

An Optimized Source Term Formulation For Incompressible SPH

Jens Cornelis · Jan Bender · Christoph Gissler · Markus Ihmsen · Matthias Teschner

Abstract Incompressible SPH (ISPH) is a promising concept for the pressure computation in SPH. It works with large timesteps and the underlying pressure Poisson equation (PPE) can be solved very efficiently. Still, various aspects of current ISPH formulations can be optimized.

This paper discusses issues of the two standard source terms that are typically employed in PPEs, i.e. density invariance (DI) and velocity divergence (VD). We show that the DI source term suffers from significant artificial viscosity, while the VD source term suffers from particle disorder and volume loss.

As a conclusion of these findings, we propose a novel source term handling. A first PPE is solved with the VD source term to compute a divergence-free velocity field with minimized artificial viscosity. To address the resulting volume error and particle disorder, a second PPE is solved to improve the sampling quality. The result of the second PPE is used for a particle shift (PS) only. The divergence-free velocity field - computed from the first PPE - is not changed, but only resampled at the updated particle positions. Thus, the proposed source term handling incorporates velocity divergence and particle shift (VD+PS).

The proposed VD+PS variant does not only improve the quality of the computed velocity field, but also accelerates the performance of the ISPH pressure computation. This is illustrated for IISPH - a recent ISPH implementation - where a performance gain factor of 1.6 could be achieved.

Keywords Three-Dimensional Graphics and Realism · Animation

1 Introduction

Incompressibility for Lagrangian fluids can be enforced in various ways. One popular way is to employ an equation of state (EOS) that relates density deviations to pressure values, e.g. [1, 4, 26, 27, 29, 42]. Such EOS solvers are simple to implement, but choosing and parameterizing an appropriate form of the state equation can be cumbersome. This issue has been addressed by iterative EOS solvers where pressure is iteratively refined, e.g. [16, 36]. Such solvers employ an EOS with an analytically motivated parameterization. In contrast to standard EOS solvers, iterative EOS solvers are conveniently parameterized by a density deviation that should be enforced and can be extended in various ways, e.g. for versatile interface handling [43, 44]. Further, incompressibility formulations for Lagrangian fluids have been presented in [6, 7], where the typical EOS is replaced by a constraint equation. The resulting formulations either compute distances [6] that can be related to pressure forces or Lagrange multipliers [7] that can be interpreted as pressure values.

Alternatively, pressure can also be computed by solving a PPE of the form $\nabla^2 p = s$ with s being a source term, e.g. [10, 11, 13, 21, 22, 31–33, 37]. SPH fluid solvers that employ such a PPE are typically referred to as

J. Cornelis
FIFTY2 Technology GmbH

J. Bender
RWTH Aachen University, Germany

C. Gissler
University of Freiburg, Germany

M. Ihmsen
FIFTY2 Technology GmbH

M. Teschner
University of Freiburg, Germany

ISPH. ISPH approaches compute the pressure by solving a linear system. This results in a rather smooth pressure field that positively affects the stability, i.e. the size of the time step. ISPH has also proven to ensure incompressibility for adaptive simulations with continuous particles sizes [40]. As shown, e.g. in [21], PPE solvers can be implemented in a memory-efficient matrix-free form. PPE solvers can be realized in various ways. On the one hand, different discretizations of $\nabla^2 p$ can be used. On the other hand, the source term s can be computed in different ways.

This paper focuses on the optimization of the source term in ISPH. Therefore, we use one specific discretization of $\nabla^2 p$ and one specific solver for the respective linear system as proposed in [21]. This choice is motivated in Sec. 2. Based on this setup, we investigate three variants of the source term s . This includes the two standard options, i.e. the divergence of the velocity that is predicted from non-pressure accelerations (VD), e.g. [11], and the corresponding predicted density deviation (DI), e.g. [24,34]. We further propose a third variant with combined velocity divergence and particle shift (VD+PS) that results in an improved quality of the computed velocity field and also in an accelerated computation of the pressure field compared to the standard source terms.

Our analyses show that the standard source terms suffer from specific issues. The DI source term introduces a significant amount of artificial viscosity, while the VD source term suffers from particle disorder and volume loss. On the other hand, the DI source term leads to a high-quality sampling with guaranteed minimal density deviation and the VD source term results in a divergence-free velocity field. These positive properties motivate a combination of both variants that can, e.g., be realized by solving two PPEs with different source terms. Similar combinations have already been proposed for SPH and position-based fluids, e.g. [5,18,23]. These combination, however, typically result in inconsistent particle positions and velocities, i.e. the particles are not advected with their velocity. Our proposed combination resolves this issue. A first PPE is solved with the VD source term to compute a divergence-free velocity field with minimal artificial viscosity. The particles are advected with this velocity field, leading to consistent positions and velocities, a prerequisite for a Lagrangian framework. As mentioned above, this approach accumulates local volume deviations that manifest itself in particle disorder and a global volume error, typically a loss of volume. To resolve this issue, a second PPE is solved with the DI source term to compute particle shifts, i.e. small displacements. These particle shifts resolve particle disorder and volume loss,

but they do not change the divergence-free velocity field that has been computed from solving the first PPE. As the corrected final particle positions would be inconsistent with the velocities, the velocity field is resampled at the final particle positions.

The described combination of two PPEs is novel, but closely related to local particle shifting, e.g. [30,35,41]. Similar to our variant, a PPE is solved for a divergence-free velocity field. The position adjustment, however, is computed locally by shifting sample positions from high to low concentrations. A similar strategy is also used in standard constraint approaches that first remove illegal accelerations with a global Lagrange multiplier approach, followed by a local Baumgarte stabilization to address the drift problem, e.g. [9]. In contrast to these approaches, our method computes the adjustment of the sample positions in a global way using a PPE with DI source term.

Our paper focuses on the investigation of the properties of the three described source term variants in ISPH. We illustrate artificial viscosity, i.e. diffusion, in the velocity field when working with the DI term and we analyze the volume drift when using the VD term. We show how the proposed combination VD+PS resolves both issues which results in an improved quality of the computed velocity field. We further show that the proposed source term handling reduces the computation time of the pressure solver compared to solving the PPE with one of the standard forms, DI and VD.

The remainder of the paper is organized as follows. Sec. 2 shows a derivation of a PPE with a VD source term. The ISPH discretization that is employed for the Laplacian in the PPE and the ISPH solver for the PPE are motivated. Sec. 3 derives the three considered variants of the source term handling: VD, DI, and VD+PS. Sec. 4 describes implementation details, while Sec. 6 discusses additional properties and aspects of the different source terms and the proposed handling. Sec. 5 illustrates the improved quality of the velocity field when using our approach. Experiments also show the improved performance of VD+PS compared to VD and DI.

2 Pressure Poisson Equation

In order to derive the PPE, non-pressure and pressure accelerations are separated in the incompressible Navier-Stokes momentum equation

$$\frac{D\mathbf{v}(t)}{Dt} = -\frac{1}{\rho^0} \nabla p(t) + \mathbf{a}^{\text{nonp}}(t) \quad (1)$$

with time t , velocity \mathbf{v} , rest density ρ^0 , pressure p and non-pressure accelerations \mathbf{a}^{nonp} , in particular gravity

and acceleration due to viscosity [8]. An intermediate velocity \mathbf{v}^* is predicted from non-pressure accelerations with

$$\mathbf{v}^* = \mathbf{v}(t) + \Delta t \mathbf{a}^{\text{nonp}}(t) \quad (2)$$

and the final velocity $\mathbf{v}(t + \Delta t)$ is computed with

$$\mathbf{v}(t + \Delta t) = \mathbf{v}^* - \Delta t \frac{1}{\rho^0} \nabla p(t). \quad (3)$$

In order to compute the pressure $p(t)$, Eq. 3 is rewritten as

$$\frac{\mathbf{v}(t + \Delta t) - \mathbf{v}^*}{\Delta t} = -\frac{1}{\rho^0} \nabla p(t). \quad (4)$$

Taking the divergence of both sides, we get

$$\nabla \cdot \left(\frac{\mathbf{v}(t + \Delta t) - \mathbf{v}^*}{\Delta t} \right) = -\nabla \cdot \left(\frac{1}{\rho^0} \nabla p(t) \right). \quad (5)$$

Using the fact that the velocity field $\mathbf{v}(t + \Delta t)$ should be divergence-free, i.e. $\nabla \cdot \frac{\mathbf{v}(t + \Delta t)}{\Delta t} = 0$, we get a first variant of the PPE

$$\Delta t \nabla^2 p(t) = \rho^0 \nabla \cdot \mathbf{v}^* \quad (6)$$

with $\rho^0 \nabla \cdot \mathbf{v}^*$ being the VD source term that represents the divergence of the intermediate velocity field that is predicted from non-pressure accelerations $\mathbf{a}^{\text{nonp}}(t)$ according to Eq. 2.

As mentioned in Sec. 1, we focus our analyses on different forms of the source term and therefore employ implicit incompressible SPH (IISPH) [21] as one specific discretization of $\Delta t \nabla^2 p(t)$. This choice is motivated by several reasons. First of all, IISPH addresses the important issue of operator inconsistency in SPH. As the SPH discretization of $\nabla^2 p$ is generally not equal to the SPH discretization of $\nabla \cdot \nabla p$, i.e. $\langle \nabla^2 p \rangle \neq \langle \nabla \cdot \nabla p \rangle$, IISPH works with $\langle \nabla \cdot \nabla p \rangle$ rather than $\langle \nabla^2 p \rangle$ with $\langle x \rangle$ denoting the discretized form of a term x . This is beneficial to the convergence of the solver as the same discretization $\langle \nabla p \rangle$ is consistently used in the computation of the pressure force. A second motivation is the performance and memory consumption of IISPH in combination with a simple implementation. The experiments in [21] indicate that IISPH is faster than iterative EOS solvers for complex scenarios, while the relaxed Jacobi technique for solving the PPE can be implemented in a matrix-free way that just requires the storage of seven additional scalar values per particle. Again, our paper does not discuss different discretizations of the Laplacian or more efficient solvers of the discretized PPE. Instead, we focus on optimizing the source term in order to improve the performance of the pressure computation and also the quality of the computed velocity field.

3 Source Terms

VD: This variant of the source term has already been derived in Sec. 2 leading to a PPE of the following form:

$$\Delta t \nabla^2 p(t) = \rho^0 \nabla \cdot \mathbf{v}^*. \quad (7)$$

DI: The density invariance form of the source term can, e.g., be derived from the mass conservation law:

$$\frac{D\rho(t + \Delta t)}{Dt} + \rho(t + \Delta t) \nabla \cdot \mathbf{v}(t + \Delta t) = 0. \quad (8)$$

Using a backward difference for $\frac{D\rho(t + \Delta t)}{Dt}$, Eq. 3 for $\mathbf{v}(t + \Delta t)$ and the constraint that the density at the next timestep should be equal to the rest density, i.e. $\rho^0 = \rho(t + \Delta t)$, we get

$$\frac{\rho^0 - \rho(t)}{\Delta t} + \rho^0 \nabla \cdot \left(\mathbf{v}^* - \Delta t \frac{1}{\rho^0} \nabla p(t) \right) = 0 \quad (9)$$

which can be written as

$$\frac{\rho^0 - (\rho(t) - \Delta t \rho^0 \nabla \cdot \mathbf{v}^*)}{\Delta t} - \Delta t \nabla^2 p(t) = 0. \quad (10)$$

The term $\rho(t) - \Delta t \rho^0 \nabla \cdot \mathbf{v}^*$ is an approximation of the density at particle positions after advection with \mathbf{v}^* . This term is typically denoted as ρ^* , resulting in the following PPE:

$$\Delta t \nabla^2 p(t) = \frac{\rho^0 - \rho^*}{\Delta t}. \quad (11)$$

VD+PS: As already outlined in Sec. 1, our proposed concept first solves for a divergence-free velocity field using Eq. 7, followed by solving Eq. 11 to compute particle shifts that result in an optimized particle sampling. The predicted velocity \mathbf{v}^* is computed with Eq. 2 and a pressure p^* is computed with

$$\Delta t \nabla^2 p^* = \rho^0 \nabla \cdot \mathbf{v}^*. \quad (12)$$

The pressure p^* is used to compute a velocity change that results in a divergence-free velocity field

$$\mathbf{v}'(t + \Delta t) = \mathbf{v}^* - \Delta t \frac{1}{\rho^0} \nabla p^*. \quad (13)$$

Consistent particle positions are computed as

$$\mathbf{x}^{**} = \mathbf{x}(t) + \Delta t \mathbf{v}'(t + \Delta t). \quad (14)$$

The notation indicates that we have our final velocity field $\mathbf{v}'(t + \Delta t)$, but sampled at intermediate positions \mathbf{x}^{**} . For the computation of the final positions $\mathbf{x}(t + \Delta t)$, we estimate the density ρ^{**} for the sampling \mathbf{x}^{**} with $\rho^{**} = \rho(t) - \Delta t \cdot \rho^0 \cdot \nabla \cdot \mathbf{v}'$ and solve a second PPE of the form

$$\Delta t \nabla^2 p^{**} = \frac{\rho^0 - \rho^{**}}{\Delta t}. \quad (15)$$

The pressure p^{**} is used to compute a particle shift that results in an incompressible state of the sample positions with minimized particle disorder:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}^{**} - \Delta t^2 \frac{1}{\rho^0} \nabla p^{**}. \quad (16)$$

Eqs. 15 and 16 are a global, parameter-free variant of the particle shifting approach, e.g. [30,35,41]. Its goal is to resample the final velocity field $\mathbf{v}'(t + \Delta t)$ at intermediate positions \mathbf{x}^{**} to final velocities $\mathbf{v}(t + \Delta t)$ at particle positions $\mathbf{x}(t + \Delta t)$ with improved sampling. We resample the velocities $\mathbf{v}(t + \Delta t)$ at positions $\mathbf{x}(t + \Delta t)$ with

$$\mathbf{v}(t + \Delta t) = \mathbf{v}'(t + \Delta t) + \nabla \mathbf{v}'(t + \Delta t) \cdot (\mathbf{x}(t + \Delta t) - \mathbf{x}^{**}). \quad (17)$$

Please note again that $\mathbf{v}(t + \Delta t)$ is just a resampled version of $\mathbf{v}'(t + \Delta t)$. They both represent the same divergence-free velocity field. The solution of the second PPE with DI source term is only used to improve the sample positions. The subsequent resampling of the velocity field restores the consistency of positions and velocities.

Eq. 17 requires the computation of the velocity gradient $\nabla \mathbf{v}'(t + \Delta t)$ at the intermediate positions \mathbf{x}^{**} , i.e.

$$\nabla \mathbf{v}'_i(t + \Delta t) = \frac{1}{\rho_i} \sum_j m_j (\mathbf{v}'_j(t + \Delta t) - \mathbf{v}'_i(t + \Delta t)) \otimes \nabla W_{ij}^{**}. \quad (18)$$

As this formulation would require a second neighbor search per simulation step, we use the sampling $\mathbf{x}(t)$ instead and compute

$$\nabla \mathbf{v}'_i(t + \Delta t) = \frac{1}{\rho_i} \sum_j m_j (\mathbf{v}'_j(t + \Delta t) - \mathbf{v}'_i(t + \Delta t)) \otimes \nabla W_{ij}. \quad (19)$$

Using the current neighborhood to compute ∇W_{ij} in Eq. 19 is an approximation. Not using the correct neighborhood at intermediate positions \mathbf{x}^{**} to compute ∇W_{ij}^{**} as indicated in Eq. 18 is simply motivated by performance aspects. Both neighborhoods are supposed to be similar, while working with ∇W_{ij} saves one additional neighbor search per simulation step.

Approximate neighbor setting are commonly used in various SPH solvers. E.g., IISPH [21] uses the current neighborhood instead of the neighborhood at the next timestep (see Eq. 3 in [21]). Further, PCISPH [36] explicitly discusses approximations in the considered neighborhood, where position changes during the solver iterations are considered, but in- and out-moving samples are ignored. Non-SPH particle-based solvers also

employ similar approximations. E.g., PBF [25] bases on a constraint that is considered at the subsequent simulation step, while the neighborhood for its computation is taken from the current simulation step. The same approximation is introduced in [7].

4 Implementation

We use SPH for interpolations of a physical quantity A as $A_i = \sum_j \frac{m_j}{\rho_j} A_j W_{ij}$ where m is the mass and ρ is the density [26]. W_{ij} is a Gaussian-like kernel function. We use a cubic spline kernel. The sum considers all particles j within the smoothing length of the particle i . The rest distance of fluid particles is h whereas the support of the kernel function we employ is $2h$. For the computation of spatial derivatives we use the approximations $\nabla A_i = \rho_i \sum_j m_j \left(\frac{A_i}{\rho_i^2} + \frac{A_j}{\rho_j^2} \right) \nabla W_{ij}$ for the gradient and $\nabla \cdot \mathbf{A}_i = \frac{1}{\rho_i} \sum_j m_j (\mathbf{A}_j - \mathbf{A}_i) \nabla W_{ij}$ for the divergence.

The resulting algorithm is outlined in Algorithm 1. The implementation of the particle neighbor search is performed as described in [19], but could also be done with alternative concepts, e.g. [39]. Boundary handling is implemented for one-layer boundaries with varying sampling density, e.g. [3,20]. We employ a viscous force modeled as in [28]. In free surface scenarios, we employ surface tension as in [38]. Again, alternative formulations, e.g. [2,4], would work as well.

The pressure value of a particle in the DI solver is initialized as $p_i(t + \Delta t) = 0.5p_i(t)$. The stopping criterion of the DI solver and the PS solve of the VD+PS solver is defined as an average density error of 0.1%. The residual of the VD solver and the VD solve of the VD+PS solver is set to 0.01. All solvers are implemented as a Relaxed Jacobi method with $\omega = 0.5$.

Algorithm 1 VD+PS algorithm

procedure COMPUTE VELOCITY \mathbf{v}^*

 compute \mathbf{v}^*

 compute p^*

procedure SOLVE VELOCITY DIVERGENCE

 compute $\mathbf{v}'(t + \Delta t)$

 compute \mathbf{x}^{**}

 compute ρ^{**}

procedure SOLVE POSITION SHIFT

 compute p^{**}

 compute $\mathbf{x}(t + \Delta t)$

procedure RESAMPLE VELOCITIES

 compute $\mathbf{v}(t + \Delta t)$

5 Results

We have performed various experiments that illustrate the benefits of the proposed VD+PS method compared to the standard source term forms VD and DI.

5.1 Shear Wave Decay

We first discuss the so-called shear wave decay, e.g. [14, 17]. Please note, that this setup is named after the initial sinusoidal velocity wave that is spatially applied and that the term wave is not indicating oscillation. The shear wave decay is a 2D scenario that is particularly well suited to illustrate the numerical viscosity of the DI term and also to show the particle disorder that occurs with the VD term. The fluid bulk is simulated in a 2D square area with periodic boundaries, such that boundary handling does not influence the solution of the pressure computation. No other forces, in particular gravity or explicit viscosity, are acting on the fluid. An initial velocity is applied to the fluid particles as

$$\mathbf{v}_i(y, t = 0) = \begin{pmatrix} v_0 \sin\left(\frac{2\pi}{L}y\right) \\ 0 \end{pmatrix}, \quad (20)$$

where y is the position of the particle in y direction, v_0 is the amplitude of the sinus function and L is the side length of the square simulation domain.

Due the absence of other than pressure forces, the velocity field should not change over time. A potential decay of the initial velocity is a metric for artificial viscosity introduced by the pressure solver. The velocity profile is measured in a sensor line positioned in the median of the square domain. The setup of the sinusoidal velocity profile and the sensor line is illustrated in Figure 1.

We have performed this experiment for all three source term variants, namely velocity divergence (VD), density invariant (DI) and the proposed variant which first solves for the velocity divergence and imposes a correction of the particle sampling by restoring the density invariance afterwards (VD+PS - divergence free particle shift).

The comparison of the velocity decay at time $t = 10s$ of the velocity profile as shown in Figure 2 indicates that DI results in a noisy and locally divergent velocity field with significant numerical diffusion. In contrast, VD and the proposed VD+PS method almost maintain the initial velocity profile at this time which indicates less numerical diffusion.

However, comparing the decay of the amplitude over time as shown in Figure 3, VD+PS clearly outperforms both DI and VD in terms of preserving the initial velocity. After 30 physical seconds, the amplitude in the

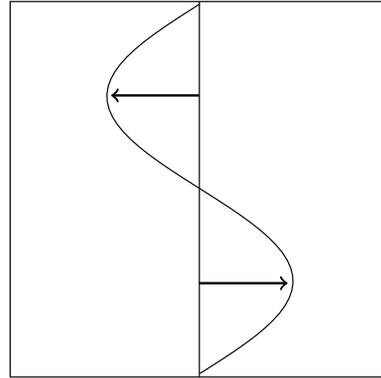


Fig. 1 Setup of the 2D shear wave scene.

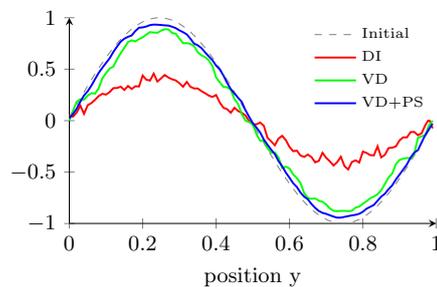


Fig. 2 Velocity profile measured in the median of the square domain at time $t = 10s$.

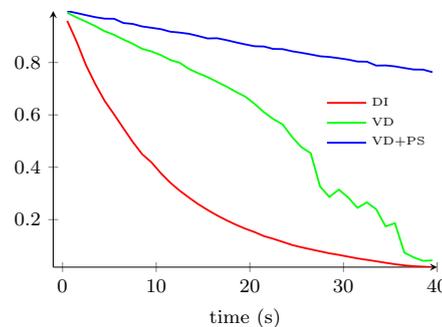


Fig. 3 Shear wave. Maximum velocity amplitudes. DI suffers from large numerical viscosity. VD suffers from less numerical viscosity. The particle disorder in VD, however, leads to additional errors in the velocity field. The proposed VD+PS variant minimizes the numerical viscosity.

VD simulation suddenly decreases and the curvature gets unsteady due to the increasing irregularity of the particle distribution.

This issue is emphasized by the maximum density as illustrated in Figure 4. While DI and VD+PS preserve the maximum density, the density using VD increases due to the local clustering of particles.

While numerical diffusion in DI is introduced by the divergent velocity field, VD is suffering from the increasingly irregular sampling of the simulation domain.

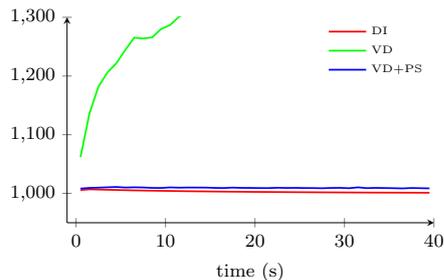


Fig. 4 Shear wave max densities with rest density $\rho_0 = 1000$. The particle disorder in the VD variant results in large local density errors as indicated by the maximum density. VD and VD+PS do not suffer from particle disorder.

VD+PS addresses both limitations and results in a divergence free velocity field while maintaining a homogeneous particle distribution. Like this, VD+PS outperforms the compared source term variants by introducing the least numerical diffusion while preserving the fluid volume and a robust spatial discretization. The visual comparison of the different variants is shown in Figure 5.

5.2 Breaking Dam 2D

We demonstrate the properties of the different source term variants with a 2D simulation of a breaking dam. The initial volume of fluid is 58.8 litres. The simulation has been performed for a physical time of 5s. For an increased visual comparability, the fluid surface is meshed using Preon Mesher [12]. The simulation outcome is very similar for the initial phase of the simulation, as illustrated in the comparison in Figure 6.

However, the results start to significantly differ after only short physical time. As depicted in Figure 7, after 3.56s the surface of the fluid computed with DI is very unsteady and shows many splashes, that do not occur in the respective VD and VD+PS simulations. The reason for this behavior is the local divergence in the velocity field that becomes obvious especially at the surface of the fluid, when incoming droplets introduce sudden velocity changes that the DI solver is not directly resolving. The VD solver results in a very smooth velocity field but is suffering from the increasing compression of the particles that is not sufficiently representing the fluid volume and leads to severe volume loss. Due to the volume loss, the VD solver is not able to represent the actual fluid behavior after only a few physical seconds.

At the end of the simulation at $t = 5$ s, as shown in Figure 8, the average density of the fluid simulated with the DI solver is $1800 \frac{kg}{m^3}$ which corresponds to a volume loss of 45%. The behavior of fluid on a macroscopic scale of both, the DI and the VD+PS solver, is

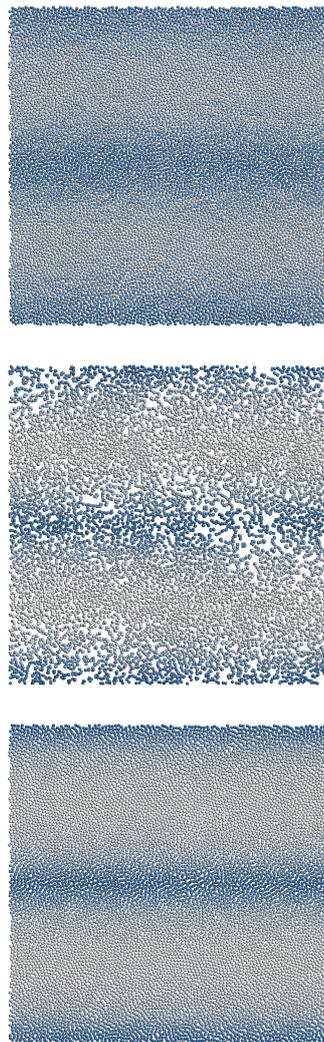


Fig. 5 Shear Wave Decay after 6 seconds simulated with (from top to bottom) DI, VD and VD+PS. The particle velocity is color-coded. The DI variant suffers from larger numerical viscosity compared to VD+PS. The VD variant suffers from particle disorder which does not occur with our proposed VD+PS variant.

very similar. Both solvers are able to preserve the initial fluid volume. However, DI suffers from artifacts that are introduced by the locally divergent velocity field, while VD+PS is not prone to such anomaly. Additionally, the performance of VD+PS is superior to the DI variant. This is analyzed in detail Section 5.3 and discussed in Section 6.

5.3 Breaking Dam 3D

The performance is compared using a corner breaking dam in a $3 \times 3 \times 3$ m box with $3m^3$ water initialized in one corner of the respective box. The setup is illus-

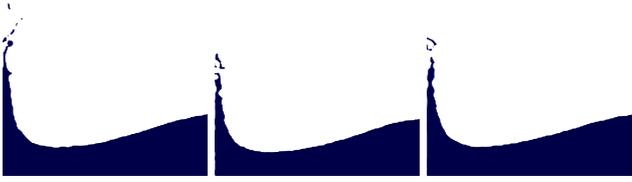


Fig. 6 2D breaking dam at $t=0.74s$. From left to right: DI, VD, VD+PS. The results of all variants are very similar.



Fig. 7 2D breaking dam at $t=3.56s$. From left to right: DI, VD, VD+PS. DI is suffering from local divergence that becomes obvious in splashes and an unsteady behaviour at the free surface. For VD the increasing volume loss gets obvious. VD+PS both conserves volume and keeps a smooth velocity field preserving the vorticity.



Fig. 8 2D breaking dam at $t=5.0s$. From left to right: DI, VD, VD+PS. DI still suffers from the unsteady surface coming from the divergent velocity field, while VD has lost 45% of its original volume and cannot resemble the correct velocity anymore, due to the irregularity of the particle distribution. VD+PS still serves stable velocity conditions and preserved fluid volume.

trated in Figure 9. We compare the DI implementation using the density invariant source term as described in [21] with the proposed VD+PS variant in terms of convergence and performance using the maximum density and divergence as residual. We do not consider the VD variant in the following comparisons and analyses since, while being the fastest approach, the result severely suffers from volume loss up to 50% after only 3 physical seconds.

The performance of the solver is mainly affected by the number of iterations the solver takes. Please note, that in terms of VD+PS, the number of iterations is consisting of the iterations of both, the velocity divergence source term solve and the density deviation source term solve as shown in Figure 10.

For the comparison with DI, the total number of iterations and the total computation time for solving the system is taken into account. Figure 11 shows, that VD+PS takes less iterations than DI during the complete simulation. This results in less computational ef-

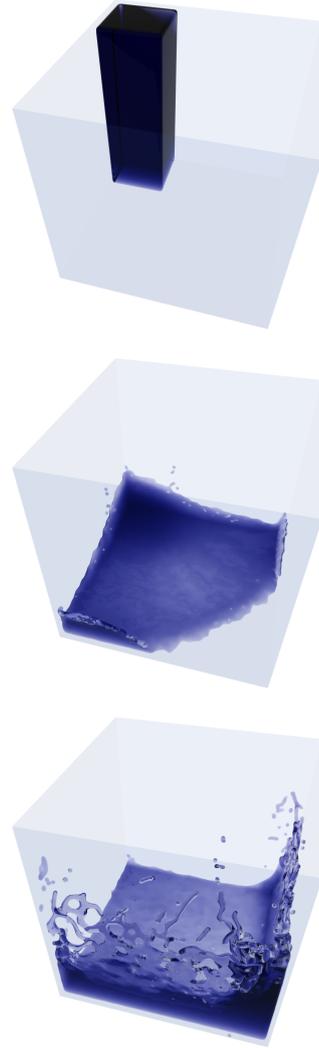


Fig. 9 Breaking dam simulated with VD+PS.

fort per simulation step, as shown in Figure 12. This results in faster computation of VD+PS compared to DI by the factor of 1.6 while suffering less from numerical diffusion leading to higher vorticity.

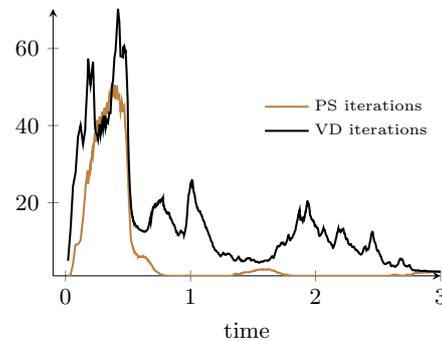


Fig. 10 VD+PS iterations per PPE solve for the breaking dam scenario in Fig. 9.

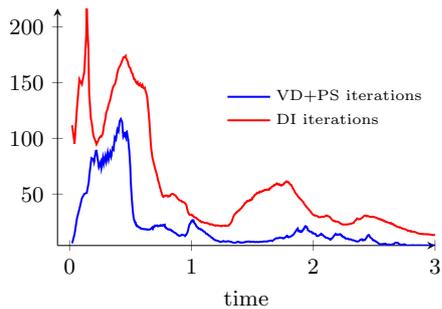


Fig. 11 Number of solver iterations for the breaking dam scenario in Fig. 9.

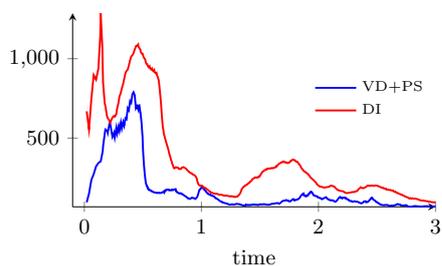


Fig. 12 Pressure computation time in ms for the breaking dam scenario in Fig. 9.

5.4 Production Example

We have used VD+PS to simulate a waterfall consisting of 1.1 million fluid particles as shown in Figure 13. The simulation lasts 20 physical seconds and was computed on a Dual-CPU XEON E5 computer with 32 cores in 36 hours. The rendering was performed using Preon Renderer [12].

6 Discussion

Contribution: Solving PPEs with DI our VD source terms is well-accepted. Also, the particle shift concept is well-established. Now, we propose to interpret the solution of a PPE with DI source term as a variant to realize particle shift in a global way. And, we propose to combine a VD PPE solver with the proposed global particle shift. So on one hand, we use and combine well-established, well-accepted concepts in our proposed VD+PS solver. On the other hand, however, the proposed scheme is novel and unique. We also provide an analysis that illustrates issues of the DI and VD source terms, namely significant artificial viscosity and volume loss, respectively. We further show that the proposed VD+PS scheme improves the quality of the

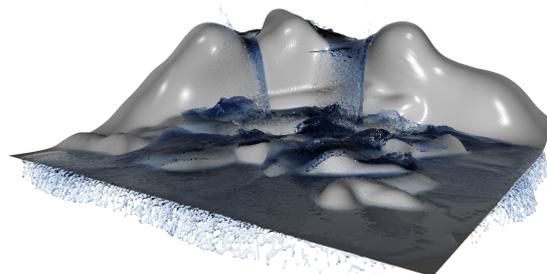
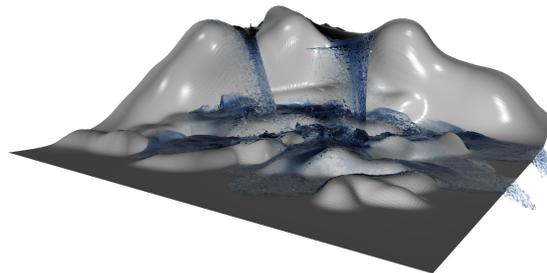
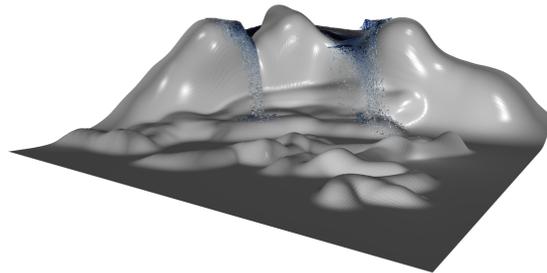


Fig. 13 Waterfall simulated with VD+PS.

computed velocity field which manifests in an improved visual quality and also in an improved performance of the PPE solver.

Relation of VD+PS to existing PPE solvers with DI source term: PPE solvers with DI source term apply the computed pressure accelerations to update the sample velocity which is used to advect the samples. We move the samples in exactly the same way, but we do not update the velocities using the solution of the PPE solver with the DI source term. Instead, we keep the previously computed velocity field from the PPE solver with VD source term unchanged. So, our goal is to preserve the velocity field from the VD step due to minimal artificial viscosity, but we would like to sample this field at positions that we have computed from the PPE solver with DI source term due to optimal sampling quality. Although we update the sample velocities after the position update from the DI step, this veloc-

ity update does not constitute a change in the velocity field. Instead, it is just a resampling of the VD velocity field which is in contrast to standard DI PPE solvers.

Relation of VD+PS to existing particle shift PS approaches: Our PS differs to all existing PS variants. Existing variants compute local so-called concentration gradients and apply a position shift from high concentration to low concentration to improve the particle sampling. We use the same concept, but we compute optimal position shifts in a global way by solving a PPE with DI source term. At first, it sounds expensive, but the solver requires very few iterations. On the other hand, our global approach does not require a user-defined parameter to scale the particle shift that is used in all local approaches. If this user-defined parameter is too small, the resulting sampling quality is not as good as it could be. On the other hand, if the parameter is too large, over-correction occurs which can result in even worse sampling qualities after the particle shift. So, in contrast to all existing PS variants, we compute a global, parameter-free, stable and optimal solution to the PS problem.

Artificial viscosity due to the resampling of the velocity field: The resampling of the velocity field in the second solver step is a regular SPH interpolation which introduces artificial viscosity. So, why don't we see this artificial viscosity, e.g. in Fig. 2? First of all, the viscosity levels of VD and VD+PS are very similar as both variants constitute the same velocity field that is sampled at different sample sets. Although the PPE solver with DI source term introduces a comparatively large amount of artificial viscosity, this viscosity is not transferred to the VD+PS solver. The VD+PS solver only uses the final positions computed by the DI PPE solver, but not the respective velocities due to the large amount of introduced artificial viscosity. Instead, the original DF velocity field with minimal artificial viscosity is resampled at the positions we got from the DI PPE solver step. Still, this resampling introduces artificial viscosity which should be visible in the graph in Fig. 2. We speculate that this is not the case for the two following reasons. First, the artificial viscosity introduced by resampling the velocity field with an interpolation is much smaller compared to the artificial viscosity that is introduced to the velocity field as a result of the PPE solver with DI source term. Second, it even looks like VD+PS has slightly less numerical viscosity than VD. This can be attributed to the different sampling qualities in VD and VD+PS. The sampling quality in VD degrades with time, while the sampling quality in VD+PS is rather constant over time. This can result in the fact that the small amount of arti-

cial viscosity introduced by the PS step is finally not perceivable due to the improved sampling in VD+PS.

Performance of the VD+PS solver: It is not possible to show that VD+PS is always faster than other VD PPE solvers or DI PPE solvers. This is simply due to the fact that there exist simple scenarios where VD or DI solvers are definitely faster than our combined VD+PS solver. If one has, e.g., one layer of fluid particles, VD and DI solvers require the minimum number of solver iterations which can obviously not be reduced by solving two PPEs instead. The main motivation for the proposed combined solver is actually the combination of positive aspects of both solver types, i.e. combining minimal artificial viscosity from the VD solver with optimal sampling from the DI solver. The experienced performance gain is a remarkable side effect for complex scenarios.

We can speculate why the combined VD+PS reduces the overall iteration count in complex scenarios. PPE solvers with DI source term compute a perfect particle sampling, but the divergence of the final velocity field is not necessarily zero. This might be due to the fact that small density deviations are corrected by the DI solver which requires the introduction of divergence in the computed velocity change due to pressure. While this divergence is perfect for the current simulation step, it has unknown effects for the subsequent simulation step. The subsequent step assumes that it starts with a divergence free velocity field which is not the case. So, the DI solver has to handle two error sources. The sampling quality of predicted positions, i.e. the predicted density, has to be preserved, but previously introduced velocity divergences have to be corrected as well. This can lead to increased solver iterations in the DI case. This issue is addressed by the VD+PS scheme. At the end of the respective simulation step, we do not only have a perfect sampling, but we also have a divergence free velocity field which makes it easier and faster for the solver to find the solution of the subsequent simulation step. Again, this discussion is speculative and might be incomplete. We do not rule out that other aspects and phenomena contribute to the performance differences.

In general, all complex solvers only improve the performance compared to simplistic EOS solvers in case of complex scenarios. Again, think of one layer of fluid particles on a solid plane. Even, if one has millions or billions of such particles, an EOS solver will always outperform any other iterative solver. An EOS solver can successfully handle such a scenario very efficiently. Pressure is just computed within in one iteration, e.g. Becker et al. [4] use $P = B((\frac{p}{\rho_0})^y - 1)$.

All alternative iterative solvers require a certain minimal number of iterations, e.g. PCISPH and IISPH state a minimum of three iterations. DFSPH requires at least two iterations, LPSPH [15] requires three iterations. So, all these solvers are less efficient than an EOS solver, e.g. [4] in the outlined scenario. Practically however, the outlined scenario with one layer of fluid particles is not relevant. Instead, we are interested in flow phenomena that involve certain fluid depths. For such scenarios, PCISPH showed a huge performance gain over WCSPH[36], IISPH showed a performance gain over PCISPH [21] and DFSPH showed a performance gain over IISPH [5]. But, none of the respective publications showed a general proof that they are always faster for arbitrary scenarios, because that is simply not the case.

7 Conclusion

We have presented an analysis of source terms in ISPH. Based on shortcomings of the standard forms VD and DI, we have proposed a novel form VD+PS with minimal artificial viscosity, while a density-invariant particle sampling is preserved. The proposed concept requires the solution of two PPEs. We propose to interpret the solution of a PPE with DI source term as a variant to realize particle shift in a global way. We further propose to combine a VD PPE solver with the proposed global particle shift. It is particularly remarkable that solving the two PPEs does not only improve the quality of the computed velocity field and the visual quality of the simulation, but can also be faster than solving one PPE when using VD or DI. Our measurements indicate that the improved convergence of the solver in VD+PS is due to the improved combined quality of the velocity field and its sampling as a result of the pressure computation.

The implementation of the source term is not the only degree-of-freedom when realizing an incompressible SPH fluid solver. The pressure Laplacian could be implemented in various ways and there exist also various options for the boundary handling. E.g., Cummins and Rudman [11] discuss various SPH discretizations of the Laplacian that could be analyzed and compared in terms of the resulting velocity field and also in terms of performance.

References

- Adams, B., Pauly, M., Keiser, R., Guibas, L.: Adaptively sampled particle fluids. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 26, pp. 48:1–48:7 (2007)
- Akinci, N., Akinci, G., Teschner, M.: Versatile surface tension and adhesion for SPH fluids. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* **32**(6), 182:1–182:8 (2013)
- Akinci, N., Ihmsen, M., Akinci, G., Solenthaler, B., Teschner, M.: Versatile rigid-fluid coupling for incompressible SPH. *ACM Transactions on Graphics (TOG)* **31**(4), 62 (2012)
- Becker, M., Teschner, M.: Weakly compressible SPH for free surface flows. In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 209–217 (2007)
- Bender, J., Koschier, D.: Divergence-free Smoothed Particle Hydrodynamics. In: *Proceedings of the 2015 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM (2015)
- Bender, J., Mueller, M., Otaduy, M.A., Teschner, M., Macklin, M.: A survey on position-based simulation methods in computer graphics. *Computer Graphics Forum* **33**(6), 228–251 (2014)
- Bodin, K., Lacoursire, C., Servin, M.: Constraint fluids. *IEEE Transactions on Visualization and Computer Graphics* **18**(3), 516–526 (2012)
- Chorin, A.J.: Numerical solution of the Navier-Stokes equations. *Mathematics of Computation* **22**(104), 745–762 (1968)
- Cline, M.B., Pai, D.K.: Post-stabilization for rigid body simulation with contact and constraints. In: *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 3, pp. 3744–3751. IEEE (2003)
- Cornelis, J., Ihmsen, M., Peer, A., Teschner, M.: IISPH-FLIP for incompressible fluids. In: *Computer Graphics Forum*, vol. 33, pp. 255–262. Wiley Online Library (2014)
- Cummins, S., Rudman, M.: An SPH projection method. *Journal of Computational Physics* **152**(2), 584–607 (1999)
- FIFTY2 Technology GmbH: PreonLab. <http://www.fifty2.eu> (2017). [Online; accessed 21-Dec-2017]
- Fürstenau, J.P., Avci, B., Wriggers, P.: A comparative numerical study of pressure-poisson-equation discretization strategies for SPH. In: *12th International SPHERIC Workshop, Ourense, Spain, June*, pp. 1–8 (2017)
- Geier, M., Greiner, A., Korvink, J.: Galilean invariant viscosity term for an thermal integer lattice Boltzmann automaton in three dimensions. *NSTI Nanotechnology Conference and Trade Show* p. 255258 (2004)
- He, X., Liu, N., Li, S., Wang, H., Wang, G.: Local poisson sph for viscous incompressible fluids. In: *Computer Graphics Forum*, vol. 31, pp. 1948–1958. Wiley Online Library (2012)
- He, X., Liu, N., Wang, H., Wang, G.: Local Poisson SPH for viscous incompressible fluids. *Computer Graphics Forum* **31**, 1948–1958 (2012)
- Hess, B.: Determining the shear viscosity of model liquids from molecular dynamics simulations. *The Journal of Chemical Physics* **116**(1), 209 (2002)
- Hu, X., Adams, N.A.: An incompressible multi-phase SPH method. *Journal of computational physics* **227**(1), 264–278 (2007)
- Ihmsen, M., Akinci, N., Becker, M., Teschner, M.: A parallel SPH implementation on multi-core CPUs. *Computer Graphics Forum* (2011). DOI 10.1111/j.1467-8659.2010.01832.x
- Ihmsen, M., Akinci, N., Gissler, M., Teschner, M.: Boundary handling and adaptive time-stepping for PCISPH. In: *Proceedings VRIPHYS*, pp. 79–88 (2010)

21. 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)
22. Ihmsen, M., Orthmann, J., Solenthaler, B., Kolb, A., Teschner, M.: SPH fluids in computer graphics. *Eurographics 2014 - State of the Art Reports* pp. 21–42 (2014)
23. Kang, N., Sagong, D.: Incompressible SPH using the divergence-free condition. In: *Computer Graphics Forum*, vol. 33, pp. 219–228. Wiley Online Library (2014)
24. Khayyer, A., Gotoh, H., Shao, S.: Enhanced predictions of wave impact pressure by improved incompressible SPH methods. *Applied Ocean Research* **31**(2), 111 – 131 (2009). DOI 10.1016/j.apor.2009.06.003
25. Macklin, M., Müller, M.: Position based fluids. *ACM Transactions on Graphics (TOG)* **32**(4), 104 (2013)
26. Monaghan, J.: Smoothed particle hydrodynamics. *Ann. Rev. Astron. Astrophys.* **30**, 543–574 (1992)
27. Monaghan, J.: Simulating free surface flows with SPH. *Journal of Computational Physics* **110**(2), 399–406 (1994)
28. Morris, J., Fox, P., Zhu, Y.: Modeling low Reynolds number incompressible flows. *Journal of Computational Physics* **136**, 214–226 (1997)
29. Mueller, M., Charypar, D., Gross, M.: Particle-based fluid simulation for interactive applications. In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 154–159 (2003)
30. Nestor, R., Basa, M., Quinlan, N.: Moving boundary problems in the finite volume particle method. In: *3rd ERCOFTAC SPHERIC Workshop on SPH Applications*, Lausanne, Switzerland, June, pp. 118–123 (2008)
31. Peer, A., Gissler, C., Band, S., Teschner, M.: An implicit SPH formulation for incompressible linearly elastic solids. *Computer Graphics Forum* pp. n/a–n/a. DOI 10.1111/cgf.13317. URL <http://dx.doi.org/10.1111/cgf.13317>
32. Peer, A., Ihmsen, M., Cornelis, J., Teschner, M.: An implicit viscosity formulation for SPH fluids. *ACM Trans. Graph.* **34**(4), 114:1–114:10 (2015)
33. Peer, A., Teschner, M.: Prescribed velocity gradients for highly viscous SPH fluids with vorticity diffusion. *IEEE Transactions on Visualization and Computer Graphics* **23**(12), 2656–2662 (2017)
34. Shao, S., Lo, Y.: Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Advances in water resources* **26**(7), 787–800 (2003)
35. Skillen, A., Lind, S., Stansby, P.K., Rogers, B.D.: Incompressible Smoothed Particle Hydrodynamics (SPH) with reduced temporal noise and generalised fickian smoothing applied to body–water slam and efficient wave–body interaction. *Computer Methods in Applied Mechanics and Engineering* **265**, 163–173 (2013)
36. Solenthaler, B., Pajarola, R.: Predictive-corrective incompressible SPH. *ACM Transactions on Graphics (Proc. SIGGRAPH)* **28**, 40:1–40:6 (2009)
37. Takahashi, T., Dobashi, Y., Nishita, T., Lin, M.: An efficient hybrid incompressible SPH solver with interface handling for boundary conditions. *Computer Graphics Forum* (2016)
38. Tartakovsky, A.M., Panchenko, A.: Pairwise force Smoothed Particle Hydrodynamics model for multiphase flow. *J. Comput. Phys.* **305**(C), 1119–1146 (2016)
39. Winchenbach, R., Hochstetter, H., Kolb, A.: Constrained neighbor lists for SPH-based fluid simulations. In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 49–56. Eurographics Association (2016)
40. 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>
41. Xu, R., Stansby, P., Laurence, D.: Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *Journal of Computational Physics* **228**(18), 6703–6725 (2009)
42. Yan, X., Jiang, Y.T., Li, C.F., Martin, R.R., Hu, S.M.: Multiphase SPH simulation for interactive fluids and solids. *ACM Trans. Graph.* **35**(4), 79:1–79:11 (2016). DOI 10.1145/2897824.2925897. URL <http://doi.acm.org/10.1145/2897824.2925897>
43. Yang, T., Lin, M.C., Martin, R.R., Chang, J., Hu, S.M.: Versatile interactions at interfaces for SPH-based simulations. In: L. Kavan, C. Wojtan (eds.) *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*. The Eurographics Association (2016)
44. Yang, T., Martin, R.R., Lin, M.C., Chang, J., Hu, S.M.: Pairwise force SPH model for real-time multi-interaction applications. *IEEE Transactions on Visualization & Computer Graphics* **23**(10), 2235–2247 (2017)