the ISPH method, (i) the no-slip boundary condition is implemented in the linear system of Eq. (49); and, (ii) pressure boundary conditions should be incorporated into the linear system in Eq. (51). In regard to (i), the modified row of the linear system associated with the boundary marker  $a$  for the velocity equation Eq. (49) is

$$\mathbf{A}_a^v \mathbf{u}_x = 2(u_x^p)_a \sum_{b \in \mathbf{F}} W_{ab}, \quad \mathbf{A}_a^v \mathbf{u}_y = 2(u_y^p)_a \sum_{b \in \mathbf{F}} W_{ab}, \quad \mathbf{A}_a^v \mathbf{u}_z = 2(u_z^p)_a \sum_{b \in \mathbf{F}} W_{ab}, \quad (55)$$

$$\mathbf{A}_a^v = \left[ \dots, \underbrace{\sum_{b \in \mathbf{F}} W_{ab}}_{a\text{th element}}, \dots, \underbrace{W_{ab}}_{b\text{th element s.t. } b \in \mathbf{F} \text{ and } \in \mathcal{S}(a)}, \dots \right]. \quad (56)$$

In regard to (ii), the interplay between Eqs. (50) and (54) leads to

$$\mathbf{A}_a^p \mathbf{p} = (\mathbf{g} - \mathbf{a}_a^p) \cdot \sum_{b \in \mathbf{F}} \rho_b \mathbf{r}_{ab} W_{ab}, \quad (57)$$

$$\mathbf{A}_a^p = \left[ \dots, \underbrace{\sum_{b \in \mathbf{F}} W_{ab}}_{a\text{th element}}, \dots, \underbrace{-W_{ab}}_{b\text{th element s.t. } b \in \mathbf{F} \text{ and } \in \mathcal{S}(a)}, \dots \right]. \quad (58)$$

Note that instead of Eq. (54), one can enforce at the boundary the condition that the pressure satisfy  $\nabla p^{n+1} \cdot \mathbf{n}|_{\partial\Omega} = 0$ , which mimics the traditional Eulerian handling of pressure boundary conditions. Insofar as the discretized pressure equation is then concerned, the row in the linear system in Eq. (50) that corresponds to boundary particle  $a$  will read

$$\begin{aligned} \mathbf{A}_a^p \mathbf{p} &= 0, \\ \mathbf{A}_a^p &= \left[ \dots, \underbrace{\sum_{b \in \mathbf{F}} \mathbf{A}_{ab}^G \cdot \mathbf{n}_a}_{a\text{th element}}, \dots, \underbrace{\mathbf{A}_{ab}^G \cdot \mathbf{n}_a}_{b\text{th element s.t. } b \in \mathbf{F} \text{ and } \in \mathcal{S}(a)}, \dots \right]. \end{aligned}$$

Above,  $\mathbf{A}_{ab}^G \in \mathcal{R}^3$  is the  $b$ th element of discretized gradient matrix  $\mathbf{A}_a^G$  (see Eq. (26)),  $\mathbf{n}_a$  is the surface normal vector at the position of particle  $a$ , and  $\mathbf{p}$  was defined in Eq. (34).

The fluid–structure forces on the boundary markers may be obtained from the pressure and viscous forces acting on a boundary marker  $a$  according to the momentum balance equations as follows:

$$\mathbf{F}_a = m_a \frac{d\mathbf{u}_a}{dt} = m_a (v \nabla^2 \mathbf{u}_a^{n+1} - \frac{1}{\rho} \nabla p_a^{n+1}). \quad (59)$$

In the discrete sense, similar expressions as the ones discussed in Eq. (33) may be applied to the right-hand side terms to obtain the contributions of the fluid domain on the structure.

Finally, KCSPH handles boundary conditions via multibody dynamics techniques [44]. The no-penetration condition between SPH markers and boundaries is enforced just as in multibody dynamics, i.e., via unilateral constraints (see Eqs. (2d)–(2e)). This strategy allows for (i) strict (per SPH particle) enforcement of the no-penetration, and (ii) more flexibility in terms of the geometry of the boundary. On the downside, in the current implementation this strategy can only be used to enforce the *no-penetration* condition but not the *no-slip*, which is unlike WCSPH and ISPH. In its current implementation, this limits the KCSPH approach to problems where the boundary layer effects are negligible compared to the pressure gradient and the inertia terms.

### 3.5 Particle shifting

The advection of SPH particles can lead to scenarios characterized by high particle disorder and/or regions with high particle depletion/plenitude. Maintaining the accuracy and stability of the SPH method under these circumstances requires mitigating measures, one being particle shifting. The latter calls for retiring a particle only to introduce it back in a consistent fashion, slightly away from the previous location and the streamlines in order to improve the

uniformity of the SPH particle distribution. Particle shifting in conjunction with an ISPH-type implementation was used in [77] and subsequently reported to be effective in [72]. It has also been used in conjunction with a WCSPH-class implementation in [63]. The shifting vector is computed for particle  $i$  as:

$$\delta \mathbf{r}_i = \beta r_0^2 u_{\max} \Delta t \sum_{j \in \mathcal{S}(i)} \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^3},$$

where  $r_0 = \frac{1}{N_i} \sum_j |\mathbf{r}_{ij}|$ ,  $u_{\max}$  is the maximum velocity in the domain,  $\Delta t$  is the integration time step, and  $\beta$  is an adjustable dimensionless parameter determining the magnitude of the shifting vector. At the end of each time step the position of particle  $i$  is shifted by  $\mathbf{x}_i^{\text{new}} = \mathbf{x}_i + \delta \mathbf{r}_i$ . Accordingly, the field variable  $\rho_i$ ,  $p_i$ , and  $\mathbf{u}_i$  are updated as:

$$p_i^{\text{new}} = p_i + \nabla p_i \cdot \delta \mathbf{r}_i, \quad \rho_i^{\text{new}} = \rho_i + \nabla \rho_i \cdot \delta \mathbf{r}_i, \quad \text{and} \quad \mathbf{u}_i^{\text{new}} = \mathbf{u}_i + \nabla \mathbf{u}_i \cdot \delta \mathbf{r}_i.$$

## 4 Numerical experiments

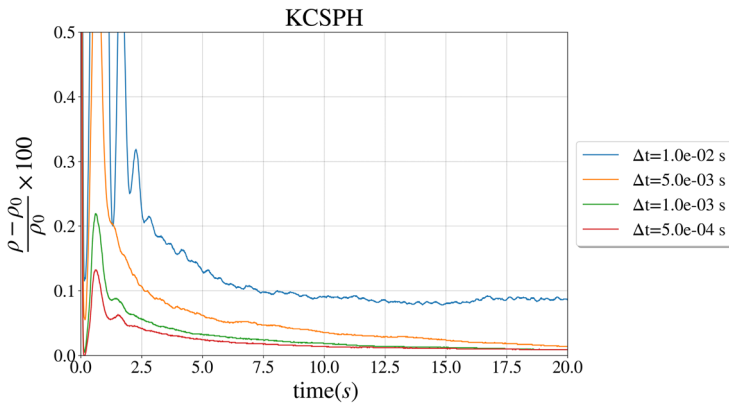
The objective of this contribution is to compare KCSPH, a multibody dynamics-based solution, against two established CFD methods, WCSPH and ISPH. To that end, two sets of numerical experiments were carried out in this study: a first set was concerned with ensuring that our WCSPH and ISPH implementations were sound. This set of experiments is of lesser relevance; as such, it was moved into the [Appendix](#). Therein, WCSPH and ISPH are validated in the context of a Poiseuille flow in [Appendix A.1](#); and for a flow around a cylinder in [Appendix A.2](#). This section reports on the second set of numerical experiments, in which we compare KCPSH against WCSPH and ISPH in conjunction with four tests chosen to probe complementary aspects of the CFD solution. A “fluid-at-rest” test was chosen to gauge effectiveness in enforcing incompressibility; the dam break probed the effectiveness of each solution for a free-surface problem that displays high transients; the floating cylinder problem was selected to assess the ability of WCSPH, ISPH, and KCSPH to handle fluid–solid interaction phenomena; and small amplitude sloshing was chosen since it had an analytical solution.

We use the standard SPH discretizations without particle shifting for the compressibility test, the dam break, and the sloshing experiments. This is mainly due to the limitation of the standard particle shifting method for free-surface problems. We use particle shifting in ISPH and WCSPH for the Poiseuille flow and the flow around cylinder experiments in order to maintain regular particle distribution. Finally, consistent discretizations are used in ISPH for the Poiseuille flow and the flow around cylinder experiments to reduce the number of neighbors and the computational costs of the pressure solver. All simulations were run in 3D using an open source code called Chrono [56] that is publicly available on GitHub [57].

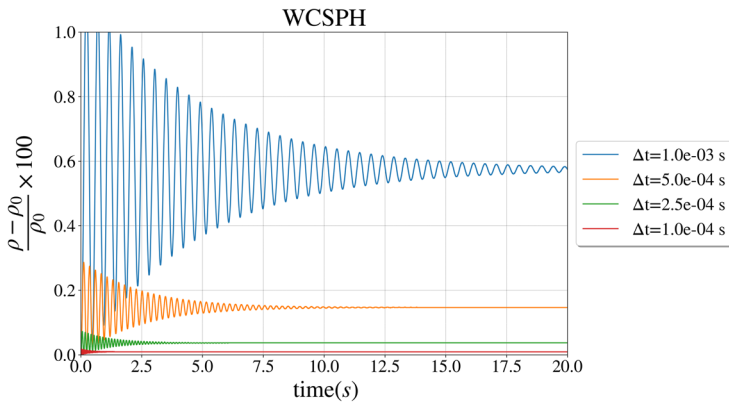
### 4.1 The “fluid-at-rest” compressibility test

In this experiment we monitor the time evolution of the density for a fluid stored in a rectangular container of dimensions  $1.1 \text{ m} \times 1.1 \text{ m} \times 1.2 \text{ m}$  (length/width/height). This “fluid-in-a-bucket” test probes the free surface handling by the Lagrangian approach that underpins WCSPH, ISPH and KCSPH, as well as the ability to enforce incompressibility. The model consists of 11 000 fluid SPH particles.





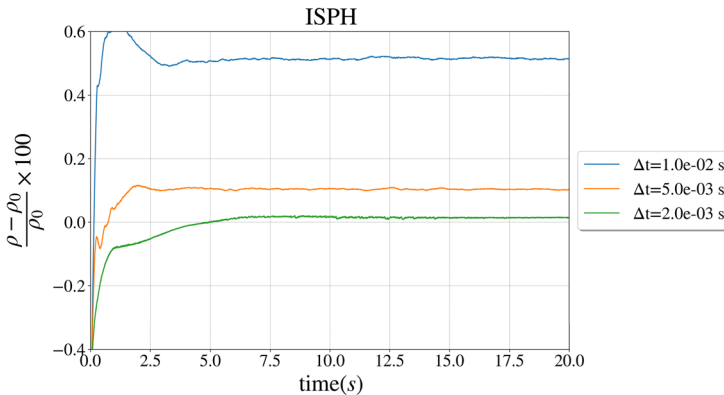
**Fig. 2** Time variation of the average compressibility offset for the KCSPH method (Color figure online)



**Fig. 3** Time variation of the average compressibility offset for the WCSPH method (Color figure online)

Finally, Fig. 2 reports KCSPH results. Setting aside the initial transients, KCSPH enforces incompressibility better than ISPH. This goes back to Eq. (35), which imposes incompressibility in KCSPH assertively. WCSPH is lacking this mechanism. Instead, an indirect, “penalty” approach controls compressibility by a stiff equation of state that penalizes density drift. One could draw a parallel between WCSPH and a widely used approach to handling contact forces in granular dynamics, in which one uses a stiff penalty force to enforce no-penetration conditions (a proxy for volume preservation, or incompressibility) at the mutual contact point between two elements [17]. Just like in the case of WCSPH, the penalty approach in contact mechanics requires small integration time steps for numerical stability. For granular dynamics, a comparison of an assertive (kinematically constrained) approach vs. a penalty approach is reported in [55].

Figure 3 reports WCSPH results for the same test. In line with expectations, compared to ISPH, a smaller time step is required by WCSPH to remain below the 1% compression threshold. This is explained by the numerical stiffness introduced by the equation of state in Eq. (39), which translates into a small integration time step size owing to the conditions in Eq. (41). The higher the value of the maximum velocity or  $k$  parameter, the higher the stiffness, and thus the smaller the time step at which the simulation can proceed stably. As a



**Fig. 4** Time variation of the average compressibility offset for the ISPH method (Color figure online)

rule of thumb, an WCSPH step-size 10 times smaller than the ISPH time step was noted to lead to comparable levels of errors in the two solvers. This is offset by the observation that one WCSPH time step is markedly more inexpensive than an ISPH one owing to the latter solver requiring the solution of a Poisson problem.

Finally, Fig. 4 reports the history of the average compression for ISPH, which reaches 1% compression at relatively large step size ( $\Delta t = 0.01$  s). The 1% compression threshold was selected to be in line with results reported elsewhere, e.g., [9].

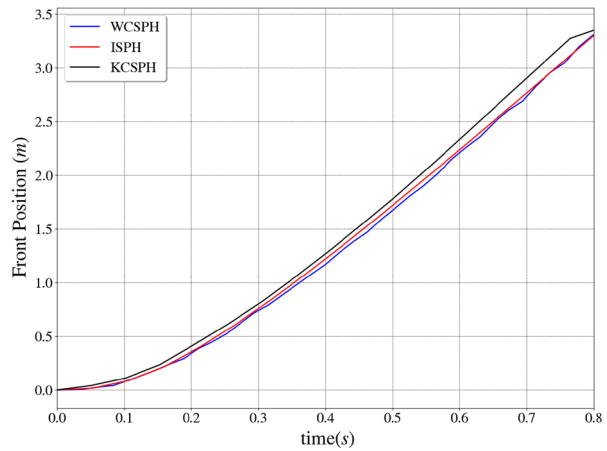
## 4.2 Dam break

Lagrangian methods such as SPH use the dam break benchmark test to assess the performance of fluid solvers in conjunction with a high transients, free-surface problem [14, 32, 42, 59, 76]. The fluid domain is a rectangular prism of size  $2.0 \text{ m} \times 0.5 \text{ m} \times 1.0 \text{ m}$  consisting of 64000 SPH particles that are placed on a regular lattice at  $t = 0$  s. The reference density and viscosity are  $\rho_0 = 1000 \text{ kg/m}^3$  and  $\mu = 0.001 \text{ Pa s}$ . The gravity  $g = -9.8 \text{ m/s}^2$  is applied in the  $z$  direction. The KCSPH, WCSPH, and ISPH results are compared from two perspectives: (i) the fluid front position over time, and (ii) the roll up and the second splash – two characteristics highlighted in previous studies [2, 14]. Figure 5 shows the fluid front position as a function of time. While the KCSPH solution slightly over-estimate the front speed (no dissipation term is implemented in KCSPH), nearly identical front propagation is predicted by WCSPH and ISPH. With regard to the roll-up and second splash characteristics, all three methods predict well these two hallmark features of the dam break experiment, see Fig. 6. In terms of pressure profiles, it is practically less challenging to obtain smooth distributions via ISPH method, mainly due to numerical and physical dissipation aspects involved in the solution. In the KCSPH method, the pressure field is defined by the Lagrange multiplier of the constant density constraints, and similarly to WCSPH method, it is in general noisy owing to the index 3 DAE solution approach employed by KCSPH. See Sect. A.2 for a more detailed discussion on comparison of pressure profiles.

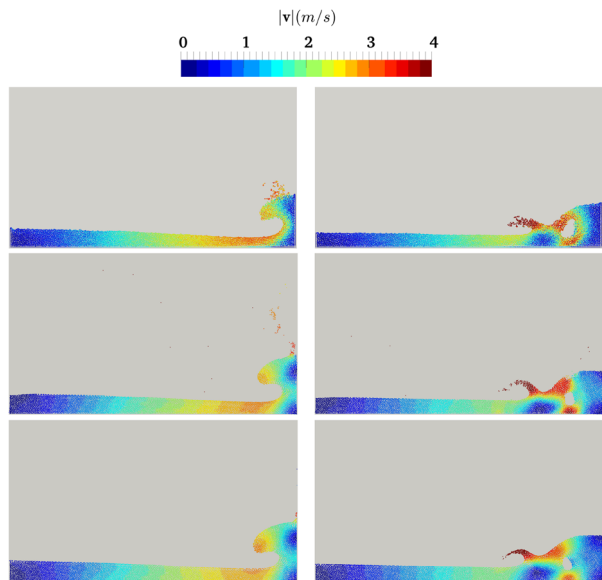
## 4.3 Fluid–solid interaction: a floating cylinder problem

This experiment was used to compare the methods in conjunction with a 3D scenario that displayed ample fluid–solid boundary movement. It may be regarded as a simplified prob-

**Fig. 5** Comparison of water-front propagation between KCSPH (top), WCSPH (middle), and ISPH (bottom) (Color figure online)

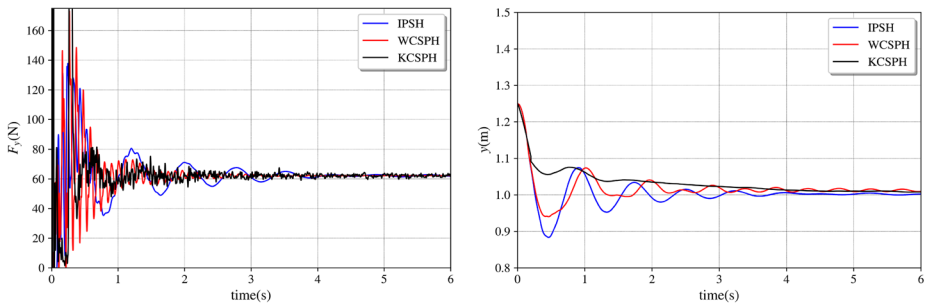


**Fig. 6** Comparison of the roll-up ( $t = 1.75$  s, left) and the second splash ( $t = 2.05$  s, right) characteristics between KCSPH (top), WCSPH (middle), and ISPH (bottom) (Color figure online)



lem that, in more complex forms, is conspicuous in several fluid–structure interaction applications in coastal and offshore structures, e.g., renewable energy devices and caissons. A cylindrical object of radius  $r = 0.12$  m and length  $L = 0.2$  m is released from the height  $h = 0.25$  m above the surface of a tank of water at  $t = 0$  s. The dimension of the fluid domain is  $1.0 \text{ m} \times 1.0 \text{ m} \times 0.2 \text{ m}$  (width  $\times$  height  $\times$  depth). The gravity  $g = -9.8 \text{ m/s}^2$  is applied in the  $y$  direction. The reference density and viscosity of the water are  $\rho = 1000 \text{ kg/m}^3$  and  $\mu = 0.001 \text{ Pa s}$ . The density of the cylinder is  $\rho_s = 0.7\rho$ , which turns it into a floating structure.

The cylinder oscillates until its initial potential energy damps out. The *steady-state* solution of this problem is given by Newton’s second law and basic hydrostatics. Indeed, the upward buoyant force exerted on the body should balance the weight of the object, i.e.,  $\rho_s g V = 62.0$ . Figure 7 illustrates the vertical position of the cylinder and the vertical com-



**Fig. 7** Upward buoyant force imparted over time by the fluid to the floating cylinder (left) and the position of the cylinder over time (right) (Color figure online)

ponent of the fluid–structure interaction forces and shows that all methods close in on the steady-state solution approximately 6 s into the simulation.

The variation of the vertical position of the cylinder is fairly smooth for all three methods and the discrepancy between the steady-state results is less than the SPH kernel length employed for the three methods. Note, however, that the forces predicted by WCSPH and KCSPH are noisier than IPSH. Many aspects of the underlying numerical methods of each approach contribute to such transient discrepancy. From the discretization standpoint, the number of particles within the kernel support, as well as the kernel function itself are the main contributing factors. From the integration standpoint, the numerical dissipation of the time integrator as well as the treatment of the pressure equation play important roles. For instance, the diffusion nature of the Poisson equation solved in IPSH has a smooth effect to the pressure field, while the state equation used in WCSPH is stiff and contributes to the noise in the pressure field. As seen in Fig. 7(right) though, after a transient phase, all solvers are settling on the same steady state configuration.

#### 4.4 Sloshing

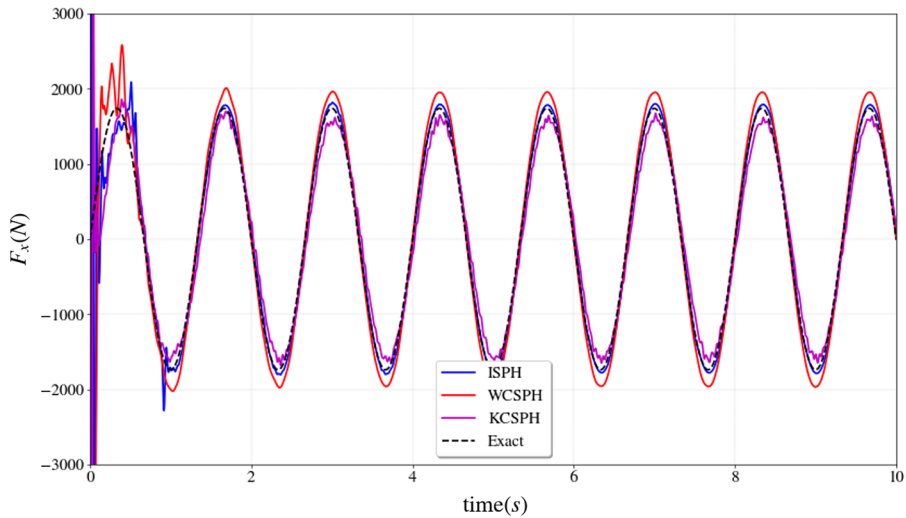
Sloshing probes the ability of a CFD solver to handle a free-surface problem that experiences lower frequency transients (when compared to the dam break). In this experiment a fluid container undergoes a forced vibration motion according to  $x = X_0 \sin(2\pi f t)$ , where  $t$  is time, and  $f$  and  $X_0$  are the frequency and the amplitude of the vibration, respectively. The analytical solution of this problem under the inviscid flow assumption was discussed in [18] and involves a set of natural frequencies  $\omega_n$  defined as

$$\omega_n^2 = (2n - 1) \pi \left( \frac{g}{w} \right) \tanh \left[ (2n - 1) \pi \left( \frac{h}{w} \right) \right],$$

where  $n$  is the mode number,  $g$  is the value of the gravitational acceleration,  $w$  is the tank width, i.e., the tank dimension in the direction of oscillation, and  $h$  is the height of the fluid at rest. The analytical amplitude  $F_{x_0}$  of the net force exerted by the fluid on a sloshing tank in a forced vibration can then be expressed as

$$\frac{F_{x_0}}{\Omega^2 X_0 m_l} = 1 + 8 \frac{w}{h} \sum_{n=1}^N \frac{\tanh[(2n - 1) \pi h/w]}{(2n - 1)^3 \pi^3} \frac{\Omega^2}{\omega_n^2 - \Omega^2}.$$

Above,  $\Omega = 2\pi f$  is the frequency of the oscillation and  $m_l$  is the mass of the liquid. The exerted force on the container in the horizontal direction is expressed as  $F_x = F_{x_0} \sin(2\pi f t)$ .



**Fig. 8** Sloshing experiment: Comparison of fluid–structure interaction force along the axis of periodic motion (Color figure online)

We compare the KCSPH, WCSPPH, and ISPH approximations with the analytical solution for the following problem: the dimension of the domain ( $w \times b \times h$ ) is  $1.2 \text{ m} \times 0.4 \text{ m} \times 1.7 \text{ m}$  and the fluid density is  $\rho_0 = 1000 \text{ kg/m}^3$ . The gravity  $g = -9.8 \text{ m/s}^2$  is applied in the  $z$  direction. The numerical model is comprised of 98k SPH markers, which are placed on a uniform Cartesian lattice at  $t = 0 \text{ s}$ . The frequency and amplitude of the vibration are set to be  $f = 0.75 \text{ s}^{-1}$  and  $X_0 = 0.1 \text{ m}$ . Figure 8 reports the sloshing force exerted on the container over time.

Ignoring the initial transient period, all methods predict the dynamic interaction forces with reasonable accuracy. We attribute the initial discrepancy between the numerical and the analytical solutions to the fact that the simulation tank starts its motion from rest; hence the initial condition of the simulation is different from the analytical solution, the latter only reported in the steady-state regime. However, the initial transient motion is damped in approximately 1 s. As shown in Fig. 8, the steady-state solution of ISPH closely matches the analytical solution; the steady-state solution of the WCSPPH slightly overpredicts whereas the KCSPH solution underpredicts the analytical result.

## 5 Discussion, conclusions, and future work

This effort set out to gauge the performance of a multibody dynamics-based approach to fluid dynamics, by comparing it with two classical Lagrangian CFD approaches: (i) WCSPPH, which uses an equation of state for pressure along with explicit time stepping; and (ii) ISPH, which falls back on a Poisson problem to enforce incompressibility with Chorin-style time stepping. KCSPH enforces incompressibility via kinematic constraints that assert the constant value of the density; a half-implicit symplectic Euler integrator is used for time stepping.

*KCSPH vs. WCSPPH, and ISPH: implementation aspects.* The three methods discussed were implemented in an open source simulation engine called Chrono [56, 71]. WCSPPH and ISPH

**Table 1** Dam break, qualitative information: solver execution times for the first, high transients, second of the simulation

method	$\Delta t$ (s)	simulation time (s)	average time per step (s)
WCSPH	$1.0 \times 10^{-4}$	1646	0.17
WCSPH	$2.5 \times 10^{-4}$	682	0.17
ISPH	$1.0 \times 10^{-3}$	1692	1.69
ISPH	$5.0 \times 10^{-3}$	369	1.84
KCSPH	$1.0 \times 10^{-3}$	1523	1.52
KCSPH	$5.0 \times 10^{-3}$	531	2.65

leveraged GPU computing through CUDA [52]; KCSPH used multicore parallel computing via OpenMP [53]. The amount of effort required to implement these solvers in software was quite different. Implementing the WCSPH solver was easier than ISPH, which was easier than KCSPH. ISPH requires the solution of a sparse linear system on the GPU; such a solver is not readily available and herein the implementation resorted to Krylov subspace iterative methods such as BICGSTAB and GMRES [62]. A memory-efficient solver requires sparse storage of the underlying systems (see Eqs. (56) and (58)). Assembling and solving these sparse linear systems made the ISPH implementation nontrivial. KCSPH was more challenging to implement since it required at each integration step the solution of a constrained quadratic optimization problem, see [44]. This optimization problem was posed in hundreds of thousands of variables – as many as SPH particles used in the formulation and its solution was found using a Nesterov-type method [43].

*KCSPH vs. WCSPH, ISPH: solution robustness.* Due to the coupling they establish between the field variables, KCSPH provides equally robust solution as ISPH. Owing to the integration scheme used in KCPSH and ISPH, the velocity and pressure/density are coupled more tightly than in WCSPH, where pressure depends only on the density. Specifically in KCSPH, the pressures, which are a proxy of the Lagrange multipliers for the density kinematic constraints, are coupled with the velocity in a fully implicit sense.

*KCSPH vs. WCSPH and ISPH: quality of numerical solution.* A simple answer to the question “which method provides better quality results?” is difficult to produce as multiple factors come into play in determining the quality of the solution, e.g., the particle resolution, the consistency vs. conservancy dichotomy, the decision to use or not particle shifting, the size of the time step, the nature of the problem solved, etc. In general, ISPH turned out to be easier to set up and get good results with. It is robust (non-finicky) and for a given simulation time budget it provides better quality results. One stumbling block is that ISPH requires a sparse linear solver. WCSPH, particularly for the inconsistent version, usually yields more noisy results – this also came through in the flow-around-the-cylinder test. KCSPH shows great potential (short simulation times, see discussion below), but it is a new method and its current implementation needs further work to address modeling (handling of viscosity, no slip boundary conditions), and numerical solution (scalability) aspects.

*KCSPH vs. WCSPH, ISPH: solution time.* While an efficiency comparison between KCSPH, WCSPH, and ISPH falls outside the scope of this contribution owing to the sheer scope of such an undertaking, we considered insightful to provide timing results to understand, insofar as order of magnitude is concerned, how long an SPH simulation would take. The problem considered was the dam break simulation of Sect. 4.2. There was no attempt to optimize the WCSPH, ISPH, and KCSPH implementations. Implemented in Chrono, WCSPH and ISPH rely on GPU computing; KCSPH draws on OpenMP. Table 1 shows *qualitative* information regarding the first second of simulation of the dam break problem. All 3D solvers

used 215 000 SPH markers. WCSPH takes smaller step sizes due to the explicit time stepping. However, its effort per time step is both step-size independent and cheaper than for ISPH or KCSPH. For the latter two, given the iterative nature of the solution process, higher computational costs are incurred for larger step sizes. Note that even for the same integration time step, the amount of time required to converge at different points in the ISPH/KCSPH simulation might be different owing to the difference in the number of iterations that the underlying linear system/optimization solvers require to converge. Note that at small time steps, for a highly accurate solution, KCSPH is the faster method.

*KCSPH vs. WCSPH and ISPH: where applicable.* WCSPH is the method of choice for many single-phase and free-surface CFD problems such as the dam break and Poiseuille flow. ISPH is a better choice when more complex physics, e.g., vortex shedding, boundary layer separation, etc., are present. This is the case for problems like flow around cylinder and flow over backward facing step at higher Reynolds numbers ( $Re \approx 10^2$ – $10^3$ ). ISPH is also the solver of choice for FSI problems with relatively simple boundary geometries. On the other hand, KCSPH leverages multibody dynamics approaches, which opens the door to monolithic frameworks for more involved FSI problems, e.g., fording scenarios with complex vehicle models (wheeled, tracked) [44]. Indeed, imposing ISPH or WCSPH boundary conditions on a complex mesh, e.g., a vehicle, requires a uniformly generated point cloud, whereas KCSPH does not have this restriction and is more flexible in terms of boundary condition enforcement.

*KCSPH, WCSPH, ISPH: connections to other physics.* The inspiration for the KCSPH method used in this study is a granular dynamics solution approach. Indeed, WCSPH has a granular dynamics twin in the discrete element method (DEM) [17]: in DEM, there is a penalty force associated with violating the no-penetration condition; in WCSPH there is a penalty force (pressure) tied to the violation of the incompressibility condition. However, there is a second class of granular dynamics methods that belongs to the family of complementarity formulations, or differential variational inequality approaches, in which the no-penetration condition is enforced by kinematic constraints, see, for instance, [66]. KCSPH is in fluid dynamics the twin of the complementarity approach in frictional contact multibody dynamics. Continuing this granular dynamics–fluid dynamics parallel, the analog of the contact force in granular dynamics is the pressure in fluid dynamics. Another salient point in this analogy is that the pressure in ISPH is the analog of the Lagrange multiplier that enforces the  $\rho(t) = \rho_0$  in KCSPH, made clear in this contribution by casting the SPH spatial discretization in a matrix–vector form. This is unsurprising given that in an incompressible, Newtonian fluid model the pressure is devoid of a thermodynamic meaning and instead becomes a mechanical attribute of the flow. For Newtonian fluids though, the analogy ends here – while for a Newtonian fluid the shear force only depends on velocity, in granular dynamics the friction force is tied to the contact (normal) force through a yield condition that caps the former at a value equal to the normal force scaled by  $\mu$ , the friction coefficient,  $f_f \leq \mu f_n$ .

*KCSPH: what comes next.* There are several directions in which KCSPH needs to be improved. First, KCSPH should be endowed with support for computing viscous forces. In its current implementation, it only handles inviscid flows. Second, our KCSPH implementation needs a mechanism to enforce no-slip boundary conditions. No-slip could be implemented through the friction between the SPH particles and the boundary; or, by kinematically constraining the motion of particles via bilateral constraints. Third, ongoing work is focused on improving the particle shifting, a development that will benefit KCSPH but is applicable to a broader class of SPH approaches when handling free surface flows. Preliminary results

in this direction are reported elsewhere [31]. Likewise, it remains to carry out a more systematic analysis of the capabilities of the new method for FSI applications by using a more thorough set of benchmark results as described, for instance, in [74]. Lastly, KCSPH is only implemented to leverage parallel computing using multi-core hardware via OpenMP. For a fair comparison, it should be implemented on the GPU, just like WCSPH and ISPH.

**Acknowledgement** The first author was funded through National Science Foundation grant CMMI-1635004. The last author was funded through US Army Research Office project W911NF1910431.

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

## Appendix: Verification and validation of ISPH and WCSPH

### A.1 Poiseuille flow

Due to the absence of the pressure gradient term, this test is often used to validate the accuracy of the viscosity model in the Navier–Stokes equations. The problem is set up as follows: A rectangular domain of fluid with dimensions  $1 \text{ m} \times 0.2 \text{ m} \times 0.08 \text{ m}$  (in the direction of flow, width of the channel, depth of the channel) consisting of 24 000 SPH particles is subjected to a body force  $\mathbf{f}_b = 0.01 \text{ m/s}^2$  in the positive  $x$  direction. Boundary conditions enforce periodic boundary at the left and right, as well as at the front and back, patch pairs of the domain. The top and bottom boundary conditions are no-slip velocity, see Eq. (53). The fluid is accelerated by the body-force and reaches to a steady-state parabolic velocity profile. The analytical solution of this problem is given by [49]

$$u_x(y, t) = \frac{\rho|\mathbf{f}_b|}{2\mu}y(y - L) + \sum_{n=0}^{\infty} \frac{4\rho|\mathbf{f}_b|L^2}{\mu\pi^3(2n+1)^3} \sin\left(\frac{\pi y}{L}(2n+1)\right) \exp\left(-\frac{(2n+1)^2\pi^2\mu}{\rho L^2}t\right), \quad (60)$$

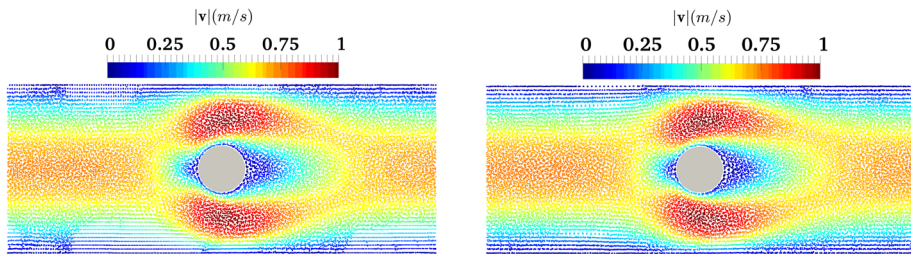
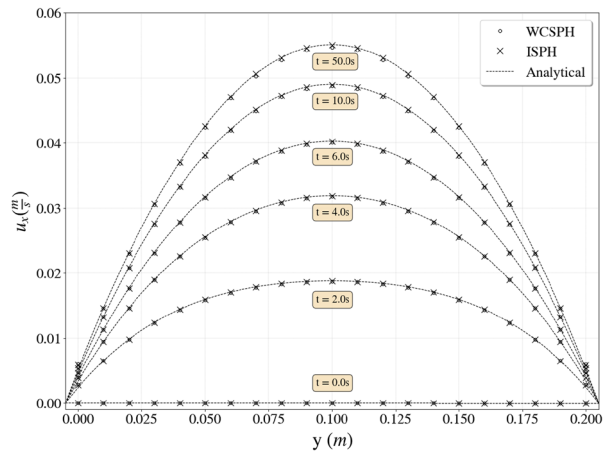
where  $\rho_0 = 1000 \text{ kg/m}^3$ ,  $\mu = 1.0 \text{ Pa s}$  and  $L = 0.2 \text{ m}$ . The WCSPH and ISPH results show a close match with the velocity profiles obtained from the analytical solution as shown in Fig. 9. The time required for reaching steady-state solution is assessed according to the change in the maximum velocity. The relative difference between the maximum velocity at  $t = 50 \text{ s}$  and  $t = 100 \text{ s}$  is 0.0001%, making the solution at  $t = 50 \text{ s}$  a good approximation of the steady-state solution for most practical purposes. The average error of the velocity in the flow direction at  $t = 50 \text{ s}$  is 0.1% and 0.7% for ISPH and WCSPH, respectively. The numerical error associated with the spatial and temporal discretization decrease by increasing the resolution and decreasing the time step, respectively. The standard discretizations (see Eqs. (9)–(10)) are used in the WCSPH formulation along with larger kernel support radius to maintain the spatial accuracy, while the consistent discretization (see Eqs. (12)–(13)) is applied in the ISPH formulation. A more in-depth discussion on the accuracy of the discretization method was presented in [30].

### A.2 Flow around cylinder

In the Poiseuille flow example, the flow field was the outcome of an interplay between inertia, viscous, and body forces. The “flow around cylinder” test is used to verify the ISPH

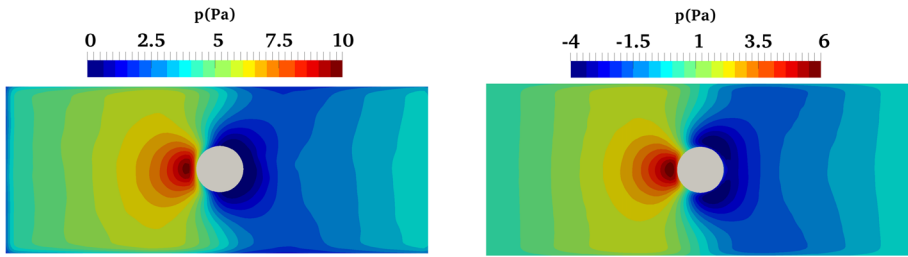


**Fig. 9** Velocity profile of transient Poiseuille flow obtained from numerical simulation of WCSPH, ISPH, and series solution at different times



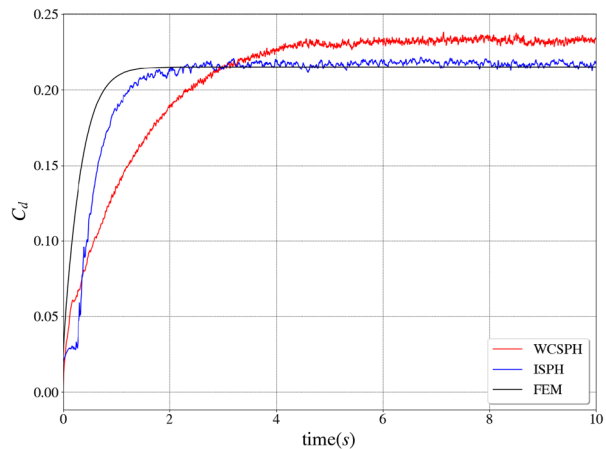
**Fig. 10** Comparison of the steady-state velocity profiles predicted with WCSPH (left) and ISPH (right) (Color figure online)

and WCSPH implementations in a scenario in which pressure gradient effects in the Navier–Stokes equations are not negligible. In this test, the cylinder of radius 0.05 m is positioned at the center of a rectangular domain of height 0.4 m and length 1.0 m. No-slip boundary conditions are applied to the top and bottom walls while periodic (cyclic) conditions are maintained at the left (inlet) and right (outlet) patches. A constant body force  $\mathbf{f}_b = 1.0 \text{ m/s}^2$  is applied in order to balance the viscous force. The density and viscosity are set to  $\rho_0 = 10.0 \text{ kg/m}^3$  and  $\mu = 0.1 \text{ Pa}\cdot\text{s}$ . The simulation time-steps are  $10^{-3} \text{ s}$  and  $2 \times 10^{-4} \text{ s}$  for the ISPH and WCSPH methods, respectively; the models use 36000 SPH fluid markers. Figure 10 illustrates the steady-state *velocity* predicted by WCSPH and ISPH; the profiles are nearly identical to each other and also to an FEM-generated solution whose plot is not shown here but reported in [60]. In contrast, the WCSPH and ISPH *pressure* fields are markedly less similar. As shown in Fig. 11, which reports the pressure fields on fixed grids, the results are qualitatively but not quantitatively similar. The quality of the ISPH pressure solution was superior (based on a comparison with a FEM approach [60]), an aspect that can come into play, for instance, in acoustics applications. We also note that there is no unique ISPH solution for pressure when solving the Poisson equation under pure Neumann boundary conditions – in fact, the pressure solution is unique up to a constant. In other words, in the absence of Dirichlet boundary conditions, pressure is a relative quantity in the Navier–Stokes equations. Whereas the reference (*minimum*) pressure for WCSPH is often set to zero, in ISPH, we choose the pressure solution with the *average* value of zero. Consequently, the difference between the legend of the plots in Fig. 11 has no real significance as long as the difference



**Fig. 11** Comparison of the steady-state pressure profiles predicted with WCSPH (left) and ISPH (right) (Color figure online)

**Fig. 12** Variation of the drag coefficient over time (Color figure online)



between the solutions is a constant offset that only indicates a reference pressure. Note that the choice of the reference pressure does not affect the gradient of the pressure, which is the quantity that comes into play in the Navier–Stokes equation.

Herein, the expression used for the drag coefficient is  $C_d = \frac{F_d}{0.5\rho AU^2}$ , where  $F_d$  is the drag force magnitude along the  $x$  axis,  $\rho = \rho_0$  and  $U = 1$  m/s are the reference density and velocity, and  $A$  is the frontal area of the cylinder. Comparing drag coefficient results is insightful since this exercise provides a macro-scale perspective that, while looking past micro-scale fluctuations, captures emergent behavior of practical relevance. In Fig. 12, where a Finite Element Method (FEM) analysis provides the “ground truth”, ISPH and WCSPH show different drag coefficients at the onset of the simulation yet the steady state solutions are in good agreement – the relative error of the time-averaged drag coefficient over the last 2 s of the simulations is 6.4%. We posit that two reasons contributing to these discrepancies are: (i) the different time-integration schemes used in the formulations (see Sects. 3.3 and 3.2), each with its own amount of numerical damping [24]; and (ii) the vastly different treatment of the pressure by the two formulations.

## References

1. Acary, V., Brogliato, B.: Numerical Methods for Nonsmooth Dynamical Systems: Applications in Mechanics and Electronics, vol. 35. Springer, Berlin (2008)

2. Adami, S., Hu, X., Adams, N.: A generalized wall boundary condition for smoothed particle hydrodynamics. *J. Comput. Phys.* **231**(21), 7057–7075 (2012)
3. Adami, S., Hu, X., Adams, N.: A transport-velocity formulation for smoothed particle hydrodynamics. *J. Comput. Phys.* **241**, 292–307 (2013)
4. Anitescu, M.: Optimization-based simulation of nonsmooth rigid multibody dynamics. *Math. Program.* **105**(1), 113–143 (2006). <https://doi.org/10.1007/s10107-005-0590-7>
5. Anitescu, M., Hart, G.D.: A constraint-stabilized time-stepping approach for rigid multibody dynamics with joints, contact and friction. *Int. J. Numer. Methods Eng.* **60**(14), 2335–2371 (2004)
6. Antoci, C., Gallati, M., Sibilla, S.: Numerical simulation of fluid–structure interaction by SPH. *Comput. Struct.* **85**(11), 879–890 (2007)
7. Antuono, M., Colagrossi, A., Marrone, S.: Numerical diffusive terms in weakly-compressible SPH schemes. *Comput. Phys. Commun.* **183**(12), 2570–2580 (2012)
8. Asai, M., Aly, A.M., Sonoda, Y., Sakai, Y.: A stabilized incompressible SPH method by relaxing the density invariance condition. *J. Appl. Math.* **2012** (2012). <https://doi.org/10.1155/2012/139583>
9. Becker, M., Teschner, M.: Weakly compressible SPH for free surface flows. In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 209–217. Eurographics Association (2007)
10. Bertails-Descoubes, F., Cadoux, F., Daviet, G., Acary, V.: A nonsmooth Newton solver for capturing exact Coulomb friction in fiber assemblies. *ACM Trans. Graph.* **30**(1), 6 (2011)
11. Bodin, K., Lacoursiere, C., Servin, M.: Constraint fluids. *IEEE Trans. Vis. Comput. Graph.* **18**(3), 516–526 (2012)
12. Bonet, J., Lok, T.S.: Variational and momentum preservation aspects of smooth particle hydrodynamic formulations. *Comput. Methods Appl. Mech. Eng.* **180**(1–2), 97–115 (1999)
13. Chorin, A.J.: Numerical solution of the Navier–Stokes equations. *Math. Comput.* **22**(104), 745–762 (1968)
14. Colagrossi, A., Landrini, M.: Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *J. Comput. Phys.* **191**(2), 448–475 (2003)
15. Crespo, A.J., Domínguez, J.M., Rogers, B.D., Gómez-Gesteira, M., Longshaw, S., Canelas, R., Vaconadio, R., Barreiro, A., García-Feal, O.: DualSPHysics: open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH). *Comput. Phys. Commun.* **187**, 204–216 (2015)
16. Cummins, S.J., Rudman, M.: An SPH projection method. *J. Comput. Phys.* **152**(2), 584–607 (1999)
17. Cundall, P., Strack, O.: A discrete element model for granular assemblies. *Geotechnique* **29**, 47–65 (1979)
18. Dodge, F.T.: *The New “Dynamic Behavior of Liquids in Moving Containers”*. Southwest Research Inst., San Antonio (2000)
19. Ellero, M., Serrano, M., Español, P.: Incompressible smoothed particle hydrodynamics. *J. Comput. Phys.* **226**(2), 1731–1752 (2007). <https://doi.org/10.1016/j.jcp.2007.06.019>
20. Español, P., Revenga, M.: Smoothed dissipative particle dynamics. *Phys. Rev. E* **67**(2), 026705 (2003). <https://doi.org/10.1103/PhysRevE.67.026705>
21. Fatehi, R., Manzari, M.T.: Error estimation in smoothed particle hydrodynamics and a new scheme for second derivatives. *Comput. Math. Appl.* **61**(2), 482–498 (2011)
22. Gingold, R.A., Monaghan, J.J.: Smoothed particle hydrodynamics-theory and application to non-spherical stars. *Mon. Not. R. Astron. Soc.* **181**(1), 375–389 (1977)
23. Gurtin, M.E., Fried, E., Anand, L.: *The Mechanics and Thermodynamics of Continua*. Cambridge University Press, Cambridge (2010)
24. Hairer, E., Norsett, S., Wanner, G.: *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, Berlin (2009)
25. Haug, E.J.: *Computer-Aided Kinematics and Dynamics of Mechanical Systems Volume-I*. Prentice-Hall, Englewood Cliffs (1989)
26. Heyn, T., Anitescu, M., Tasora, A., Negrut, D.: Using Krylov subspace and spectral methods for solving complementarity problems in many-body contact dynamics simulation. *Int. J. Numer. Methods Eng.* **95**(7), 541–561 (2013). <https://doi.org/10.1002/nme.4513>
27. Hosseini, S.M., Feng, J.J.: Pressure boundary conditions for computing incompressible flows with SPH. *J. Comput. Phys.* **230**(19), 7473–7487 (2011)
28. Hu, X.Y., Adams, N.A.: An incompressible multi-phase SPH method. *J. Comput. Phys.* **227**(1), 264–278 (2007). <https://doi.org/10.1016/j.jcp.2007.07.013>
29. Hu, W., Pan, W., Rakhsha, M., Negrut, D.: An overview of an SPH technique to maintain second-order convergence for 2D and 3D fluid dynamics. *Tech. Rep. TR-2016-14, Simulation-Based Engineering Laboratory, University of Wisconsin-Madison* (2016)
30. Hu, W., Pan, W., Rakhsha, M., Tian, Q., Hu, H., Negrut, D.: A consistent multi-resolution smoothed particle hydrodynamics method. *Comput. Methods Appl. Mech. Eng.* **324**, 278–299 (2017)

31. Hu, W., Rakhsha, M., Yang, L., Kamrin, K., Negrut, D.: Modeling granular material dynamics and its two-way coupling with moving solid bodies using a continuum representation and the SPH method. *Computer methods in applied mechanics and engineering - under review* **00** (2021)
32. Hughes, J.P., Graham, D.I.: Comparison of incompressible and weakly-compressible SPH models for free-surface water flows. *J. Hydraul. Res.* **48**(S1), 105–117 (2010)
33. Ihmsen, M., Cornelis, J., Solenthaler, B., Horvath, C., Teschner, M.: Implicit incompressible SPH. *IEEE Trans. Vis. Comput. Graph.* **20**(3), 426–435 (2014)
34. Islam, M.R.I., Chakraborty, S., Shaw, A.: On consistency and energy conservation in smoothed particle hydrodynamics. *Int. J. Numer. Methods Eng.* **116**(9), 601–632 (2018)
35. Kaufman, D.M., Pai, D.K.: Geometric numerical integration of inequality constrained, nonsmooth Hamiltonian systems. *SIAM J. Sci. Comput.* **34**(5), A2670–A2703 (2012)
36. Kaufman, D.M., Sueda, S., James, D.L., Pai, D.K.: Staggered projections for frictional contact in multi-body systems. *ACM Trans. Graph.* **27**(5), 164 (2008)
37. Krimi, A., Rezoug, M., Khelladi, S., Nogueira, X., Deligant, M., Ramírez, L.: Smoothed Particle Hydrodynamics: a consistent model for interfacial multiphase fluid flow simulations. *J. Comput. Phys.* **358**, 53–87 (2018). <https://doi.org/10.1016/j.jcp.2017.12.006>
38. Lastiwka, M., Basa, M., Quinlan, N.J.: Permeable and non-reflecting boundary conditions in SPH. *Int. J. Numer. Methods Fluids* **61**(7), 709–724 (2009)
39. Libersky, L., Petschek, A., Carney, T., Hipp, J., Allahdadi, F.: High strain Lagrangian hydrodynamics: a three-dimensional SPH code for dynamic material response. *J. Comput. Phys.* **109**(1), 67–75 (1993)
40. Liu, G., Liu, M.B.: *Smoothed Particle Hydrodynamics: A Mesh-Free Particle Method*. World Scientific, Singapore (2003)
41. Lucy, L.B.: A numerical approach to the testing of the fission hypothesis. *Astron. J.* **82**, 1013–1024 (1977)
42. Martin, J.C., Moyce, W.J.: Part IV. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philos. Trans. R. Soc. A, Math. Phys. Eng. Sci.* **244**(882), 312–324 (1952). <https://doi.org/10.1098/rsta.1952.0006>
43. Mazhar, H., Heyn, T., Tasora, A., Negrut, D.: Using Nesterov’s method to accelerate multibody dynamics with friction and contact. *ACM Trans. Graph.* **34**(3), 32 (2015)
44. Mazhar, H., Pazouki, A., Rakhsha, M., Jayakumar, P., Negrut, D.: A differential variational approach for handling fluid–solid interaction problems via smoothed particle hydrodynamics. *J. Comput. Phys.* **371**, 92–119 (2018)
45. Monaghan, J.J.: An introduction to SPH. *Comput. Phys. Commun.* **48**(1), 89–96 (1988)
46. Monaghan, J.J.: Simulating free surface flows with SPH. *J. Comput. Phys.* **110**(2), 399–406 (1994). <https://doi.org/10.1006/jcph.1994.1034>
47. Monaghan, J.J.: Smoothed particle hydrodynamics. *Rep. Prog. Phys.* **68**(1), 1703–1759 (2005)
48. Monaghan, J.J., Kocharyan, A.: SPH simulation of multi-phase flow. *Comput. Phys. Commun.* **87**(1), 225–235 (1995). [https://doi.org/10.1016/0010-4655\(94\)00174-Z](https://doi.org/10.1016/0010-4655(94)00174-Z)
49. Morris, J.P., Fox, P.J., Zhu, Y.: Modeling low Reynolds number incompressible flows using SPH. *J. Comput. Phys.* **136**(1), 214–226 (1997)
50. Nair, P., Pöschel, T.: Dynamic capillary phenomena using incompressible SPH. *Chem. Eng. Sci.* **176**, 192–204 (2018). <https://doi.org/10.1016/j.ces.2017.10.042>
51. Negrut, D., Serban, R., Tasora, A.: Posing multibody dynamics with friction and contact as a differential complementarity problem. *J. Comput. Nonlinear Dyn.* **13**(1), 014503 (2017). <https://doi.org/10.1115/1.4037415>
52. NVIDIA: *CUDA Programming Guide* (2015). Available online at <http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>
53. OpenMP: *Specification Standard 4.5* (2017). Available online at <http://openmp.org/wp/>
54. Pazouki, A., Negrut, D.: A numerical study of the effect of particle properties on the radial distribution of suspensions in pipe flow. *Comput. Fluids* **108**, 1–12 (2015)
55. Pazouki, A., Kwarta, M., Williams, K., Likos, W., Serban, R., Jayakumar, P., Negrut, D.: Compliant versus rigid contact: a comparison in the context of granular dynamics. *Phys. Rev. E* **96**, 042905 (2017). <https://doi.org/10.1103/PhysRevE.96.042905>
56. Project Chrono: Chrono: an open source framework for the physics-based simulation of dynamic systems. <http://projectchrono.org>. Accessed: 2016-03-07
57. Project Chrono Development Team: Chrono: an open source framework for the physics-based simulation of dynamic systems. <https://github.com/projectchrono/chrono>. Accessed: 2019-12-07
58. Quinlan, N.J., Basa, M., Lastiwka, M.: Truncation error in mesh-free particle methods. *Int. J. Numer. Methods Eng.* **66**(13), 2064–2085 (2006)
59. Rakhsha, M., Pazouki, A., Serban, R., Negrut, D.: Using a half-implicit integration scheme for the SPH-based solution of fluid–solid interaction problems. *Comput. Methods Appl. Mech. Eng.* **345**, 100–122 (2019)

60. Rakhsha, M., Yang, Y., Kees, C., Negrut, D.: Lagrangian vs. Eulerian: an analysis of two solution methods for free-surface flows and fluid solid interaction problems. Under Review (2019)
61. Randles, P.W., Libersky, L.D.: Smoothed Particle Hydrodynamics: some recent improvements and applications. *Comput. Methods Appl. Mech. Eng.* **139**(1–4), 375–408 (1996)
62. Saad, Y.: Overview of Krylov subspace methods with applications to control problems. Research Institute for Advanced Computer Science, NASA Ames Research Center (1989)
63. Shadloo, M.S., Zainali, A., Yildiz, M., Suleman, A.: A robust weakly compressible SPH method and its comparison with an incompressible SPH. *Int. J. Numer. Methods Eng.* **89**(8), 939–956 (2012)
64. Shao, S., Lo, E.: Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Adv. Water Resour.* **26**(7), 787–800 (2003). [https://doi.org/10.1016/S0309-1708\(03\)00030-7](https://doi.org/10.1016/S0309-1708(03)00030-7)
65. Smith, B., Kaufman, D.M., Vouga, E., Tamstorf, R., Grinspun, E.: Reflections on simultaneous impact. *ACM Trans. Graph.* **31**(4), 106 (2012). <https://doi.org/10.1145/2185520.2185602>
66. Stewart, D.E.: Rigid-body dynamics with friction and impact. *SIAM Rev.* **42**(1), 3–39 (2000)
67. Stewart, D.E., Trinkle, J.C.: An implicit time-stepping scheme for rigid-body dynamics with inelastic collisions and Coulomb friction. *Int. J. Numer. Methods Eng.* **39**, 2673–2691 (1996)
68. Takeda, H., Miyama, S.M., Sekiya, M.: Numerical simulation of viscous flow by smoothed particle hydrodynamics. *Prog. Theor. Phys.* **92**(5), 939–960 (1994)
69. Tartakovsky, A.M., Panchenko, A.: Pairwise Force Smoothed Particle Hydrodynamics model for multiphase flow: surface tension and contact line dynamics. *J. Comput. Phys.* **305**, 1119–1146 (2016). <https://doi.org/10.1016/j.jcp.2015.08.037>
70. Tasora, A., Anitescu, M.: A convex complementarity approach for simulating large granular flows. *J. Comput. Nonlinear Dyn.* **5**(3), 1–10 (2010). <https://doi.org/10.1115/1.4001371>
71. Tasora, A., Serban, R., Mazhar, H., Pazouki, A., Melanz, D., Fleischmann, J., Taylor, M., Sugiyama, H., Negrut, D.: Chrono: an open source multi-physics dynamics engine. In: Kozubek, T. (ed.) *High Performance Computing in Science and Engineering. Lecture Notes in Computer Science*, pp. 19–49. Springer, Berlin (2016)
72. Trask, N., Kim, K., Tartakovsky, A., Perego, M., Parks, M.L.: A highly-scalable implicit SPH code for simulating single- and multi-phase flows in geometrically complex bounded domains. Tech. rep., Sandia National Lab. (SNL-NM), Albuquerque, NM (United States) (2015)
73. Trask, N., Maxey, M., Kimb, K., Perego, M., Parks, M.L., Yang, K., Xu, J.: A scalable consistent second-order SPH solver for unsteady low Reynolds number flows. *Comput. Methods Appl. Mech. Eng.* **289**, 155–178 (2015)
74. Turek, S., Hron, J.: Proposal for numerical benchmarking of fluid–structure interaction between an elastic object and laminar incompressible flow. In: *Fluid–Structure Interaction*, pp. 371–385. Springer, Berlin (2006)
75. Wang, Z.B., Chen, R., Wang, H., Liao, Q., Zhu, X., Li, S.Z.: An overview of smoothed particle hydrodynamics for simulating multiphase flow. *Appl. Math. Model.* **40**(23), 9625–9655 (2016). <https://doi.org/10.1016/j.apm.2016.06.030>
76. Xu, X., Deng, X.L.: An improved weakly compressible SPH method for simulating free surface flows of viscous and viscoelastic fluids. *Comput. Phys. Commun.* **201**, 43–62 (2016)
77. Xu, R., Stansby, P., Laurence, D.: Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *J. Comput. Phys.* **228**(18), 6703–6725 (2009)
78. Yang, Q., Jones, V., McCue, L.: Free-surface flow interactions with deformable structures using an SPH–FEM model. *Ocean Eng.* **55**, 136–147 (2012)