ISPH–PBD: coupled simulation of incompressible fluids and deformable bodies

Nadine Abu Rumman, Prapanch Nair, Patric Müller, Loic Barthe, David Vanderhaeghe

This is an author version of an article published in The Visual Computer Journal.
The final authenticated version is available online at: http://dx.doi.org/10.1007/s00371-019-01700-y.

ISPH-PBD: Coupled Simulation of Incompressible Fluids and Deformable Bodies

Nadine Abu Rumman[1] · **Prapanch Nair**[2] · **Patric Müller**[2] · **Loïc Barthe**[3] · **David Vanderhaeghe**[3]

Published online 18 May 2019

Abstract We present an efficient and stable method for simulating the two-way coupling of incompressible fluids and deformable bodies. In our method, the fluid is represented by particles, and simulated using divergence-free Incompressible Smoothed Particle Hydrodynamics (ISPH). The deformable bodies are represented by polygonal meshes, where the elastic deformations are simulated using a Position Based Dynamics (PBD) scheme. Our technique enforces incompressibility on the fluid using divergence-free constraints on the velocity field, while it effectively simulates the physical features of deformable bodies. Most current ISPH methods are struggling with the issue of free surface boundary conditions. We handle this problem by introducing a novel free surface formulation, where our free surface model obviates the need to identify the surface particles. For the interaction between the fluid and the deformable solids, we model the forces that both phases, fluid and solid, exert upon each other. We demonstrate that our approach effectively handles complex coupling scenarios between fluids and thin deformable shells or highly deformable solids, and produces plausible results.

[1]N. Aburumman
University College London (UCL)
London, United Kingdom
Tel.: +44 20 7679 2000
E-mail: Firstname.Surname@irit.fr

[2]P. Nair, P. Müller
Institute for Multiscale Simulation (MSS)
Erlangen, Germany
Tel.: +49 9131 8520867
E-mail: Firstname.Surname@fau.de

[3] L. Barthe D. Vanderhaeghe
IRIT, Université de Toulouse, CNRS, INPT, UPS, UT1C, UT2J.
Toulouse, France
Tel.: +33 05 6155 6312
E-mail: Firstname.Surname@irit.fr

Keywords ISPH · PBD · Thin shells · Deformable bodies · Fluids

1 Introduction

Two-way coupling between fluids and deformable bodies can produce visually and mechanically realistic behaviour, such as the effect of the object's elasticity pushing fluid away, while simultaneously getting deformed under the load of the fluid. Simulating such complex fluid behaviour involving deformable objects is at the core of numerous applications in computational physics as well as in computer animation. In addition, coupled simulation between fluids and deformable solids is an increasingly demanding topic in computer graphics, where it has many applications, including animated feature films, virtual surgery, and movie production. However, the interaction between fluids and deformable solids is complex and difficult, making it computationally demanding and expensive. Moreover, the repulsive forces between deformable solid masses and fluid particles frequently lead to situations where very small time steps are required to guarantee numerical stability. Although various fluid flow dynamics in the physical world have been studied extensively [26], methods to capture the complicated interaction behaviour between elastic objects and fluids have received less attention in the computer graphics literature [35]. Previous approaches tried to address the computation and instability issues in these simulations [34, 69]. Nevertheless, providing efficient and plausible two-way coupling between particle-based fluids and deformable bodies represented by meshes requires particular attention and is still a challenging open problem.

In a two-way coupling scenario, the choice of the approaches to simulate both the fluid and deformable body plays an important role. In the case of fluids, particle-based

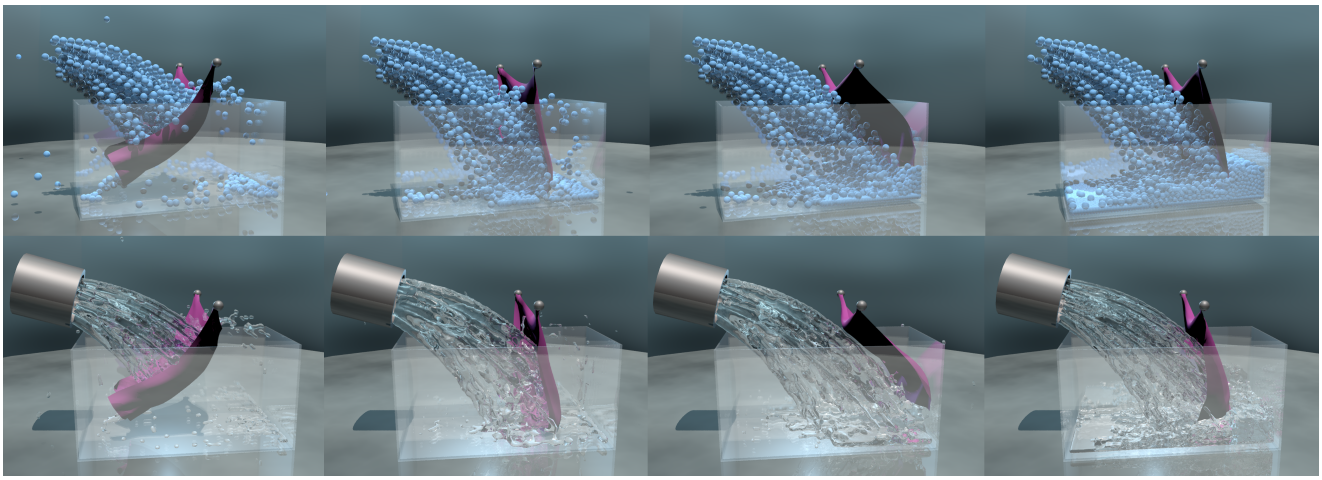


Fig. 1 Coupled simulation of a deformable thin shell and a fluid using our method ($11K$ cloth constraints and $27K$ fluid particles, at 21 fps), where the cloth deforms under the impact of the fluid, while the fluid splashes around. Top row: Particle representation of the coupled simulation. Bottom row: Rendering of the reconstructed surface of the coupled simulation.

methods like Smoothed Particle Hydrodynamics (SPH) have some advantages over mesh-based approaches, particularly in their ability to handle interfaces with other materials. Standard SPH is suitable for the simulation of compressible fluids. However, most fluids we encounter in nature feature incompressible behavior, which means that enforcing incompressibility is essential to produce realistic animations. We employ Incompressible SPH (ISPH) for simulating these fluids, which is a variant of SPH that is suitable for this purpose [20]. Numerical approximation issues with the standard SPH method are rectified by incompressible SPH. Furthermore, the recent work by Chow et al. [17] promises much faster Poisson solvers on GPUs than previously achieved in incompressible SPH simulations. Since the pressure equation in incompressible SPH is essentially the Poisson equation discretized as a sparse system of linear equations, the increased interest in solving linear systems makes incompressible SPH very promising and much more attractive than conventional weakly compressible SPH.

Although SPH is able to simulate complicated free boundaries such as splashes and droplets [66] when ghost particles are seeded, incompressible SPH suffers from a long-standing problem in handling free surfaces [11]: The pressure equation that is solved to obtain the pressure field requires a constant pressure boundary condition (BC) at the free surface, which is often applied by ad-hoc identification of particles, and this leads to inaccurate computation of the pressure forces at the interface. To simulate deformable objects, several approaches can be used, from efficient methods such as mass-spring systems to more accurate methods like *Finite Element Methods (FEMs)*. We decide to employ *Position Based Dynamics (PBD)* [53] to handle deformable bodies at high frame rates. Thus, the choice of solvers for the liquid and solid domains are deliberate, considering both the computing cost and the accuracy that translates to visually plausible physics.

Contributions: We propose a practical and efficient method for simulating two-way coupling between a divergence-free incompressible SPH fluid and a deformable solid simulated by PBD. To tackle the free surface issues in incompressible SPH, we propose a novel free surface formulation that handles the air-liquid interface. This is done by imposing a Dirichlet boundary condition for pressure at the free surface through modifying the leading diagonal terms of the coefficient matrix of the linear system obtained from discretizing the Pressure Poisson Equation (PPE). This enables an accurate computation of pressures for the particles at the interface, and these pressures are used to determine the forces exerted in the coupling. We couple the fluids and deformable solids by modelling the interaction of the forces that both exert upon each other. Employing PBD for simulating the elasticity of deformable bodies provides controllable dynamic behaviour, and guarantees stability over the deformable solids. However, our two-way coupling works with any deformation technique, as long as the deformable body is represented by a polygonal mesh. Our interaction model can achieve interactive-rates, and can take care of a highly dynamic coupled system like a water balloon, where small changes in the state of the deformable body cause almost instantaneous changes in the fluid, and vice versa (Fig. 7). A highlight of our coupling method is the ability to handle thin deformable shells and to avoid leakage, which is a rather complex problem in the case of two-way coupling (Fig. 1 and Fig. 14).

2 Related work

In this section, we review literature with a focus on the interaction between fluids and deformable solids. For a

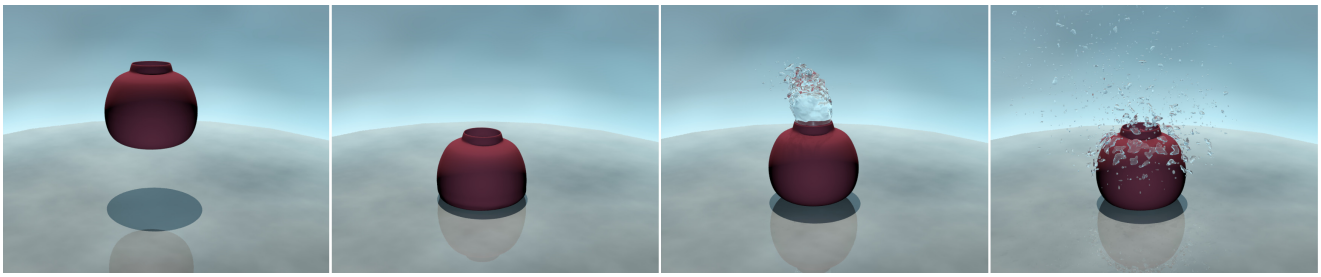


Fig. 2 An elastic flower pot filled with liquid falls on the ground ($21K$ PBD constraints and $27K$ fluid particles, at 28 fps), deforms as it hits the ground, bounces, and squirts the liquid out.

more thorough treatment on deformable bodies and fluid simulation, we refer the reader to the excellent references [12, 13, 34, 37, 57].

Eulerian methods have been used in computer graphics for simulating water [16], soft objects, melting effects [6, 61], and fluids in general [25]. The simulation of different materials by treating solids as highly viscous or visco-elastic Eulerian fluids was first presented by Goktekin et al. [30]. Carlson et al. [14] proposed a two-way coupling between a fluid and a rigid body, where they used a splitting operator that constrains the fluid velocities within the solids to behave rigidly at the end of each time step. However, this method uses a two-step projection approach that leads to visual artifacts and fluid loss. Chentanez et al. [15] avoided some of these artifacts by enforcing coupling and incompressibility constraints, while combining both the pressure projection and implicit integration steps into one set of simultaneous equations. Guendelman et al. [31] proposed to handle thin deformable and rigid shells coupled with fluids by using a ray casting technique, which increases the number of interpolations and prevents fluid from leaking through a triangulated surface. Later, Batty et al. introduced a variational approach, which provides a robust solution on relatively coarse Cartesian grids [5], allowing faster coupling between fluid and an arbitrary solid. The possibility of simulating hyper-elastic solids within an Eulerian framework allows to explore the two-way coupling between deformable bodies and fluid in a fully Eulerian fashion [24, 42]. These methods enable the simulation of deformable solids within an Eulerian framework, as was done by Robinson-Mosher et al. [62, 63], where they used sophisticated geometric operations inside the coupling scenario. Recently, Teng et al. [70] presented a solver that couples an incompressible fluid to multiple deformable objects undergoing frictional contact. By using an implicit time integration scheme, their method is able to resolve complex contact scenarios and can handle large time steps. Another two-way coupled simulation has been achieved by coupling incompressible fluids to reduced deformable bodies, using the method proposed by Lu et al. [45].

Zarifi and Batty [73] presented an Eulerian approach that simulates the coupling between fluids and deformable bodies. It uses a tetrahedral Lagrangian representation for the solid, where the solid should have a certain lower bound on the thickness, thus it cannot handle thin shells. Moreover, Akbay et al. [1] introduced an extended partitioned method (XPM) for two-way solid-fluid coupling of incompressible fluids to rigid and deformable solids and shells, where reduced models are employed to stabilize the convergence of the coupling, while using a partitioned approach.

Although Eulerian methods have as advantages less computation time and an easy management of topological changes, fluid properties such as pressure and velocity fields are limited by the grid resolution. Therefore, for very rapid and detailed flows, tracking changes as they occur at a fixed point in space appears unrealistic. In addition, grid-based techniques often suffer from mass loss at interfaces, and have dissipation no matter how formally accurate they are.

On the other hand, recent Eulerian-based methods such as Fluid Implicit Particle methods (FLIPs) are widely used to simulate special effects from splashes to flooding. Such methods are suitable for fluids with low viscosity and are therefore very well-suited for water effects. However, it results in unwanted visible noise on the surface in case of high viscosity. Furthermore, FLIP methods require a dense sampling of the fluid domain with particles, thus yielding expensive simulation. They are potential to mass gain due to a reseeding of FLIP particles [29]. Level set methods can indeed handle domain fragmentation, however the smallest fragments possible are of the order of a few grid cells [44]. Hence situations like breaking waves and sprays are not visually accurately captured by the method efficiently. Also, level set methods with high density ratio are still a problem, which is being researched widely. These two issues are tackled by SPH—sprays are resolved up to the particle dimensions and the density ratio for free surface flows is infinite, closely resembling an air-water system.

Lagrangian methods have the ability to solve the fluid equations of motion directly on the fluid particles. Also, they trivially guarantee mass conservation and provide a conceptually simple simulation framework. In Lagrangian methods such as Smoothed Particle Hydrodynamics (SPH), fluid properties

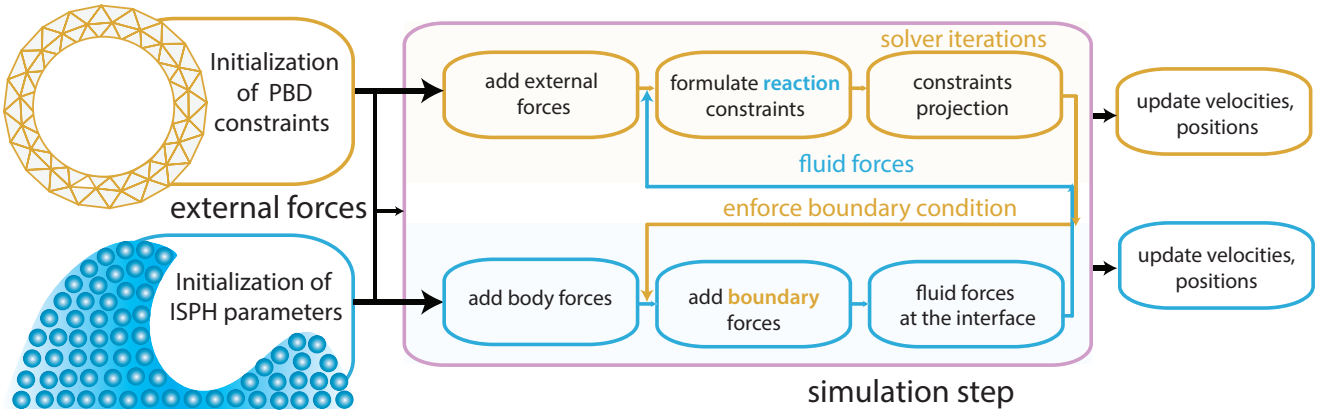


Fig. 3 System overview. *Top row:* The deformable body dynamics mechanism, where the deformable solid is represented by a mesh. *Bottom row:* The fluid simulation, in which the fluid positions, velocities and internal forces are carried by the particles (fluid forces include external, viscous, and pressure forces). During the simulation, the deformable solid and fluid exert forces on each other. Both pressure force (ISPH) and constraint projection (PBD) operations are iterated within each time step (where an iteration is depicted by the blue and yellow interconnecting arrows), until a convergence criterion is achieved, which is checked by verifying whether the velocity attribute of the fluid matches the velocity of the deformable solid.

like mass, density, and velocity are carried by moving virtual particles, which are tracked during the simulation [18]. Desbrun and Cani [22] were among the first to introduce smoothed particle systems to the Computer Graphics community, where they used smoothed particles for simulating highly deformable objects. Later, Müller et al. [52] have popularized particle-based methods for simulating fluids and the interaction between fluids and deformable bodies at interactive rates [54]. Premože et al. [60] also obtained a realistic looking fluid simulation by solving the Navier-Stokes equations based on the Moving Particle Semi-Implicit (MPS) method proposed by Koshizuka and Oka [40]. An interesting unified particle-based coupling of a fluid and thin deformable shells has been presented by Lenaerts and Dutré [41], in which SPH is used to simulate fluids, deformable volumes, and rigid volumes. However, they do not simulate incompressible SPH. The simulation of incompressible fluids by means of the traditional SPH method is limited by small time steps determined by the speed of sound in the near-incompressible liquid. Many works have successfully addressed this practical limitation of standard SPH implementations. One approach is to use prediction-correction schemes for the correct approximation of the pressure forces [10, 68]. Other works suggested implicit formulations when solving the Pressure Poisson Equation (PPE) [3, 19, 33] or iterative schemes to compute the density of the fluid [47]. Recent works proposed stable methods which inherently maintain both a divergence-free velocity field as well as constant density [9, 23, 38]. The interaction between a fluid and a deformable object within the SPH framework occurs at the interface, where the fluid exerts pressure forces at the deformable object while the deformable object imposes boundary fluxes on the fluid. Génevaux et al. [28] studied coupling of fluids and elastic bodies represented by particles and springs, respectively. Harada et al. [32] also proposed a simulation method for the coupling of cloth and fluids computed by using SPH. However, their algorithm is intricate and computationally complex. More recently, Koschier and Bender [39] couples solids and fluids, where they use pre-processed density maps to handle non-penetration constraints. Although, this method robustly handles rigid dynamic boundaries, it can not handle deformable bodies.

Akinci et al. [2] presented a simulation of coupled phenomena emphasizing an SPH unified approach and implicit treatment of coupling forces. Their approach combines SPH forces with the explicit collision handling scheme of Bell et al. [8] and applies position correction to prevent leakage in case of large deformations. Their method requires sampling the boundary of a triangle mesh with particles to prevent undesired fluid leakage. Macklin et al. [47, 49] proposed to handle fluid coupling with rigid and deformable bodies in real-time. In their unified framework, enforcing incompressibility depends on solving an iterative scheme to compute the density of the fluid. This scheme formulates an artificial pressure term, and is sensitive to the number of iterations, which needs to be carefully set up to avoid clustering of the particles. Most recently, Peer et al. [59] presented a two-way fluid-solid coupling that uses the implicit SPH approach [33] to simulate both the deformable solid and fluid. Their method can allow large time steps, although the current formulation of their method has difficulties with achieving an interactive rate.

In contrast to the methods mentioned above, we employ incompressible SPH for simulating fluids, while using Position Based Dynamics (PBD) to simulate the deformation of soft objects. Our method uses divergence-free incompressible SPH with a novel free surface formulation. We use PBD for simulating deformable bodies, which provides more con-

trollability over the final deformation. We efficiently model the interaction between the fluid and the deformable body by the forces that both phases exert upon each other. The works of Akinci et al. [2] and Macklin et al. [49] are both closely related to our method, and we provide a detailed comparison with their techniques in Sec. 9.

3 Overview

We present ISPH-PBD: a two-way interaction model between particle-based fluids and mesh-based deformable bodies. The inputs of our method are surface meshes representing the deformable solids, and a state list of the Incompressible Smoothed Particle Hydrodynamics (ISPH) particles representing the fluids. At the interface, the mesh nodes of the deformable solids function as boundary particles for the ISPH solver.

In the initialization phase, for the deformable solid we generate a volumetric tetrahedral mesh according to the input surface mesh using the method presented by Si [67]. Then, the vertices of the original deformable mesh are mapped to tetrahedral elements, and the tetrahedral elements (or triangles, in the case of thin shells) are used for defining the geometric constraints within the Position Based Dynamics (PBD) framework. These geometric constraints are used for emulating elastic behaviour, and provide the permissible volumetric strain in order to mimic the bulk elastic response of the 3D deformable models. This initial step of generating the tetrahedral mesh is not required if we are only simulating thin shells.

For the fluid, we initialize the particle distribution, particle mass, and other parameters, and we simulate the fluid using divergence-free ISPH. One of the problems in ISPH methods is the handling of free surface boundary conditions. To tackle this issue, we propose a novel free surface model in Sec. 5, which efficiently handles free surfaces representing the air-liquid interface. Our semi-analytic pressure boundary condition at the free surface enables accurate computation of the pressures at the interface, which is later used in the computation of the pressure forces for the coupling model. To the best of our knowledge, we couple ISPH and PBD successfully for the first time. Fig. 3 provides a visual overview of our algorithm for a single time step, and the pseudocode is detailed in Algorithm 1. During the simulation, at the beginning of each time step, the deformable bodies and fluids are propagated separately. We then exchange momentum between the deformable solid and fluid by transferring the forces and velocities at the interfaces, thereby enforcing kinematic boundary conditions (Sec. 6). PBD is a position-based deformable solver, where the vertices represent mass points and edges represent constraints [36, 53]. Thus, the boundary condition forces are formulated from the PBD constraints. Both the pressure and constraint projection processes are iterated within each time step (where an iteration is depicted by the blue and yellow interconnecting arrows in Fig. 3) until a convergence criterion is achieved. The latter is checked by verifying whether the velocity of the fluid matches the velocity of the deformable solid. This convergence criterion has to satisfy a certain threshold, which we discuss in Sec. 8.

4 Technical Background

Our method has as main components two basic and separate approaches: the deformable bodies are simulated by Position Based Dynamics (PBD), and the fluids are simulated using the Incompressible Smoothed Particle Hydrodynamics (ISPH) approach. We first briefly summarize the core ideas of the two methods, and the way they are used in our approach.

4.1 Deformable bodies simulation

In PBD, the physical system is modelled through equations governing external and internal forces that are applied to deformable solids, but these equations are formulated as a set of constraints [53, 64, 65]. PBD avoids the use of internal forces, and the positions are updated such that the angular and the linear momenta are implicitly conserved [48]. In this way, the process is not affected by the typical instabilities of interactive physics-based methods. The deformable bodies in PBD are modelled as a set of n *PBD-particles*[1] whose motion is governed by a set of m non-linear geometric constraints. Each PBD-particle $\mathbf{p}_i$ corresponds to a vertex in the input mesh, and a functional relationship C_j between PBD-particles is applied as a geometric constraint. The set of constraints is composed of non-linear equalities and inequalities such that:

$$C_i(\mathbf{p}) \succeq 0, \qquad i = 1, \dots m, \tag{1}$$

where the symbol $\succeq$ stands for either $=$ or $\geq$, $\mathbf{p} = \left[\mathbf{p}_1^T, \dots, \mathbf{p}_n^T\right]^T$ is the vector of PBD-particle positions, n is the number of PBD-particles and m is the number of constraints.

The set of constraints must always be satisfied, or at least, the error should be as small as possible. The constraints can be solved sequentially through Gauss-Seidel iterations. For an extensive treatment on how the constraints are solved within PBD, please refer to the excellent survey by Bender et al. [36].

Solving each constraint using Gauss-Seidel in a serial way is efficient when the number of constraints is relatively small. In our implementation, the constraints are solved in parallel using a graph coloring algorithm. Parallelizing the computation of the PBD constraints yields a fast performance even in case of a fairly high number of constraints [27].

[1] We use the term *PBD-particle* to distingish explicitly from the particles used by SPH.

4.2 Fluid simulation

In SPH methods, the fluid domain is discretized into particles of constant mass, which are treated as interpolation points. The discrete attributes at the interpolation points are smoothed to obtain a continuous field. The interpolation function $A(\mathbf{x})$ is defined as

$$A(\mathbf{x_a}) = \int A(\mathbf{x_b}) W(\mathbf{x_a} - \mathbf{x_b}, h) \quad d\mathbf{x_b}, \tag{2}$$

where W is the normalized radially symmetric smoothing kernel, h is the smoothing length of the kernel, and $\mathbf{x_a}$ is the position of the particle a where the property A is to be evaluated. Furthermore, W acts as the weighting factor for the contributions from the neighborhood interpolation points denoted by $\mathbf{x_b}$, where $d\mathbf{x_b}$ is the differential volume represented at $\mathbf{x_b}$. The above interpolant in SPH is approximated by a summation over the surrounding particles, in which the integral operator is replaced by a summation operator, and the differential volume element $d\mathbf{x_b}$ is replaced by the volume $V_b = m_b/\rho_b$. Thus,

$$A(\mathbf{x_a}) \approx \sum_b A(\mathbf{x_b}) W(\mathbf{x_a} - \mathbf{x_b}, h) \frac{m_b}{\rho_b}, \tag{3}$$

where the properties at discrete points given by the subscript b (such as the mass m_b, density ρ_b, and position $\mathbf{x_b}$ within a finite neighborhood of the point $\mathbf{x_a}$) are used for the discretization. Spatial derivatives of the attribute to any order can be defined using the discrete smoothing operator given by Eq. (3) [50].

Governing equations. The momentum conservation equations for a Newtonian fluid are solved using the SPH method in the Lagrangian framework. Therefore, the Navier-Stokes equations governing the momentum conservation of incompressible isothermal flow is given by the following equation:

$$\frac{d\mathbf{u}}{dt} = \frac{1}{\rho}\left(-\nabla p + \nabla \cdot (2\mu \mathbf{D}) + \mathbf{f}^B\right), \tag{4}$$

where $\mathbf{u}$ is the velocity, p is the pressure, ρ is the density, μ is the coefficient of viscosity of the fluid, $\mathbf{D} = (\nabla \mathbf{u} + \nabla \mathbf{u}^T)/2$ is the deformation rate tensor, $\mathbf{f}^B$ is the body force per unit mass on the fluid element, and t is the time. In the Lagrangian formulation, d/dt is the material derivative following a fluid element. The mass conservation equation is defined by:

$$\frac{1}{\rho}\frac{d\rho}{dt} + \nabla \cdot \mathbf{u} = 0. \tag{5}$$

In the case of incompressible flows, the material derivative of the density is zero, therefore, the condition for incompressibility is given by $\nabla \cdot \mathbf{u} = 0$.

SPH formulation. The SPH discretization of the governing equation Eq. (4) can be solved using a projection method [20], in which the pressure needed to enforce incompressibility is found by projecting an estimate of the velocity field onto a divergence-free space. Such discretization is approximated as follows:

$$\left.\frac{d\mathbf{u}}{dt}\right|_a = \sum_b \left[\left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2}\right)\nabla_a W_{ab} + \frac{\mu}{\rho_a} F_{ab} \frac{\mathbf{u}_{ab}}{\rho_b}\right] m_b + \mathbf{f}_a^B, \tag{6}$$

where ∇W_{ab} is the gradient of the kernel function for the displacement $\mathbf{x}_{ab} = \mathbf{x}_a - \mathbf{x}_b$ between two particles a and b, and $F_{ab} = (\mathbf{x}_{ab} \cdot \nabla_a W_{ab})/(\mathbf{x}_{ab}^2 + \varepsilon^2)$ is the radial derivative of the kernel, where ε is a parameter to avoid division by zero when two particles come very close to each other. The first term on the right hand side approximates the pressure gradient and is a symmetric formulation that conserves momentum [50]. The second term on the right hand side approximates the viscous force [51] and the third term on the right is the body force term.

To ensure incompressible flows, pressure cannot be related to density by an equation of state, hence the pressure serves merely to maintain zero divergence of velocity throughout the simulation domain. This hydrodynamic pressure is obtained by solving the following Pressure Poisson equation (PPE) [20] implicitly (in time) on the SPH domain:

$$\nabla \cdot \frac{\nabla p}{\rho} = \frac{\nabla \cdot \mathbf{u}}{\Delta t}, \tag{7}$$

where Δt is the time step as a result of temporal discretization. The discretization of the right hand side is given as:

$$\left.\frac{\nabla \cdot \mathbf{u}}{\Delta t}\right|_a \approx -\frac{1}{\rho_a \Delta t} \sum_b m_b (\mathbf{u}_a - \mathbf{u}_b) \nabla_a W_{ab}. \tag{8}$$

5 Free Surface Boundary Conditions

In this section, we present our novel analytical solution that effectively handles free surfaces. In incompressible SPH methods, when applying the pressure Poisson equation (PPE) to free surface flows, we need to impose a Dirichlet boundary condition for pressure at the free surface. One way to do this is by explicitly identifying the particles at the free surface, based on the change in the density or an ambient pressure value (usually zero). Such a treatment is known to affect the accuracy of the pressure computation, which we use for modelling the interaction of forces between the fluid and deformable solids. Our solution is rather to impose a constant pressure boundary condition on free surfaces by analytically computing the kernel contribution of the region that is falling outside of the free surface. Thus, the zero pressure Dirichlet condition can be applied accurately. In this case, the zero

pressure is applied right outside the free surface by identifying kernel deficiency of particles *near* the free surface, rather than *at* the particles on the surface. This is achieved by a modification of the leading diagonal terms of the coefficient matrix of the linear system obtained by discretizing the PPE as explained below.

The second order derivative approximation based on the finite difference is used for approximating the Laplacian terms in the PPE Eq. (7). Therefore, the second derivative approximation for the PPE in incompressible SPH is given by [20]:

$$\nabla\cdot\left(\frac{\nabla p}{\rho}\right)_a = \sum_b \frac{m_b}{\rho_b}\frac{4}{\rho_a+\rho_b}(p_a-p_b)F_{ab}, \tag{9}$$

where F_{ab} is the radial derivative of the kernel smoothing function W, and p is the pressure.

Let R represent all particles in the full kernel of a surface particle a. Let R_i and R_o be the regions inside the fluid and outside the fluid respectively, that fall within the kernel of a particle, as illustrated in Fig. 4, so that $R = R_i \cup R_o$. Let p_a represent the pressure corresponding to the particles in region R_i and let p_o represent the pressure corresponding to the (implicit) particles in region R_o.

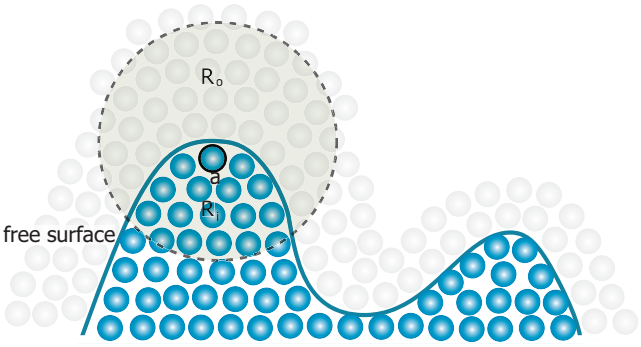


Fig. 4 Illustration of our free surface approach. The symbol R_i represents the region within the fluid, while R_o represents the region outside the fluid. Both R_i and R_o are overlapping within the kernel region of the particle a near the interface (the highlighted circle), in which free surface flow is simulated as discussed above. The grey particles represent the air, which are implicitly modeled in the simulation process.

For a full kernel of a particle near the interface, with the kernel deficiency complemented with implicit particles of zero pressure, we have:

$$\nabla\cdot\left(\frac{\nabla p}{\rho}\right)_a = \sum_b \frac{m_b}{\rho_b}\frac{4}{\rho_a+\rho_b}p_aF_{ab} - \sum_{b_i}\frac{m_b}{\rho_b}\frac{4}{\rho_a+\rho_b}p_bF_{ab} - \sum_{b_o}\frac{m_b}{\rho_b}\frac{4}{\rho_a+\rho_b}p_oF_{ab}, \tag{10}$$

$$\nabla\cdot\left(\frac{\nabla p}{\rho}\right)_a = (p_a-p_o)K - \sum_{b_i}\frac{m_b}{\rho_b}\frac{4}{\rho_a+\rho_b}p_bF_{ab} + \sum_{b_i}\frac{m_b}{\rho_b}\frac{4}{\rho_a+\rho_b}p_oF_{ab}, \tag{11}$$

where

$$K = \sum_b \frac{m_b}{\rho_b}\frac{4}{\rho_a+\rho_b}F_{ab}. \tag{12}$$

The value of K is found for any interior particle in the initial regular configuration of particles when errors are expected to be minimal. In the coefficient matrix for the PPE, K appears in the leading diagonal position corresponding to every particle. This imposes the zero Dirichlet condition implicitly by choosing $p_o = 0$. We easily implement this by making a simple modification to the coefficient matrix of the PPE. This analytical solution allows the Dirichlet condition to be applied accurately, which is important to avoid incorrect pressure values in incompressible SPH. This effects the computation of the pressure force, which is used for the coupling with the deformable solids.

K is an approximation for the term:

$$\int_\Omega \frac{2}{\rho}\frac{dW}{d\mathbf{x}}dV, \tag{13}$$

where Ω is the domain volume V within the cut-off radius of a given particle position $\mathbf{x}$, when the kernel is fully supported. It remains approximately constant throughout a simulation where incompressibility is ensured. Please note that this stems from the fact that the number of neighbors of a particle in an incompressible simulation (with given discretization parameters) remains approximately constant throughout the simulation.

In this free surface formulation, if only b_i particles are included and if no diagonal correction is made, then this would amount to a truncated kernel for all particles near the free surface and hence is approximately equivalent to a homogeneous Neumann boundary condition for pressure. This means that such a formulation might result in artifacts. In reality, the free surface is a Dirichlet boundary condition, which is imposed by the diagonal correction.

6 Two-way coupling of fluid and deformable bodies

In this section, we introduce our interaction simulation between fluids and deformable bodies. For incompressible SPH, pressure is used to enforce incompressibility, which we also employ in the coupling between fluid and deformable solids. In ISPH, a reasonable boundary condition implementation

is to use static boundary particles for the deformable bodies. When a difference approximation is used for divergence computations, these particles at the interface of the deformable body implicitly impose a Neumann boundary condition [31]. In this configuration, the deformable bodies within the PBD framework consists of particles and a set of constraints, where each *PBD-particle* has mass, position and velocity. We enforce fluid pressure to exert forces on PBD particles in contact with the fluid, by formulating the pressure projection in order to map the pressure to a net force on the deformable body. The net force in this case is defined as:

$$\mathbf{F}_{pressure} = -\frac{1}{\rho}\nabla p, \tag{14}$$

where ρ is the density and ∇p is the pressure gradient.

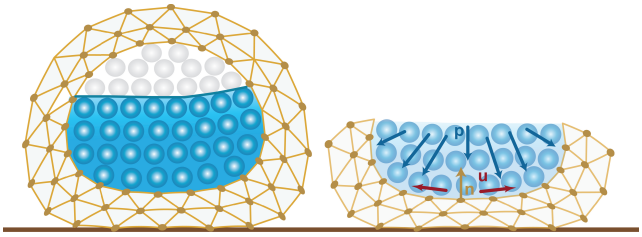
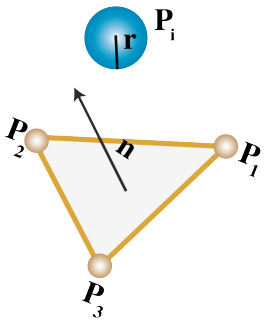


Fig. 5 Illustration of the coupling situation between fluids and deformable solids. *Left:* 2D cut through a 3D mesh with both free surface interface and liquid-deformable solid interactions. *Right:* Depiction of the relevant interaction components: **p** is the fluid pressure, **n** is the outward normal on the solid, and **u** is fluid velocity.

On the basis of Newton's third law of motion, the symmetric interaction force applied by a fluid particle $\mathbf{p}_i$ on its adjacent triangle vertices is $\mathbf{F}_{solid\leftarrow fluid} = \mathbf{F}_{pressure}$ and $\mathbf{F}_{solid\leftarrow fluid} = -\mathbf{F}_{solid\rightarrow fluid}$. However, fluid penetration through the boundary occurs when $\mathbf{u}\cdot\mathbf{n} < 0$, where $\mathbf{u}$ the velocity corresponding to a fluid particle at the boundary and $\mathbf{n}$ is the outward normal of the solid into the fluid as shown (Fig. 5), assuming a boundary that is not moving. In the case of deformable solid where the boundary is moving, this generalizes to the constraint $\mathbf{u}\cdot\mathbf{n} > \mathbf{v}\cdot\mathbf{n}$, where $\mathbf{v}$ is the velocity vector of the boundary. We might also want to enforce the no-slip condition, which means that the fluid particles at the boundary do not freely slip or slide over the surface. Below we describe how to address the non-penetration and no-slip conditions at the boundary.

In each time step, the new positions and velocities are updated based on the results of the previous time step. To prohibit penetration of the fluid particles into the deformable body boundary, we detect collisions based on the updated positions of PBD particles and fluid particles. Then, we formulate non-penetration constraints between the fluid and deformable body. In order to detect collisions, we employ a two-grid cell-based spatial hashing procedure with temporal marks based on [71]. The two-grid cell-based spatial hashing data structure (which is two-layer grids) requires more construction time than the original spatial hash. However, it performs 20% better during the simulation.



To ensure satisfied boundary conditions, we formulate non-penetration constraints once a collision is detected. To enforce such non-penetration constraints between the fluid and deformable solid, a fluid particle $\mathbf{p}_i$ should stay a certain distance away from all the triangles, which is at least equal to its radius r. Let d_{ij} be the distance between particle $\mathbf{p}_i$ and triangle j. When penetration occurs, the fluid particle moves a distance of at least $d_{ij} - r$ towards the triangle (where $d_{ij} - r$ is the distance from the particle along the normal $\mathbf{n}$ in the direction dir of the triangle). Therefore, the correction is done by projecting the fluid particles position to $\|r - d_{ij}\|n$, in which we take the position of the penetrating particle and move the particle back along the normal to a point at a distance of $d_{ij} - r$ from the triangle. When a particle collides with multiple triangles with different normals, we compute the direction dir by taking into account the average of normals of the collided triangles. Furthermore, to prevent penetration artifacts during the deformation of the deformable solid, we define non-penetration constraints within the PBD framework. For a fluid particle $\mathbf{p}_i$ and its adjacent triangle ($\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$) of the object surface mesh, we define a position constraint as:

$$\begin{aligned} &C_{penetration}(\mathbf{p}_i, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) \\ &= \mp(\mathbf{x}_i - \mathbf{x}_1) \\ &\cdot \frac{(\mathbf{x}_2 - \mathbf{x}_1)\times(\mathbf{x}_3 - \mathbf{x}_1)}{|(\mathbf{x}_2 - \mathbf{x}_1)|\times|(\mathbf{x}_3 - \mathbf{x}_1)|} - Z_0, \end{aligned} \tag{15}$$

where $Z_0 = d_0 + r$, and where d_0 is the rest distance of the fluid particles (i.e., the distance by which a fluid particle should be removed from the solid boundary, in addition to its radius), r is the fluid particle radius, and $\mathbf{x}_i, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are the fluid particle position and the triangle vertex positions respectively. If the fluid particle $\mathbf{p}_i$ penetrates the triangle ($\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$) from below with respect to the triangle's normal, the first term of the right hand side of Eq. (15) picks the minus. Conversely, the first term picks the plus. According to the PBD method, the position change for a fluid particle and the triangle vertices is calculated as:

$$\Delta\mathbf{p}_\mathbf{c} = \lambda_i \nabla_{\mathbf{p}_\mathbf{c}} C_{penetration}, \tag{16}$$

where $c \in [i, 1, 2, 3]$, and λ_i is a scalar value computed by the PBD method.

In case the no-slip condition is required, it is imposed by the constraint $\mathbf{u}\cdot\tau = \mathbf{v}\cdot\tau$, where $\mathbf{u}$ the velocity corresponding to a fluid particle at the boundary, $\mathbf{v}$ the velocity of the solid boundary and τ is the tangent at the boundary. To address this condition, we add a tangential friction term for each particle which collides against a triangle ($\mathbf{u}\cdot k\tau$, where k is a coefficient of friction and it is a positive constant). This friction force is opposed to the current velocity of the particle, simulating the loss of energy when the two (fluid particle and triangle) interact. This no-slip law is phenomenologically reasonable and moreover has some justification in kinetic theory [46].

7 ISPH-PBD Simulation

At a given time step t, the vector of all fluid particle positions is $\mathbf{x}_t$, the vector of all fluid particle velocities is $\mathbf{u}_t$, and similarly the *PBD particle* positions and velocities are $\mathbf{p}_t$ and $\mathbf{v}_t$ respectively. The ISPH-PBD simulation works as described in Algorithm 1. In line 1, F_f^{Total} corresponds to $\mathbf{f}^B$ in Eq. 4, and in line 5, F_s^{Total} is the external force, mainly gravity. In lines 10-15, the pressure is computed by solving the linear system that is described by the PPE. Lines 25-28 refer to the PBD iterative solver that manipulates position estimates such that they satisfy the constraints as described in Sec. 4.1, where *Iterations* refers to a parameter specified by the user, indicating the number of PBD constraint solver iterations to be run. Another point within the outer loop where an iterative algorithm may be used is at lines 10-15: in the case of our simulations below, we approximate $F_{pressure}$ using BiCGSTAB, which runs for a number of iterations such that a certain bound on the volume loss is guaranteed (see the next section for more details). Hence, two iterative solvers are invoked at each iteration of the outer loop at line 9.

Algorithm 1 ISPH-PBD Simulation

1: compute forces F_f^{Total} ▷ (F_f^{Total}) forces acting on the fluid
2: $\mathbf{x}' \leftarrow \mathbf{x}_t + \Delta t\mathbf{u}_t$
3: $\mathbf{u}^* \leftarrow \mathbf{u}_t + \Delta t(v\mathbf{u}_t + F_f^{Total})$
4:
5: compute forces F_s^{Total} ▷ (F_s^{Total}) forces acting on the solid
6: $\mathbf{v}' \leftarrow \mathbf{v}_t + \Delta t F_s^{Total}$
7: $\mathbf{p}' \leftarrow \mathbf{p}_t + \Delta t\mathbf{v}'$
8:
9: **repeat**
10: approximate $F_{pressure}$ according to Eq. 14, where:
11: **for** each fluid particle $\mathbf{p}$ **do**
12: **if** $\mathbf{p}$ on the surface **then**
13: solve pressure $p \leftarrow$ with Eq. 11
14: **else**
15: solve pressure $p \leftarrow$ SPH approximation to Eq. 7
16:
17: $\mathbf{u} \leftarrow \mathbf{u}^* + \Delta t F_{pressure}$
18: $\mathbf{x} \leftarrow \mathbf{x}' + \Delta t(\mathbf{u}^* + \mathbf{u})/2$
19:
20: **for** each fluid particle j **do**
21: let $i_1, \ldots, i_k$ be the PBD particles in contact with j
22: add force $(F_{pressure})_j/k$ to $(F_s^{Total})_{i_1}, \ldots, (F_s^{Total})_{i_k}$.
23: $\mathbf{v} \leftarrow \mathbf{v}' + \Delta t F_s^{Total}$
24: $\mathbf{p} \leftarrow \mathbf{p}' + \Delta t\mathbf{v}$
25:
26: generate non-penetration constraints with Eq. 15
27: **for** *Iterations* **do** ▷ the constraints solver
28: $projectConstraints(\mathbf{C}_m)$
29: update velocities $\mathbf{u}$ and $\mathbf{v}$ from the positions ▷ based on the solver outcome
30:
31: **until** convergence ▷ this criterion is discussed in Sec. 8
32:
33: update the fluid velocities $\mathbf{u}_{t+1} \leftarrow \mathbf{u}$ and positions $\mathbf{x}_{t+1} \leftarrow \mathbf{x}$
34: update the solid velocities $\mathbf{v}_{t+1} \leftarrow \mathbf{v}$ and positions $\mathbf{p}_{t+1} \leftarrow \mathbf{p}$

8 Results

In this section, we demonstrate the capability of our method using different scenarios, where our choice of the ISPH parameters is described as follows: The smoothing radius r is set to 0.05, Δx is the initial particle spacing, and is set to 0.005, h is the smoothing length, and is set to 0.015, and the density ρ is set to 1000. We employ the Wendland Quintic kernel, also called the Wendland C2 kernel [21], where the results show that a good compromise between accuracy and time computation cost is reached by the use of the Wendland kernel. This kernel also has interesting properties, in particular it maintains a highly ordered particle distribution in dynamical simulations and it does not fall prey to the pairing instability. A graph plotting the accuracy of the pressure gradient estimation (as expressed in the first term of the summation in Eq. (6)) can be seen in Fig. 6, where the figure shows a comparison of three different kernels; Wendland C2, cubic spline, and quartic spline kernels over different choices of the smoothing parameter. Indeed, the Wendland C2 comes with good results, and continuously improves the gradient estimate as the value of the smoothing parameter increases.

8.1 Timings

In order to enforce incompressibility, we solve the PPE equations by a Krylov subspace method, in particular the Biconjugate Gradient Stabilized (BiCGSTAB) solver using the LIS linear solver library [58]. In Ihmsen et. al. [33], the Krylov subspace method used is the Conjugate Gradient method which is originally restricted to symmetric matrices. In our work, we use BiCGSTAB, a variant of CG specifically suited for asymmetric matrices. In our experience, BiCGSTAB solver tends to provide a solution even when multiple solutions are present (for example, when the matrix $\mathbf{A}$ is singular and when $\mathbf{b}$, the RHS vector, is in

Table 1 Various timing and constraint statistics of our simulations. The column names mean the following. #S: number of stretch constraints, #T: number of cloth balloon constraints, #B: number of bend constraints, #V: number of volume constraints, #PBD: total number of PBD constraints, #ISPH: number of particles used during the simulation, #ISPH iterations: Avg. number of ISPH iterations per time step when 0.025% error is tolerated, # Avg. iterations: average number of iterations in order to meet the convergence criterion, #CT: average computation time during a 1 second simulation.

Scene name	#S	#T	#B	#V	#PBD	#ISPH	#PBD iterations	#ISPH iterations	# Avg. iterations	time step	$CT_{total}[ms]$
Water balloon	$9K$	1	-	-	$9K$	$22K$	25	21.3	2.1	0.001	32.5
Double elastic clothes	$12K$	-	$9K$	-	$21K$	$95K$	12	30.4	2.5	0.005	81.6
Thin shell in glass basin	$7K$	-	$4K$	-	$11K$	$27K$	8	32.2	1.4	0.005	45.7
Shower on cloth	$13K$	-	$9K$	-	$22K$	$75K$	12	43.1	2.4	0.005	77.1
Deformable bodies	$15K$	1	-	$20K$	$35K$	$45K$	36	40.2	2.7	0.001	87.3
Flower pot	$9K$	-	-	$12K$	$21K$	$27K$	20	20.3	1.6	0.001	35.2

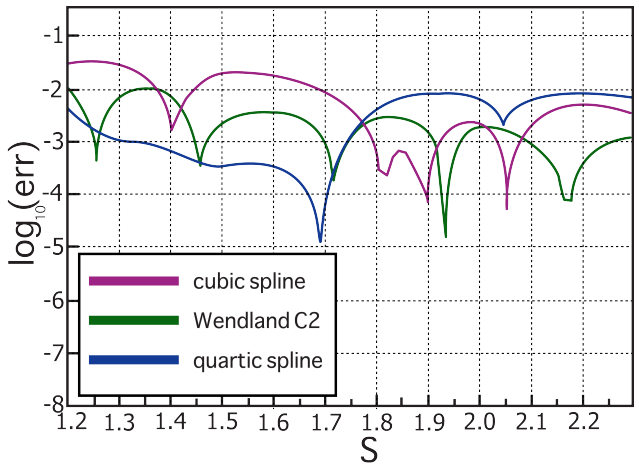


Fig. 6 Accuracy of pressure gradient estimates for three different kernels: cubic spline, quartic spline and Wendland C2 kernels obtained from particles that are placed in a simple cubic lattice. The particles have identical masses, are uniformly spaced, and the pressure increases linearly with the three axes, so that the true pressure gradient is constant. The error is measured as $(|\hat{g}-g|)/|g|$ where g denotes the true pressure gradient and $\hat{g}$ denotes the gradient as estimated by the SPH method under the respective kernel (in a single coordinate). We observe that the Wendland kernel performs particularly well in its gradient estimation ability, and improves the gradient estimate as the value of the smoothing parameter S increases. The smoothing parameter S determines the smoothing length h as $h = SV_b$, where V_b is the particle volume. The dips in the graph are caused by the use of the log-scale and the fact that we compare absolute values in our error measure (the difference $\hat{g}-g$ crosses 0 at these points.

the range of **A**), without the need to pin a point in the domain to a fixed value. Such situations arise when a fluid/fluid-solid domain is completely bounded by a solid boundary. Also BiCGSTAB is stable even for high condition numbers. Hence BiCGSTAB has been robust for our applications.

Table 1 summarizes our testing scenarios and run-times, where the tolerated error is set to 0.025% overall volume change of the fluid. All experiments described in this section have been performed on a MacBook Pro equipped with a 2,8 GHz quad-core i7 Intel and 16 GB of RAM. During the simulations, we iterate to verify whether the velocity of the fluid matches the velocity of the deformable solid in which an error with a threshold of 0.5×10^{-3} has to be satisfied as the convergence criterion. The reported times do not include rendering. In terms of computation time, the ISPH-component of the algorithm is the most expensive by an order of magnitude, when compared to the PBD-component. Note that we currently rely on very modest computer hardware, and we are using a CPU-implementation that is multi-threaded. Clearly, a GPU implementation would allow much larger scenarios and better performance.

8.2 Visual Results

Enforcing incompressibility. Fig. 7 shows the stability of our interaction model, where a two-way coupling simulation of an elastic balloon (modelled as a thin shell) is provided. The simulation consists of $9K$ geometric PBD constraints, and the balloon is filled with $22K$ ISPH particles. The PBD solver iteration count is set to 25 iterations. We employ a fixed time step of 0.001. The balloon bounces upon impact with the ground, expanding and deforming due to the fluid load. In contrast, the elastic behaviour of the balloon causes the water to flow inside and form waves, while enforcing a boundary condition on the fluid flow.

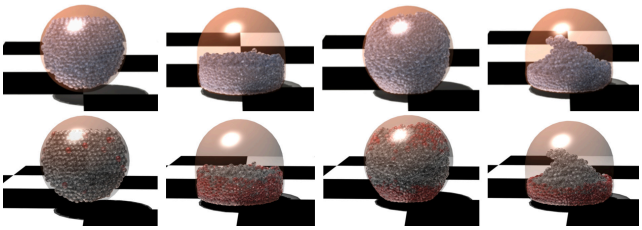


Fig. 7 *Top row:* Our method simulating a water balloon upon impact with the ground. *Bottom row:* The same simulation but with fluid particle pressures that are color-coded and proportional to their red saturation, where red indicates high pressure values ($9K$ PBD constraints and $22K$ fluid particles, at 30 fps).

Fig. 8 shows the adaptability of our method: it allows for tuning the elasticity of the deformable body. This is done by changing the stiffness of the constraints inside the

PBD method, so that we can reduce or increase the elasticity of the final deformation until the desired effect is reached.

Fig. 8 Comparison of the water balloon simulation with varying elasticity settings. These deformations are obtained by setting the stiffness to 0.9 (left) and to 0.75 (right).

Fig. 9 shows the ability of our method to preserve fluid volume, in comparison to the standard weakly compressible SPH method presented by Becker and Teschner based on the Tait equation [7], where the stiffness values in both are set to be 0.80. The volume loss resulting from using the weakly compressible SPH can be less if a larger stiffness value is used. In this case, the time step size for weakly compressible SPH must be decreased, which makes the simulation slow. In Table 2, we report the computation time comparison of WCSPH and ISPH for three different scenarios.

Fig. 9 *Left:* Simulation of a water balloon upon its impact with the ground using weakly compressible SPH, which suffers from fluid loss. *Right:* The same scene simulated using our method, which preserves the fluid volume during the simulation. The PBD-stiffness value of the elastic balloon in both simulations is set to 0.80 and stiffness value of WCSPH is 6.0×10^3.

Fig. 10 shows the error in the pressure for the water balloon scenario performed by weakly compressible SPH and ISPH over different numbers of particles. In order to compute this error, we employed high-order kernel support (e.g. 5th order B-spline kernel [43], which is well-known in the physics community for its ability to obtain minimum distortion error and well-maintain particle spacing) to get what we call the optimum pressure $\hat{p}$. Thus, the error is evaluated through the subtraction between $\hat{p}$ and the pressure obtained from the WCSPH and ISPH. Furthermore, in order to evaluate the volume preservation quality of our method, we simulate a fluid pillar of 5 meters that consists of $30k$ particles, where we show the error in the pressure distribution at different times (Fig.11). In this simulation, we measured the volume loss at the end of the simulation $t = 350$ to be 1.19×10^{-3}.

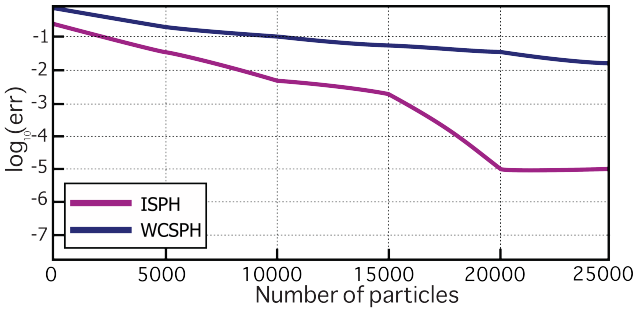


Fig. 10 Shows the error in the pressure over different numbers of particles between WCSPH and ISPH for the water balloon scenario, where the PBD-stiffness value of the elastic balloon in both simulations is set to 0.80.

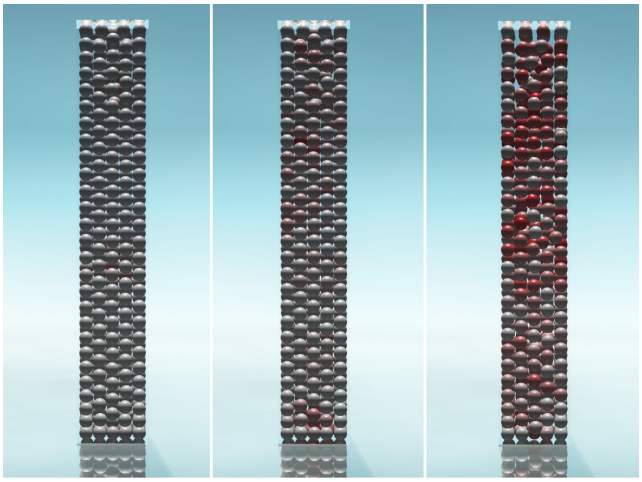


Fig. 11 A fluid pillar simulation of 5 meters height with $30k$ particles at different times, where the errors are color coded with red. The error is evaluated through the analytical solution $p = \rho \cdot gravity \cdot height$ and the pressure as estimated during the simulation. Note that at time $t = 250$ the oscillations have become visible due to errors. However, the particle arrangement at earlier times is stable.

Table 2 A computation time comparison of WCSPH and ISPH for three different scenarios, where the time step size is 0.001 and $CT_{total}\,[ms]$ is the total computation time for the whole motion.

Scene name	WCSPH $CT_{total}\,[ms]$	ISPH $CT_{total}\,[ms]$
Water balloon	77.9	42.1
Deformable bodies	123.7	95.5
Flower pot	94.4	39.18

Interaction with thin shells and avoiding leakage. Fig. 13 shows the interaction between elastic cloth and fluid, where the simulation consists of $21K$ cloth constraints and $95K$ fluid particles. The PBD solver iteration count is set to 12 iterations, where we employ a fixed time step of 0.005. The elastic cloth stretches under the water load, while it prevents water from leaking through the cloth. This simulation proves the capability of our approach to interact with elastic thin

shells, while avoiding the leakage of fluid through the boundaries.

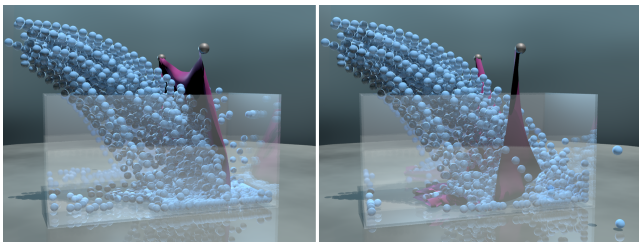


Fig. 12 A scenario where a large amount of water is sprayed on a thin shell in a glass basin, with varying elastic deformations. These deformations are obtained by setting the stiffness of the stretch constraints of the thin shell to 0.9 (left) and 0.6 (right).

Fig. 12 shows a simulation of a stream of fluid being propelled at high speed onto a piece of elastic thin shell, into a transparent basin. This simulation shows that our method enables tuning the elasticity of the thin shell. This simulation has $11K$ cloth constraints, and has $27K$ fluid particles. The PBD solver iteration count is set to 8 iterations, where we employ a fixed time step of 0.005. Fig. 14 shows a simulation of a stream of fluid being poured at high speed onto an elastic thin shell. This simulation shows the ability of our method to avoid leaking under large deformation and fast motion. In this simulation there are $22K$ cloth constraints, and $75K$ fluid particles, where we employ a fixed time step of 0.005. The PBD solver iteration count is set to 12 iterations.

Handling two-way elastic-fluid coupling. Fig. 15 shows the flexibility of our explained elastic-fluid coupling technique, where an elastic flower pot consisting of $21K$ PBD constraints is filled with a $27K$ ISPH particles. The elastic flower pot deforms as it hits the ground, bounces, and squirts the liquid out. In this simulation, the PBD solver iteration is set to 20 iterations. We employ a fixed time step of 0.001. Also, Fig. 16 shows the flexibility of our explained elastic-fluid coupling technique, where a lifebuoy and a ball consisting of $35K$ PBD constraints are floating on $45K$ ISPH particles. The PBD solver iteration is set to 32 iterations We employ a fixed time step of 0.001.

In Fig. 17, we illustrate that our method is able to couple fluid and deformable bodies with different densities, where we show an experiment that simulates floating and sinking. In addition, we demonstrate in Fig. 18 the capabilities of our method to emulate the free-slip and no-slip boundary conditions described in Sec.6. Please note that the coefficient of friction k in the no-slip simulation is set to 0.4.

9 Discussion

In this section, we discuss methods that address the two-way coupling between fluids and deformable bodies/ thin shells at interactive rate. We compare our method to the closest methods available, and we discuss the advantages and limitations of our method. Akinci et al. [2] and Macklin et al. [49] presented a two-way coupling simulation between fluids and deformable solids, including thin shells, which bears some similarities to our approach. The technique proposed by Akinci et al. [2] is a pressure-based coupling method, which employs boundary particles to represent deformable solids and effectively avoids the problem of leakage due to the high pressure ratios that these boundary particles cause. However, the method heavily relies on pre-sampling the solids and is very hard to generalize in order to handle solids with large deformations. In order to avoid leaking through boundaries when a large deformation occurs, their method tends to oversample the deformable meshes in a uniform fashion. In contrast, our method does not sample the boundaries with particles, and therefore simplifies the interaction model, avoids oversampling, and prevents leakage, although a fine triangulation is required in the case of thin shells.

The method presented by Macklin et al. [49] employs Position Based Fluid (PBF) [47] to simulate fluid, where enforcing incompressibility is not derived from the continuity equation, but from constraint dynamics. Therefore, the density constraints are enforced through the use of Lagrange multipliers and not by employing SPH formulations for pressure and pressure force. Although their method generates appealing simulation results at real-time, we observe the following: 1) The soft volumetric bodies are obtained by using a voxelized version of the original mesh, with a shape matching constraint, which *de facto* limits the deformation modes of the deformable body. The liquid particles unnaturally cluster in a shell when particles are in contact with a solid. This clustering issue cannot be eliminated even by reducing the viscosity coefficient and cohesion coefficient. In contrast, our method is based on an ISPH formulation that generates a more plausible fluid simulation, in which we avoid particle clumping. 2) Significant contact offset is used in [49], which sometimes causes a visible space between a liquid and a shell that are in contact with each other. In their method, this is necessary in order to prevent leakage through the thin shell. We are able to use a lower contact offset when handling contact, while still preventing leakage through thin shells (Fig. 19).

While we find our method practical, it comes with three main limitations. First, we are currently using PBD to simulate deformable bodies due to its stability and efficiency. However, PBD is rather sensitive to parameter tuning, and the parameters need to be tuned carefully to

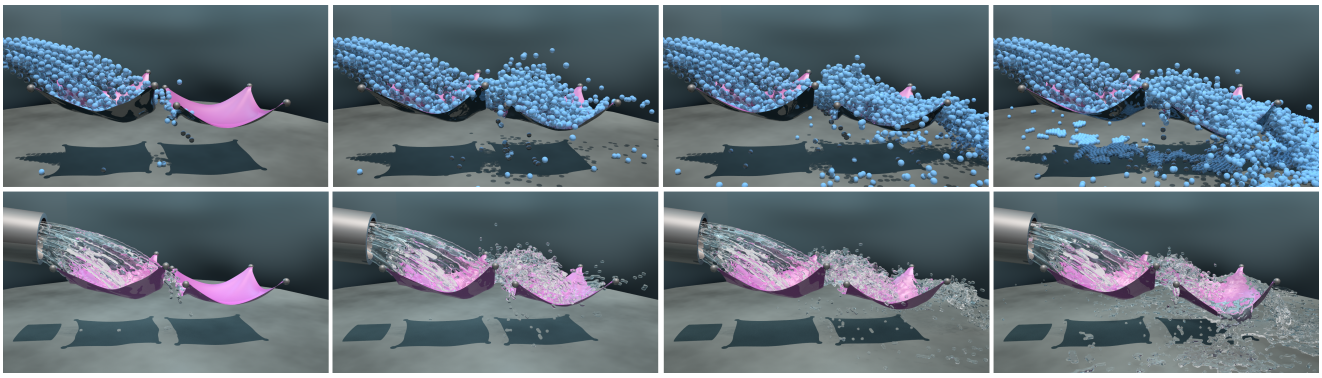


Fig. 13 Simulation of a large quantity of fluid being propelled at high speed onto two pieces of thin shell, where the elastic thin shells deform under the impact of the fluid, stretching and expanding, while the water splashes around and runs down off the cloth's sides ($21K$ cloth constraints and $95K$ fluid particles, at 12.2 fps).

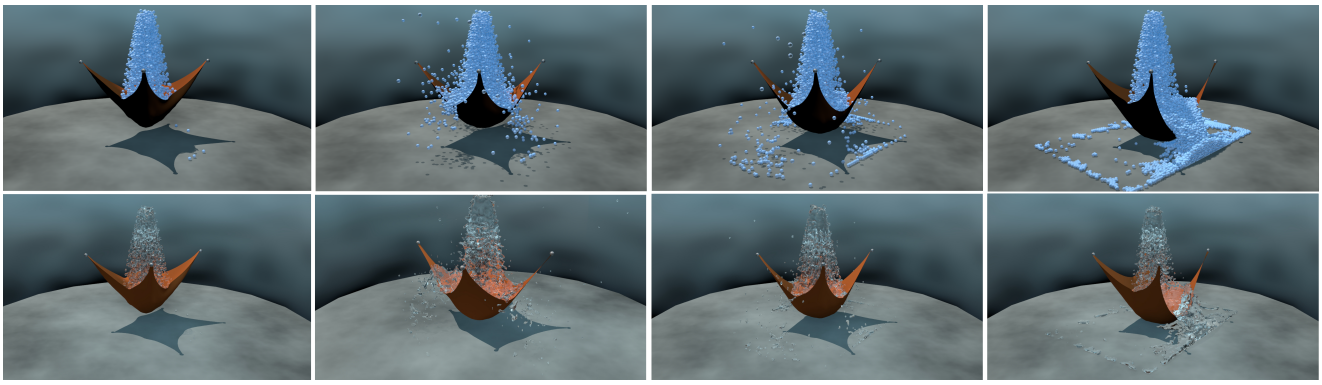


Fig. 14 A stream of liquid pours into an elastic sheet of thin shell. The elastic thin shell deforms under the impact of the water, stretching and expanding. This simulation shows that our method prevents leakage even in the case of rapid motion and large deformations ($22K$ cloth constraints and $75K$ fluid particles, at 12.8 fps).

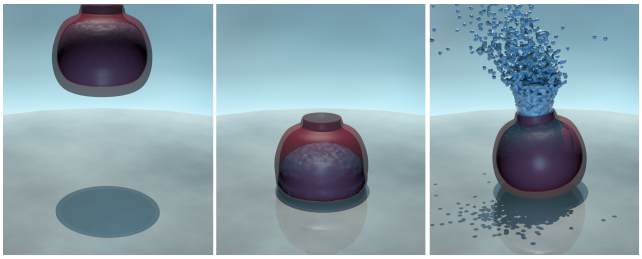


Fig. 15 An elastic flower pot filled with fluid, where the pot bounces on the floor and squirts the liquid out. Note that the pot is a deformable body (i.e., it is not a thin shell), as can be seen by the transparency of the pot in the picture.

achieve plausible deformations of the deformable solids. To avoid this problem, while still using PBD, an extended version of PBD (XPBD) has been proposed by Macklin et al. [48]. Secondly, our method relies on various parameters of the particle-based domain of SPH and the mesh-based domain of PBD, we argue that special attention is required while setting up the simulation and parameters, such as the smoothing radius and smoothing length for SPH, and the elasticity and stiffness for PBD. For instance, using a coarse triangulation for the deformable thin shell in a relatively small scale simulation leads to undesired results such as leaking. To address such issues, in future work we plan to investigate the possible use of adaptive SPH proposed by Winchenbach et al. [72]: The use of continuously adjusted particle masses and sizes with restricted spatial variation generates a very high degree of adaptivity. This can enable us to properly handle the continuous exchange of forces at the interface between the deformable bodies and fluids, while capturing small-scale phenomena, such as splashing and droplet formation. Moreover, it will be interesting as a future direction to investigate the possibility of employing the methods presented in [4, 9] to simulate the incompressible fluids in our framework. We would like to study the use of these approaches to maintain larger time steps, while achieving interactive rate performance.

The third limitation is as follows. The incompressible SPH formulation presented in Section 4.2 solves for $\nabla \cdot \mathbf{u} = 0$, but in the presence of discrete time stepping this does not also guarantee ρ to be constant. While density invariance or density invariance together with null divergence have both been used in incompressible SPH, the resulting system is known to have high pressure fluctuations. The volume error due to

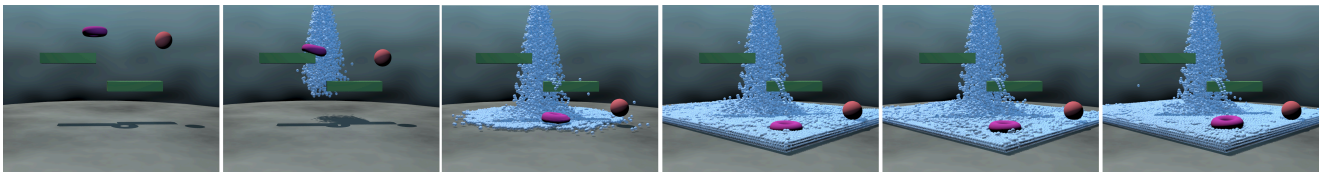


Fig. 16 A shower of fluids, which consists of $45K$ ISPH particles together with a lifebuoy and a ball of $35K$ PBD constraints. The lifebuoy and ball have different masses, and the lifebuoy is a bit heavier than the ball. While the fluid is filling a block volume, the ball floats, and the lifebuoy slowly rises to buoyancy.

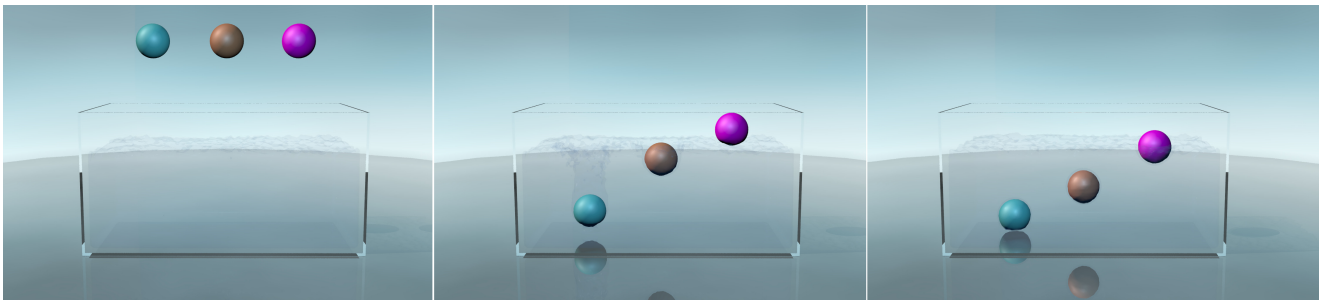


Fig. 17 To demonstrate buoyancy using our method, we drop three spheres with decreasing densities from left to right into a tank of liquid, where spheres are negatively, neutrally, and positively buoyant respectively.

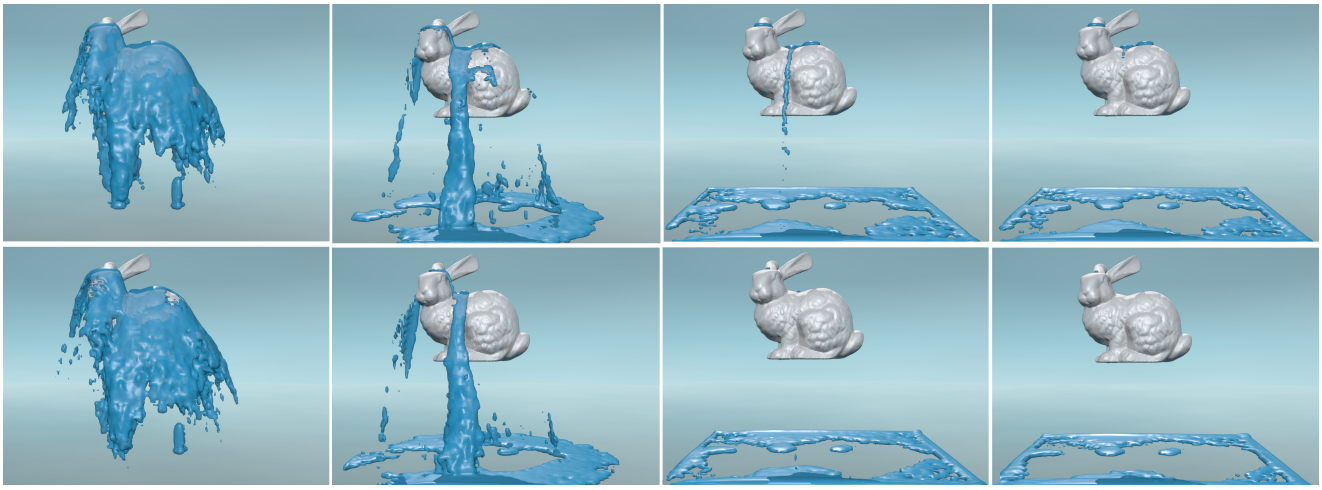


Fig. 18 A fluid pouring on a solid bunny, where we demonstrate a comparison between no-slip and free-slip boundaries. Top row: We imposed a no-slip condition on the boundary, where the simulated fluid tend to stall over a relatively rough surface. Bottom row: A free-slip boundary condition, in which the fluid smoothly flows over the bunny.

discretization can be rectified by using a deformation gradient approach as in [55, 56] or by using predictor-corrector time stepping, as presented in [56].

10 Conclusion

We have presented a novel method for simulating two-way interactions between particle-based fluids and deformable bodies represented by polygonal meshes. The deformation model of deformable solids is based on the Position Based Dynamics (PBD) scheme, which allows us to generate simulations that feature large deformations and handle contact, while remaining stable. Fluid is simulated using divergence-free Incompressible Smoothed Particle Hydrodynamics (ISPH). Incompressible SPH methods with free surfaces suffer from noisy pressures. An incompressible noise-free method for arbitrary free surface flows is essential for many problems which involve both fluid and solid interaction and multi-phase fluid simulations. In this paper, we presented a novel semi-analytic model to simulate free surface flow that does not require explicit identification of surface particles, and allows accurate pressure computation. At the deformable solid and fluid interaction interface, we generate forces and solve for the interaction. Our approach can handle complex coupling scenarios between fluids and thin deformable shells or membranes, where small changes in the state of the fluid cause tremendous changes in the membranes and vice versa. The

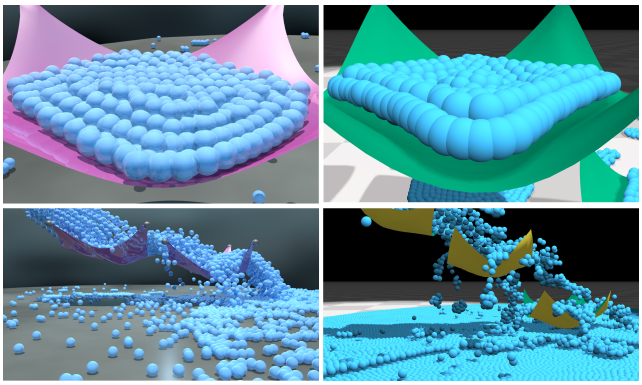


Fig. 19 Visual comparison between our method and the method presented by Macklin et al. [49] for a cloth-fluid coupling scenario with 32K fluid particles. *Left:* Our method employs less contact offset (top) and avoids particle clumping (bottom). *Right:* The method presented by Macklin et al. [49], where they use a visibly larger contact offset when handling contact (top), and the liquid particles tend to unnaturally cluster (bottom).

described method also prevents the leaking of fluid through the thin shell boundaries in the case of large deformations, which is done by enforcing an appropriate set of constraints at the interface between the two materials.

We currently employ a parallel implementation on the CPU. To further improve the performance of our method, developing an efficient GPU implementation is one of our future directions. The recent work by Chow et al. [17] promises much faster Poisson solvers on GPUs than previously achieved in incompressible SPH simulations. Since the pressure equation in incompressible SPH is essentially the Poisson equation discretized as a sparse system of linear equations, the increased interest in solving linear systems makes incompressible SPH very promising and much more attractive than conventional weakly compressible SPH. We believe that in the near future, using the incompressible SPH formulation on a GPU will become increasingly more viable than using conventional SPH. This path certainly offers a rich set of opportunities for future research with applications not only in movies and surgery simulation, but also in computer games.

11 Compliance with Ethical Standards

Funding: This work was partially funded by the IM&M project FOLDDyn (ANR-16-CE33-0015) and by the Deutsche Forschungsgemeinschaft (DFG) through the Cluster of Excellence Engineering of Advanced Materials. We are grateful to the CIMI lab for funding a postdoctoral fellowship to Nadine Abu Rumman.

Conflict of Interest: The authors declare that they have no conflict of interest.

12 Acknowledgement

We would like to thank the reviewers for their insightful comments, and the authors would like to acknowledge the funding of the Deutsche Forschungsgemeinschaft (DFG) through the Cluster of Excellence Engineering of Advanced Materials. The authors wish to express their deepest gratitude to Dr. Bart de Keijzer for the careful proof-reading of this paper.

References

1. Akbay, M., Nobles, N., Zordan, V., Shinar, T.: An extended partitioned method for conservative solid-fluid coupling. ACM Trans. Graph. **37**(4), 86:1–86:12 (2018). DOI 10.1145/3197517.3201345. URL http://doi.acm.org/10.1145/3197517.3201345 3
2. Akinci, N., Cornelis, J., Akinci, G., Teschner, M.: Coupling elastic solids with smoothed particle hydrodynamics fluids. Computer Animation and Virtual Worlds **24**(3-4), 195–203 (2013). DOI 10.1002/cav.1499. URL http://dx.doi.org/10.1002/cav.1499 4, 5, 12
3. Akinci, N., Ihmsen, M., Akinci, G., Solenthaler, B., Teschner, M.: Versatile rigid-fluid coupling for incompressible sph. ACM Trans. Graph. **31**(4), 62:1–62:8 (2012). DOI 10.1145/2185520.2185558. URL http://doi.acm.org/10.1145/2185520.2185558 4
4. Band, S., Gissler, C., Ihmsen, M., Cornelis, J., Peer, A., Teschner, M.: Pressure boundaries for implicit incompressible sph. ACM Trans. Graph. **37**(2), 14:1–14:11 (2018). DOI 10.1145/3180486. URL http://doi.acm.org/10.1145/3180486 13
5. Batty, C., Bertails, F., Bridson, R.: A fast variational framework for accurate solid-fluid coupling. In: ACM SIGGRAPH 2007 Papers, SIGGRAPH '07. ACM, New York, NY, USA (2007). DOI 10.1145/1275808.1276502. URL http://doi.acm.org/10.1145/1275808.1276502 3
6. Batty, C., Houston, B.: A simple finite volume method for adaptive viscous liquids. In: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '11, pp. 111–118. ACM, New York, NY, USA (2011). DOI 10.1145/2019406.2019421. URL http://doi.acm.org/10.1145/2019406.2019421 3
7. Becker, M., Teschner, M.: Weakly compressible sph for free surface flows. In: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '07, pp. 209–217. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2007). URL http://dl.acm.org/citation.cfm?id=1272690.1272719 11
8. Bell, N., Yu, Y., Mucha, P.J.: Particle-based simulation of granular materials. In: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '05, pp. 77–86. ACM, New York, NY, USA (2005). DOI 10.1145/1073368.1073379. URL http://doi.acm.org/10.1145/1073368.1073379 4
9. Bender, J., Koschier, D.: Divergence-free smoothed particle hydrodynamics. In: Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation, SCA '15, pp. 147–155. ACM, New York, NY, USA (2015). DOI 10.1145/2786784.2786796. URL http://doi.acm.org/10.1145/2786784.2786796 4, 13
10. Bodin, K., Lacoursiere, C., Servin, M.: Constraint fluids. IEEE Transactions on Visualization and Computer Graphics **18**(3), 516–526 (2012). DOI 10.1109/TVCG.2011.29. URL http://dx.doi.org/10.1109/TVCG.2011.29 4

11. Bøkmann, A., Shipilova, O., Skeie, G.: Incompressible sph for free surface flows. Computers & Fluids **67**, 138 – 151 (2012). DOI https://doi.org/10.1016/j.compfluid.2012.07.007. URL http://www.sciencedirect.com/science/article/pii/S0045793012002575 2
12. Braley, C., Tech, V., Sandu, A.: Fluid simulation for computer graphics: A tutorial in grid based and particle based methods (2009) 3
13. Bridson, R., Müller-Fischer, M.: Fluid simulation: Siggraph 2007 course notesvideo files associated with this course are available from the citation page. In: ACM SIGGRAPH 2007 Courses, SIGGRAPH '07, pp. 1–81. ACM, New York, NY, USA (2007). DOI 10.1145/1281500.1281681. URL http://doi.acm.org/10.1145/1281500.1281681 3
14. Carlson, M., Mucha, P.J., Turk, G.: Rigid fluid: Animating the interplay between rigid bodies and fluid. ACM Trans. Graph. **23**(3), 377–384 (2004). DOI 10.1145/1015706.1015733. URL http://doi.acm.org/10.1145/1015706.1015733 3
15. Chentanez, N., Goktekin, T.G., Feldman, B.E., O'Brien, J.F.: Simultaneous coupling of fluids and deformable bodies. In: ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 83–89 (2006). URL http://graphics.cs.berkeley.edu/papers/Chentanez-SCP-2006-08/ 3
16. Chentanez, N., Müller, M.: Real-time eulerian water simulation using a restricted tall cell grid. ACM Trans. Graph. **30**(4), 82:1–82:10 (2011). DOI 10.1145/2010324.1964977. URL http://doi.acm.org/10.1145/2010324.1964977 3
17. Chow, A.D., Rogers, B.D., Lind, S.J., Stansby, P.K.: Incompressible sph (isph) with fast poisson solver on a gpu. Computer Physics Communications (2018) 2, 15
18. Clavet, S., Beaudoin, P., Poulin, P.: Particle-based viscoelastic fluid simulation. In: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '05, pp. 219–228. ACM, New York, NY, USA (2005). DOI 10.1145/1073368.1073400. URL http://doi.acm.org/10.1145/1073368.1073400 4
19. Cornelis, J., Bender, J., Gissler, C., Ihmsen, M., Teschner, M.: An optimized source term formulation for incompressible sph. The Visual Computer (2018). DOI 10.1007/s00371-018-1488-8. URL https://doi.org/10.1007/s00371-018-1488-8 4
20. Cummins, S.J., Rudman, M.: An sph projection method. Journal of Computational Physics **152**(2), 584 – 607 (1999). DOI http://dx.doi.org/10.1006/jcph.1999.6246. URL http://www.sciencedirect.com/science/article/pii/S0021999199962460 2, 6, 7
21. Dehnen, W., Aly, H.: Improving convergence in smoothed particle hydrodynamics simulations without pairing instability. Monthly Notices of the Royal Astronomical Society **425**(2), 1068–1082 (2012). DOI 10.1111/j.1365-2966.2012.21439.x. URL http://dx.doi.org/10.1111/j.1365-2966.2012.21439.x 9
22. Desbrun, M., Gascuel, M.P.: Smoothed particles: A new paradigm for animating highly deformable bodies. In: Proceedings of the Eurographics Workshop on Computer Animation and Simulation '96, pp. 61–76. Springer-Verlag New York, Inc., New York, NY, USA (1996). URL http://dl.acm.org/citation.cfm?id=274976.274981 4
23. Färstenau, J.P., Avci, B., Wriggers, P.: A comparative numerical study of pressure-poisson-equation discretization strategies for sph. 12th International SPHERIC Workshop, Ourense, Spain pp. 1–8 (2017) 4
24. Faure, F., Allard, J., Nesme, M.: Eulerian contact for versatile collision processing. Rapport de recherche RR-6203, INRIA (2007) 3
25. Foster, N., Metaxas, D.: Controlling fluid animation. In: Proceedings of the 1997 Conference on Computer Graphics International, CGI '97, pp. 178–. IEEE Computer Society, Washington, DC, USA (1997). URL http://dl.acm.org/citation.cfm?id=792756.792862 3
26. Fournier, A., Reeves, W.T.: A simple model of ocean waves. SIGGRAPH Comput. Graph. **20**(4), 75–84 (1986). DOI 10.1145/15886.15894. URL http://doi.acm.org/10.1145/15886.15894 1
27. Fratarcangeli, M., Pellacini, F.: Scalable partitioning for parallel position based dynamics. Computer Graphics Forum **34**(2), 405–413 (2015). DOI 10.1111/cgf.12570. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12570 5
28. Génevaux, O., Habibi, A., Dischler, J.M.: Simulating fluid-solid interaction. In: Proceedings of the Graphics Interface 2003 Conference, June 11-13, 2003, Halifax, Nova Scotia, Canada, pp. 31–38. CIPS, Canadian Human-Computer Communication Society, Canadian Human-Computer Communications Society and A K Peters Ltd. (2003). URL http://graphicsinterface.org/wp-content/uploads/gi2003-4.pdf 4
29. Gerszewski, D., Bargteil, A.: Physics-based animation of large-scale splashing liquids. ACM Transactions on Graphics **32**(6) (2013). DOI 10.1145/2508363.2508430 3
30. Goktekin, T.G., Bargteil, A.W., O'Brien, J.F.: A method for animating viscoelastic fluids. ACM Trans. Graph. **23**(3), 463–468 (2004). DOI 10.1145/1015706.1015746. URL http://doi.acm.org/10.1145/1015706.1015746 3
31. Guendelman, E., Selle, A., Losasso, F., Fedkiw, R.: Coupling water and smoke to thin deformable and rigid shells. ACM Trans. Graph. **24**(3), 973–981 (2005). DOI 10.1145/1073204.1073299. URL http://doi.acm.org/10.1145/1073204.1073299 3, 8
32. Harada, T., Koshizuka, S., Kawaguchi, Y.: Real-time Fluid Simulation Coupled with Cloth. In: I.S. Lim, D. Duce (eds.) Theory and Practice of Computer Graphics. The Eurographics Association (2007). DOI 10.2312/LocalChapterEvents/TPCG/TPCG07/013-020 4
33. Ihmsen, M., Cornelis, J., Solenthaler, B., Horvath, C., Teschner, M.: Implicit incompressible sph. IEEE Transactions on Visualization and Computer Graphics **20**(3), 426–435 (2014). DOI 10.1109/TVCG.2013.105. URL http://dx.doi.org/10.1109/TVCG.2013.105 4, 9
34. Ihmsen, M., Orthmann, J., Solenthaler, B., Kolb, A., Teschner, M.: SPH Fluids in Computer Graphics. In: S. Lefebvre, M. Spagnuolo (eds.) Eurographics 2014 - State of the Art Reports. The Eurographics Association (2014). DOI 10.2312/egst.20141034 1, 3
35. Irving, G., Guendelman, E., Losasso, F., Fedkiw, R.: Efficient simulation of large bodies of water by coupling two and three dimensional techniques. ACM Trans. Graph. **25**(3), 805–811 (2006). DOI 10.1145/1141911.1141959. URL http://doi.acm.org/10.1145/1141911.1141959 1
36. J, B., M, M., A, O.M., M, T., M, M.: A survey on position-based simulation methods in computer graphics. Computer Graphics Forum **33**(6), 228–251 (2014). DOI 10.1111/cgf.12346 5
37. Jiang, C., Schroeder, C., Teran, J., Stomakhin, A., Selle, A.: The material point method for simulating continuum materials. In: ACM SIGGRAPH 2016 Courses, SIGGRAPH '16, pp. 24:1–24:52. ACM, New York, NY, USA (2016). DOI 10.1145/2897826.2927348. URL http://doi.acm.org/10.1145/2897826.2927348 3
38. Kang, N., Sagong, D.: Incompressible sph using the divergence-free condition. Computer Graphics Forum **33**(7), 219–228 (2014). DOI 10.1111/cgf.12490. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12490 4
39. Koschier, D., Bender, J.: Density maps for improved sph boundary handling. In: Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation, SCA '17, pp. 1:1–1:10. ACM, New York, NY, USA (2017). DOI 10.1145/3099564.3099565. URL http://doi.acm.org/10.1145/3099564.3099565 4

40. Koshizuka, S., Oka, Y.: Moving-particle semi-implicit method for fragmentation of incompressible fluid. Nuclear Science and Engineering **123**(3), 421–434 (1996). DOI 10.13182/NSE96-A24205 4
41. Lenaerts, T., Dutré, P.: Unified sph model for fluid-shell simulations. In: ACM SIGGRAPH 2008 Posters, SIGGRAPH '08, pp. 12:1–12:1. ACM, New York, NY, USA (2008). DOI 10.1145/1400885.1400898. URL http://doi.acm.org/10.1145/1400885.1400898 4
42. Levin, D.I.W., Litven, J., Jones, G.L., Sueda, S., Pai, D.K.: Eulerian solid simulation with contact. ACM Trans. Graph. **30**(4), 36:1–36:10 (2011). DOI 10.1145/2010324.1964931. URL http://doi.acm.org/10.1145/2010324.1964931 3
43. Lind, S., Xu, R., Stansby, P., Rogers, B.: Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. Journal of Computational Physics **231**(4), 1499–1523 (2012). DOI 10.1016/j.jcp.2011.10.027 11
44. Losasso, F., Talton, J., Kwatra, N., Fedkiw, R.: Two-way coupled sph and particle level set fluid simulation. IEEE Transactions on Visualization and Computer Graphics **14**(4), 797–804 (2008). DOI 10.1109/TVCG.2008.37 3
45. Lu, W., Jin, N., Fedkiw, R.: Two-way coupling of fluids to reduced deformable bodies. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '16, pp. 67–76. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2016). URL http://dl.acm.org/citation.cfm?id=2982818.2982829 3
46. Maciá, F., Antuono, M., González, L.M., Colagrossi, A.: Theoretical analysis of the no-slip boundary condition enforcement in sph methods. Progress of Theoretical Physics **125**(6), 1091–1121 (2011). DOI 10.1143/PTP.125.1091. URL http://dx.doi.org/10.1143/PTP.125.1091 9
47. Macklin, M., Müller, M.: Position based fluids. ACM Trans. Graph. **32**(4), 104:1–104:12 (2013). DOI 10.1145/2461912.2461984. URL http://doi.acm.org/10.1145/2461912.2461984 4, 12
48. Macklin, M., Müller, M., Chentanez, N.: Xpbd: Position-based simulation of compliant constrained dynamics. In: Proceedings of the 9th International Conference on Motion in Games, MIG '16, pp. 49–54. ACM, New York, NY, USA (2016). DOI 10.1145/2994258.2994272. URL http://doi.acm.org/10.1145/2994258.2994272 5, 13
49. Macklin, M., Müller, M., Chentanez, N., Kim, T.Y.: Unified particle physics for real-time applications. ACM Trans. Graph. **33**(4), 153:1–153:12 (2014). DOI 10.1145/2601097.2601152. URL http://doi.acm.org/10.1145/2601097.2601152 4, 5, 12, 15
50. Monaghan, J.J.: Smoothed particle hydrodynamics. Annual review of astronomy and astrophysics **30**(1), 543–574 (1992) 6
51. Morris, J.P., Fox, P.J., Zhu, Y.: Modeling low reynolds number incompressible flows using sph. Journal of Computational Physics **136**(1), 214 – 226 (1997). DOI http://dx.doi.org/10.1006/jcph.1997.5776. URL http://www.sciencedirect.com/science/article/pii/S0021999197957764 6
52. Müller, M., Charypar, D., Gross, M.: Particle-based fluid simulation for interactive applications. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '03, pp. 154–159. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2003). URL http://dl.acm.org/citation.cfm?id=846276.846298 4
53. Müller, M., Heidelberger, B., Hennix, M., Ratcliff, J.: Position based dynamics. J. Vis. Comun. Image Represent. **18**(2), 109–118 (2007). DOI 10.1016/j.jvcir.2007.01.005. URL http://dx.doi.org/10.1016/j.jvcir.2007.01.005 2, 5
54. Müller, M., Schirm, S., Teschner, M., Heidelberger, B., Gross, M.: Interaction of fluids with deformable solids. Computer Animation and Virtual Worlds **15**(3-4), 159–171 (2004). DOI 10.1002/cav.18. URL http://dx.doi.org/10.1002/cav.18 4
55. Nair, P., Tomar, G.: An improved free surface modeling for incompressible sph. Computers and Fluids **102**, 304 – 314 (2014). DOI http://dx.doi.org/10.1016/j.compfluid.2014.07.006. URL http://www.sciencedirect.com/science/article/pii/S0045793014002850 14
56. Nair, P., Tomar, G.: Volume conservation issues in incompressible smoothed particle hydrodynamics. J. Comput. Phys. **297**(C), 689–699 (2015). DOI 10.1016/j.jcp.2015.05.042. URL http://dx.doi.org/10.1016/j.jcp.2015.05.042 14
57. Nealen, A., Müller, M., Keiser, R., Boxerman, E., Carlson, M.: Physically based deformable models in computer graphics. Computer Graphics Forum **25**(4), 809–836 (2006). DOI 10.1111/j.1467-8659.2006.01000.x. URL http://dx.doi.org/10.1111/j.1467-8659.2006.01000.x 3
58. Nishida, A.: Experience in Developing an Open Source Scalable Software Infrastructure in Japan, pp. 448–462. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). DOI 10.1007/978-3-642-12165-4_36. URL https://doi.org/10.1007/978-3-642-12165-4_36 9
59. Peer, A., Gissler, C., Band, S., Teschner, M.: An implicit sph formulation for incompressible linearly elastic solids. Computer Graphics Forum **0**(0) (2017). DOI 10.1111/cgf.13317. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13317 4
60. Premzoe, S., Tasdizen, T., Bigler, J., Lefohn, A., Whitaker, R.T.: Particle-Based Simulation of Fluids. Computer Graphics Forum (2003). DOI 10.1111/1467-8659.00687 4
61. Rasmussen, N., Enright, D., Nguyen, D., Marino, S., Sumner, N., Geiger, W., Hoon, S., Fedkiw, R.: Directable photorealistic liquids. In: Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '04, pp. 193–202. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2004). DOI 10.1145/1028523.1028549. URL http://dx.doi.org/10.1145/1028523.1028549 3
62. Robinson-Mosher, A., English, R.E., Fedkiw, R.: Accurate tangential velocities for solid fluid coupling. In: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '09, pp. 227–236. ACM, New York, NY, USA (2009). DOI 10.1145/1599470.1599500. URL http://doi.acm.org/10.1145/1599470.1599500 3
63. Robinson-Mosher, A., Shinar, T., Gretarsson, J., Su, J., Fedkiw, R.: Two-way coupling of fluids to rigid and deformable solids and shells. In: ACM SIGGRAPH 2008 Papers, SIGGRAPH '08, pp. 46:1–46:9. ACM, New York, NY, USA (2008). DOI 10.1145/1399504.1360645. URL http://doi.acm.org/10.1145/1399504.1360645 3
64. Rumman, N.A., Fratarcangeli, M.: State of the art in skinning techniques for articulated deformable characters. In: Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications: Volume 1: GRAPP, GRAPP 2016, pp. 200–212. SCITEPRESS - Science and Technology Publications, Lda, Portugal (2016). DOI 10.5220/0005720101980210. URL https://doi.org/10.5220/0005720101980210 5
65. Rumman, N.A., Schaerf, M., Bechmann, D.: Collision detection for articulated deformable characters. In: Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games, MIG '15, pp. 215–220. ACM, New York, NY, USA (2015). DOI 10.1145/2822013.2822034. URL http://doi.acm.org/10.1145/2822013.2822034 5
66. Schechter, H., Bridson, R.: Ghost sph for animating water. ACM Trans. Graph. **31**(4), 61:1–61:8 (2012). DOI 10.1145/2185520.2185557. URL http://doi.acm.org/10.1145/2185520.2185557 2
67. Si, H.: Tetgen, a delaunay-based quality tetrahedral mesh generator. ACM Trans. Math. Softw. **41**(2), 11:1–11:36 (2015). DOI 10.1145/2629697. URL http://doi.acm.org/10.1145/2629697 5

68. Solenthaler, B., Pajarola, R.: Predictive-corrective incompressible sph. ACM Trans. Graph. **28**(3), 40:1–40:6 (2009). DOI 10.1145/1531326.1531346. URL http://doi.acm.org/10.1145/1531326.1531346 4
69. Stam, J.: Stable fluids. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '99, pp. 121–128. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (1999). DOI 10.1145/311535.311548. URL http://dx.doi.org/10.1145/311535.311548 1
70. Teng, Y., Levin, D.I.W., Kim, T.: Eulerian solid-fluid coupling. ACM Trans. Graph. **35**(6), 200:1–200:8 (2016). DOI 10.1145/2980179.2980229. URL http://doi.acm.org/10.1145/2980179.2980229 3
71. Teschner, M., Heidelberger, B., Müller, M., Pomerantes, D., Gross, M.H.: Optimized spatial hashing for collision detection of deformable objects. In: VMV (2003) 8
72. Winchenbach, R., Hochstetter, H., Kolb, A.: Infinite continuous adaptivity for incompressible sph. ACM Trans. Graph. **36**(4), 102:1–102:10 (2017). DOI 10.1145/3072959.3073713. URL http://doi.acm.org/10.1145/3072959.3073713 13
73. Zarifi, O., Batty, C.: A positive-definite cut-cell method for strong two-way coupling between fluids and deformable bodies. In: Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation, SCA '17, pp. 7:1–7:11. ACM, New York, NY, USA (2017). DOI 10.1145/3099564.3099572. URL http://doi.acm.org/10.1145/3099564.3099572 3