Prescribed Velocity Gradients for Highly Viscous SPH Fluids with Vorticity Diffusion

Andreas Peer and Matthias Teschner

Abstract—Working with prescribed velocity gradients is a promising approach to efficiently and robustly simulate highly viscous SPH fluids. Such approaches allow to explicitly and independently process shear rate, spin, and expansion rate. This can be used to, e.g., avoid interferences between pressure and viscosity solvers. Another interesting aspect is the possibility to explicitly process the vorticity, e.g. to preserve the vorticity.

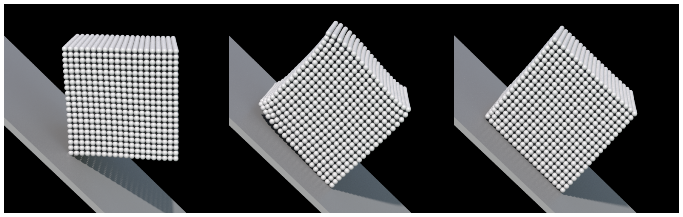
In this context, this paper proposes a novel variant of the prescribed-gradient idea that handles vorticity in a physically motivated way. In contrast to a less appropriate vorticity preservation that has been used in a previous approach, vorticity is diffused. The paper illustrates the utility of the vorticity diffusion. Therefore, comparisons of the proposed vorticity diffusion with vorticity preservation and additionally with vorticity damping are presented. The paper further discusses the relation between prescribed velocity gradients and prescribed velocity Laplacians which improves the intuition behind the prescribed-gradient method for highly viscous SPH fluids. Finally, the paper discusses the relation of the proposed method to a physically correct implicit viscosity formulation.

1 Introduction

The SPH method [1] is a well-established variant for simulating fluids in computer graphics [2]. Its Lagrangian nature allows for an efficient simulation of free-surface scenarios with small-scale effects, but it can also handle large-scale scenarios, e.g. [3]. The inherent mass preservation and improvements in volume preservation [4], [5], [6], boundary handling [7], [8], [9], multiphase simulations [10], [11] and surface tension effects [12], [13] make it a popular choice in animation.

Simulating highly viscous fluids with SPH is a challenging task. As viscosity is an important stability criterion for SPH simulations, for low viscosity fluids, several explicit formulations have been proposed, e.g. [14], [15], [16]. However, due to the explicit nature and the associated small timesteps, simulating highly viscous fluids is not practical. Only recently, various implicit viscosity formulations have been proposed. The formulation introduced by Takahashi et al. [17] is based on the variational principle that has been originally proposed for grids [18]. While being in the tradition of previous grid-based implicit viscosity formulations [19], [20], [21], it improves the handling of the free-surface boundary to capture rotational motions. This allows the simulation of realistic buckling and coiling effects.

As a versatile and efficient alternative to standard implicit formulations, Peer et al. [22] propose the usage of prescribed velocity gradients that capture many of the effects of true viscosity, and Bender and Koschier [6] adopt this idea to prescribe shear rates. The idea of [22] focuses on two principles. First, the viscosity solver should not interfere with the pressure solver. Second, the viscosity solver should approximately preserve rotational motion, i.e. vorticity. This is in contrast to [6] which does not impose any restrictions on vorticity. In separating shear viscosity from pressure and vorticity, Peer et al. propose an efficient and simple model for the simulation of highly viscous fluids that can easily be integrated into existing frameworks. However, while orthogonality to the pressure solve is a physical property of shear viscosity, orthogonality to vorticity is not.



(a) Vorticity damping (b) Vorticity preservation (c) Vorticity diffusion

Fig. 1. A viscous fluid cube on an inclined plane. All three examples are simulated with the method of prescribed velocity gradients. However, the examples differ in the handling of the vorticity. In case of vorticity damping, the cube does not rotate. With vorticity preservation as in [22], the cube rotates, but deforms. With the proposed vorticity diffusion, the cube rotates with minimal distortion.

This aspect is illustrated in Fig. 1 where three variants of the idea of prescribed velocity gradients are shown. First, vorticity damping prevents viscous fluids from rotating. Second, approximate vorticity preservation as proposed in [22] enables rotating viscous fluids, but introduces implausible distortion. Third, vorticity diffusion results in the expected behavior of a rotating fluid without distortion. The artifacts in case of vorticity damping and vorticity preservation are due to the fact that both variants are not physically motivated. Realizing viscous fluids by simply minimizing relative velocities of adjacent particles as, e.g. in XSPH [1], [14], does not only counteract shear rates, but also vorticity. This vorticity damping might be tolerable for low viscosity fluids, but leads to visible artifacts for highly viscous fluids. On the other hand, minimizing shear rates with approximate vorticity preservation as proposed in [22] reduces rotation artifacts, but introduces artificial distortions. While the vorticity preservation is certainly motivated by existing

- *A. Peer and M. Teschner are with University of Freiburg, Germany.*

vorticity confinement methods, e.g. [23], [24], [25], [26], or when involving interpolation to an underlying grid [27], it is less appropriate for highly viscous fluids.

Contribution: We describe and analyze three relevant variants to handle the vorticity in prescribed velocity gradients for highly viscous fluids, i.e. vorticity damping, vorticity preservation as in [22], and vorticity diffusion. We show that vorticity damping and vorticity preservation suffer from rotation and distortion issues. We propose a novel vorticity diffusion scheme to resolve these problems.

Organization: Sec. 2 briefly introduces the general concepts of explicit and implicit formulations for highly viscous fluids. Sec. 2.2 and Sec. 2.3 introduce and discuss the relation between prescribed velocity gradients and prescribed velocity Laplacians. Both formulations can be used to realize the diffusion of the velocity field, while prescribed gradients allow for a more intuitive specification of target values. I.e., targets for shear rate, spin and expansion rate can be separately processed. In the following, Sec. 3 discusses three variants for the specification of target gradients. These variants differ in the handling of vorticity. The first method proposed in Sec. 3.1 is based on the intuition that viscous accelerations minimize the Laplacian of the velocity field. This variant damps the vorticity. Sec. 3.2 discusses the target gradient proposed by Peer et al. [22]. This variant approximately preserves the vorticity. Sec. 3.3 describes the proposed way to specify target gradients where viscosity is diffused. The approach is motivated and explained in Sec. 4. Sec. 5 discusses the relation of our method to Takahashi's technique [17] which is based on a physically correct implicit viscosity formulation. In particular, conceptual differences and performance aspects are addressed. Finally, Sec. 6 compares the three variants of vorticity handling, i.e. damping, preservation and the proposed diffusion. Properties of the proposed viscosity approach are illustrated. The computational overhead compared to [22] is discussed. Various comparisons to Takahashi's technique [17] illustrate practical aspects of the relation to our approach.

2 Concepts for High Viscosities

In Newtonian, incompressible fluids, viscosity is handled by the acceleration term $\nu\nabla^2\mathbf{v}_i$, where $\mathbf{v}_i$ denotes the velocity of particle i, and ν the constant for kinematic viscosity. The acceleration is proportional to the difference of the velocity at a particle and the mean velocity of its neighbors. The main goal of the viscosity term is the reduction of shear rates in the fluid flow, but in practice, it also affects the vorticity and the expansion rate of the fluid. This section discusses various approaches based on $\nu\nabla^2\mathbf{v}_i$ to handle highly viscous fluids.

2.1 Explicit and implicit formulations

Low viscous SPH fluid simulations generally work with explicit formulations, i.e. $\frac{D\mathbf{v}_i(t)}{Dt} = \nu\nabla^2\mathbf{v}_i(t)$, assuming spatially constant viscosity. Due to prohibitive timestep restrictions, however, highly viscous fluid simulations prefer the implicit formulation $\frac{D\mathbf{v}_i(t)}{Dt} = \nu\nabla^2\mathbf{v}_i(t+\Delta t)$ that results in a linear system of the form $(\mathbf{I}-\Delta t\nu\nabla^2)\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i(t)$, e.g. [17], [18], [19], [20], [21], [28], [29], [30]. Instead of explicitly computing a viscous velocity change per particle, the implicit form globally computes the final velocity field. This allows for a robust handling of high viscosities at large timesteps. Instead of considering the current velocity $\mathbf{v}_i(t)$ in the system, the implicit formulation typically employs an intermediate velocity $\mathbf{v}_i^*$ after external accelerations and pressure solve, i.e.

$$(\mathbf{I} - \Delta t\nu\nabla^2)\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i^*. \tag{1}$$

2.2 Prescribed Laplacian

An alternative to the implicit viscosity formulation is the computation of final velocities $\mathbf{v}_i(t+\Delta t)$ that adhere to a prescribed vector Laplacian $\nabla_\tau^2\mathbf{v}_i^*$, as proposed by [22]. This results in a linear system of the form

$$\nabla^2\mathbf{v}_i(t+\Delta t) = \nabla_\tau^2\mathbf{v}_i^*. \tag{2}$$

The notation $\nabla_\tau^2\mathbf{v}_i^*$ indicates that the prescribed Laplacian for the velocities $\mathbf{v}_i(t+\Delta t)$ is derived from the intermediate velocity field $\mathbf{v}_i^*$ after external accelerations and pressure solve. It generally differs from the actual Laplacian of the velocities $\mathbf{v}_i^*$, e.g., $\nabla_\tau^2\mathbf{v}_i^* = \xi\nabla^2\mathbf{v}_i^*$ with $0 \le \xi < 1$ could be used (see also Sec. 3.1).

Please refer to [22] for a derivation of this method, where Eq. 2 is particularly discussed in Sec. 5.

2.3 Prescribed gradient

In order to independently process shear rate, expansion rate and vorticity, Eq. 2 can be discretized in a specific way that reformulates the prescribed Laplacian with a prescribed velocity gradient [22]. Using XSPH, we get $\nabla^2\mathbf{v}_i \approx \sum_j \frac{m_j}{\rho_j}\mathbf{v}_{ji}W_{ij}$, with $\mathbf{v}_{ji} = \mathbf{v}_j - \mathbf{v}_i$. And using the approximation $\mathbf{v}_j \approx \mathbf{v}_i + \frac{\nabla\mathbf{v}_i+\nabla\mathbf{v}_j}{2}\mathbf{x}_{ji}$, we get $\nabla^2\mathbf{v}_i \approx \sum_j \frac{m_j}{\rho_j}\frac{\nabla\mathbf{v}_i+\nabla\mathbf{v}_j}{2}\mathbf{x}_{ji}W_{ij}$. Now, the prescribed Laplacian $\nabla_\tau^2\mathbf{v}_i^*$ can be rewritten with a prescribed velocity gradient $\nabla_\tau\mathbf{v}_i^*$ as $\nabla_\tau^2\mathbf{v}_i^* \approx \sum_j \frac{m_j}{\rho_j}\frac{\nabla_\tau\mathbf{v}_i^*+\nabla_\tau\mathbf{v}_j^*}{2}\mathbf{x}_{ji}W_{ij}$. Finally using XSPH to discretize the left-hand side of Eq. 2, results in the linear system

$$\sum_j \frac{m_j}{\rho_j}\mathbf{v}_{ji}(t+\Delta t)W_{ij} = \sum_j \frac{m_j}{\rho_j}\frac{\nabla_\tau\mathbf{v}_i^*+\nabla_\tau\mathbf{v}_j^*}{2}\mathbf{x}_{ji}W_{ij} \tag{3}$$

which has been proposed in [22]. In contrast to Eq. 2, the discretized reformulation in Eq. 3 allows to derive a prescribed target velocity gradient $\nabla_\tau\mathbf{v}_i^*$ from the actual velocity gradient $\nabla\mathbf{v}_i^*$ after pressure solve and external acceleration.

Therefore, the typical decomposition of $\nabla\mathbf{v}_i^*$ into spin $\mathbf{R}_i^*$, expansion rate $\mathbf{V}_i^*$ and shear rate $\mathbf{S}_i^*$ is considered:

$$\begin{aligned}\nabla\mathbf{v}_i^* = \underbrace{\frac{1}{2}\left(\nabla\mathbf{v}_i^* - (\nabla\mathbf{v}_i^*)^T\right)}_{\mathbf{R}_i^*} + \underbrace{\frac{1}{3}(\nabla\cdot\mathbf{v}_i^*)\mathbf{I}}_{\mathbf{V}_i^*} + \\ \underbrace{\left(\frac{1}{2}\left(\nabla\mathbf{v}_i^* + (\nabla\mathbf{v}_i^*)^T\right) - \frac{1}{3}(\nabla\cdot\mathbf{v}^*)\mathbf{I}\right)}_{\mathbf{S}_i^*}.\end{aligned} \tag{4}$$

Now, target values $\mathbf{R}_i^\tau$, $\mathbf{V}_i^\tau$, $\mathbf{S}_i^\tau$ are derived from $\mathbf{R}_i^*$, $\mathbf{V}_i^*$, $\mathbf{S}_i^*$ and the target gradient in Eq. 3 is computed as $\nabla_\tau\mathbf{v}_i^* = \mathbf{R}_i^\tau + \mathbf{V}_i^\tau + \mathbf{S}_i^\tau$. Variants to specify the target values $\mathbf{R}_i^\tau$, $\mathbf{V}_i^\tau$, $\mathbf{S}_i^\tau$ are discussed in Sec. 3.

3 Computation of the target gradient

There exist various ways to define the target gradient and this section discusses three options. A comparative analysis follows in Sec. 6.

3.1 Vorticity damping

The minimization of the velocity Laplacian over time is an important effect of the viscosity acceleration $\nu\nabla^2\mathbf{v}_i$. Therefore, one option is to prescribe the target Laplacian as $\nabla^2_\mathcal{T}\mathbf{v}_i^* = \xi\nabla^2\mathbf{v}_i^*$ with $0 \leq \xi < 1$. Using the gradient form from Sec. 2.3, this corresponds to

$$\nabla^2\mathbf{v}_i(t+\Delta t) = \xi\sum_j \frac{m_j}{\rho_j}\frac{\nabla\mathbf{v}_i^* + \nabla\mathbf{v}_j^*}{2}\mathbf{x}_{ji}W_{ij}. \tag{5}$$

Applying the decomposition from Eq. 4, we get

$$\begin{aligned}&\nabla^2\mathbf{v}_i(t+\Delta t) = \\ &\sum_j \frac{m_j}{\rho_j}\frac{\xi(\mathbf{R}_i^* + \mathbf{V}_i^* + \mathbf{S}_i^*) + \xi(\mathbf{R}_j^* + \mathbf{V}_j^* + \mathbf{S}_j^*)}{2}\mathbf{x}_{ji}W_{ij}.\end{aligned} \tag{6}$$

Thus, minimizing the Laplacian over time corresponds to damping spin, expansion rate and shear rate over time. From Eq. 6 follows that in the vorticity damping case, the target gradient is computed as

$$\nabla_\mathcal{T}\mathbf{v}_i^\mathcal{T} = \xi\mathbf{R}_i^* + \xi\mathbf{V}_i^* + \xi\mathbf{S}_i^*. \tag{7}$$

3.2 Vorticity preservation

In contrast to a uniform minimization of all components of the velocity gradient, Peer et al. [22] propose a target gradient that preserves the spin, i.e. $\mathbf{R}_i^\mathcal{T} = \mathbf{R}_i^*$. Further, the expansion rate is preserved $\mathbf{V}_i^\mathcal{T} = \mathbf{V}_i^*$. This is motivated by the fact that the viscosity solver should not interfere with the result of the pressure solver. As $\mathbf{v}_i^*$ is the velocity after pressure solve, the expansion rate of the respective velocity gradient $\nabla\mathbf{v}_i^*$ encodes a volume change that is induced by the pressure solver to preserve the incompressibility of the SPH fluid. As this volume change should be preserved, the expansion rate should not be changed by the viscosity solver, thus $\mathbf{V}_i^\mathcal{T} = \mathbf{V}_i^*$. The target gradient is therefore computed as

$$\nabla_\mathcal{T}\mathbf{v}_i^\mathcal{T} = \mathbf{R}_i^* + \mathbf{V}_i^* + \xi\mathbf{S}_i^*. \tag{8}$$

3.3 Vorticity diffusion

The two target gradients in Eqs. 7 and 8 differ in the handling of the vorticity. The first variant damps the vorticity, while the second one preserves it. Both variants, however, are not physically motivated. From $\frac{D\mathbf{v}_i(t)}{Dt} = \nu\nabla^2\mathbf{v}_i(t)$, it can be seen that viscosity diffuses momentum. By taking the curl of this equation, it transforms into a vorticity form $\frac{D\boldsymbol{\omega}_i(t)}{Dt} = \nu\nabla^2\boldsymbol{\omega}_i(t)$, with $\boldsymbol{\omega}_i = \nabla\times\mathbf{v}_i$ being the vorticity. Here, it can be seen that viscosity also diffuses vorticity.

In contrast to the previously employed target gradient, we propose a novel variant of the target gradient that does not only preserve the expansion rate, but also realizes vorticity diffusion. This approach and its implementation are described in the following Sec. 4.

4 Prescribed velocity gradients for vorticity-diffusing viscosity

Following the idea of the prescribed velocity gradient, we start with a velocity field $\mathbf{v}_i(t)$, and external and pressure forces are computed to predict intermediate velocities $\mathbf{v}_i^*$. For this intermediate velocity field, velocity gradients $\nabla\mathbf{v}_i^*$ are computed with SPH as $\nabla\mathbf{v}_i^* = \frac{1}{\rho_i}\sum_j m_j(\mathbf{v}_j^* - \mathbf{v}_i^*)\otimes\nabla W_{ij}$ [1] and Eq. 4 is used to extract spin $\mathbf{R}_i^*$, expansion rate $\mathbf{V}_i^*$ and shear rate $\mathbf{S}_i^*$ from the velocity gradient $\nabla\mathbf{v}_i^*$.

Now, the target gradient $\nabla_\mathcal{T}\mathbf{v}_i^* = \mathbf{R}_i^\mathcal{T} + \mathbf{V}_i^\mathcal{T} + \mathbf{S}_i^\mathcal{T}$ is computed in a novel way. While target expansion rate $\mathbf{V}_i^\mathcal{T}$ and target shear rate $\mathbf{S}_i^\mathcal{T}$ are computed as proposed in [22] with $\mathbf{V}_i^\mathcal{T} = \mathbf{V}_i^*$ and $\mathbf{S}_i^\mathcal{T} = \xi\mathbf{S}_i^*$, the target vorticity $\mathbf{R}_i^\mathcal{T}$ is computed from a diffusion process. The general idea of the proposed vorticity-diffusing viscosity solver is depicted in Alg. 1, while details of the vorticity diffusion to compute $\mathbf{R}_i^\mathcal{T}$ are described in Sec. 4.1.

Algorithm 1 Vorticity-diffusing viscous SPH fluid solver

```
compute external and pressure forces
update velocity: $\mathbf{v}_i(t) \rightarrow \mathbf{v}_i^*$
compute velocity gradient $\nabla\mathbf{v}_i^*$
decompose velocity gradient into $\mathbf{R}_i^*$, $\mathbf{V}_i^*$ and $\mathbf{S}_i^*$
compute target shear and target expansion rate $\mathbf{S}_i^* \rightarrow \mathbf{S}_i^\mathcal{T}, \mathbf{V}_i^* \rightarrow \mathbf{V}_i^\mathcal{T}$
diffusion process to compute target vorticity $\mathbf{R}_i^* \rightarrow \mathbf{R}_i^\mathcal{T}$
compute the target gradient $\nabla\mathbf{v}_i^\mathcal{T} = \mathbf{R}_i^\mathcal{T} + \mathbf{V}_i^\mathcal{T} + \mathbf{S}_i^\mathcal{T}$
solve velocity reconstruction system
update velocity: $\mathbf{v}^* \rightarrow \mathbf{v}(t+\Delta t)$
update position: $\mathbf{x}(t) \rightarrow \mathbf{x}(t+\Delta t)$
```

4.1 Target vorticity based on diffusion

As already outlined in Sec. 3.3, viscosity diffuses vorticity according to $\frac{D\boldsymbol{\omega}_i(t)}{Dt} = \nu\nabla^2\boldsymbol{\omega}_i(t)$. In order to consider this effect in the proposed target vorticity $\mathbf{R}_i^\mathcal{T}$, we propose to solve the system

$$\nabla^2\boldsymbol{\omega}_i^\mathcal{T} = \xi\nabla^2\boldsymbol{\omega}_i^* \tag{9}$$

with $0 \leq \xi \leq 1$ for all particles i. The value $\boldsymbol{\omega}_i^*$ is extracted from the spin tensor $\mathbf{R}_i^*$ using

$$\mathbf{R}^* = \frac{1}{2}\begin{pmatrix} 0 & -\omega_z^* & \omega_y^* \\ \omega_z^* & 0 & -\omega_x^* \\ -\omega_y^* & \omega_x^* & 0 \end{pmatrix}, \quad \boldsymbol{\omega}^* = \begin{pmatrix} \omega_x^* \\ \omega_y^* \\ \omega_z^* \end{pmatrix}. \tag{10}$$

and the same relation is used to generate the target spin tensor $\mathbf{R}_i^\mathcal{T}$ from $\boldsymbol{\omega}_i^\mathcal{T}$. Considering the system in Eq. 9 is motivated by the idea that a diffusion process minimizes the Laplacian over time (see Sec. 3.1).

4.2 Solving the system

For solving the system in Eq. 9, we discretize both sides with XSPH:

$$\sum_j \frac{m_j}{\rho_j}\boldsymbol{\omega}_{ji}^\mathcal{T}W_{ij} = \xi\sum_j \frac{m_j}{\rho_j}\boldsymbol{\omega}_{ji}^*W_{ij} \tag{11}$$

This system can be rewritten according to [22] as

$$\frac{1}{\rho_i}\sum_j m_j\boldsymbol{\omega}_{ij}^\mathcal{T}W_{ij} = \xi\frac{1}{\rho_i}\sum_j m_j\boldsymbol{\omega}_{ij}^*W_{ij} \tag{12}$$

which can be transformed into the symmetric form

$$\sum_j m_j \boldsymbol{\omega}^{\tau}_{ij} W_{ij} = \xi \sum_j m_j \boldsymbol{\omega}^{*}_{ij} W_{ij} \tag{13}$$

if $m_j = m_i$. Rewriting the equation as

$$\begin{aligned}\boldsymbol{\omega}^{\tau}_{i} (\sum_j m_j W_{ij} - m_i W_{ii}) + \sum_{j \neq i} -m_j W_{ij} \boldsymbol{\omega}^{\tau}_{j} \\ = \xi \sum_j m_j \boldsymbol{\omega}^{*}_{ij} W_{ij}\end{aligned} \tag{14}$$

gives the coefficients of the respective system $\mathbf{A}\boldsymbol{\omega}^{\tau} = \mathbf{b}$ as $a_{ii} = \rho_i - m_i W_{ii}$ and $a_{ij} = -m_j W_{ij}$. The source vector $\mathbf{b}$ is composed of $b_i = \xi \sum_j m_j \boldsymbol{\omega}^{*}_{ij} W_{ij}$.

5 Discussion

5.1 Conceptual differences to Takahashi's approach [17]

Takahashi et al. [17] is an implicit viscosity formulation for SPH which is adapted from [18]. While [17] solves $\mathbf{v}_i = \mathbf{v}^*_i + \Delta t \nabla \cdot \nu_i \big(\nabla \mathbf{v}_i + (\nabla \mathbf{v}_i)^T\big)$, which reduces to Eq. 1 for incompressible and constant viscosity fluids in the continuous case, our technique solves the system in Eq. 2 (see [22]). In contrast to [17], the approach in [22] does not interfere with the pressure solver and both solvers do not affect each other. Like [22], our approach also eliminates interference between pressure and viscosity.

On the other hand, it is interesting to note that Eq. 1 and Eq. 2 converge towards each other for infinite viscosity. If we divide Eq. 1 by ν and ν runs to infinity

$$\lim_{\nu -> \infty} \Big(\frac{1}{\nu}\mathbf{I} - \Delta t \nabla^2\Big) \mathbf{v}_i(t + \Delta t) = \lim_{\nu -> \infty} \frac{1}{\nu} \mathbf{v}^*_i, \tag{15}$$

we end up

$$\nabla^2 \mathbf{v}_i(t + \Delta t) = \mathbf{0}. \tag{16}$$

which corresponds to Eq. 2 for $\nabla^2_{\tau} \mathbf{v}^*_i = \xi \nabla^2 \mathbf{v}^*_i$ with $\xi = 0$. This might be an indicator that vorticity diffusion is also a problem in approaches that are based on Eq. 1. The experiment discussed in Sec. 6.11 and shown in Fig. 10 addresses this issue. Here, two highly viscous cubes collide. While the cubes start rotating due to the collision using our approach, the induced vorticity is instantly damped out by [17].

5.2 Performance

The system in [17] is fully-coupled, leading to a size of $3N \times 3N$ with N being the particle number. Furthermore, each equation considers first- and second-ring neighbors, leading to up to 320 non-zero entries per row. Therefore, the complexity of a solver iteration is $3N \cdot 320 = 960N$. Our method, on the other side, solves six systems of size $N \times N$ and all these systems share the same system matrix. Furthermore, only first-ring neighbors are required, resulting in about 30 unknowns per row. Thus, the complexity per solver iteration is $6N \cdot 30 = 180N$, reducing the complexity by more than a factor of five compared to Takahashi's approach. Furthermore, as we solve the same system six times, any preconditioner or factorization of the system matrix has to be computed only once.

In addition to the reduced complexity per solver iteration, our approach typically requires fewer iterations than [17] for similar material properties. This is emphasized by all comparative experiments in Sec. 6.

6 Results

In this section, we illustrate the properties of the proposed target-gradient approach that considers vorticity diffusion (Sec. 3.3). We particularly focus on comparisons with the two alternative variants for prescribing the velocity gradient, namely vorticity damping (Sec. 3.1) and vorticity preservation (Sec. 3.2).

In all presented experiments, we use SPH [1], [2] with a cubic spline kernel [31] with an influence radius of twice the particle spacing. Pressure and pressure forces are computed with IISPH by solving a pressure Poisson equation [5]. The maximum overall deviation of the fluid volume in all scenarios is less than 0.1%. Solids are sampled with particles. One- and two-way coupling of fluid and solids, as well as surface tension, is realized with the methods of Akinci et al. [8], [12]. Number-density formulations are used for fluids with multiple phases [10]. Fluid-air interfaces are reconstructed with dual contouring [32]. Our implementation is fully parallelized [33]. All experiments are computed on a 16-core Intel Xeon workstation with 3.10 GHz using 50 frames per second.

6.1 Cube in a rotational force field

This scenario illustrates a highly viscous fluid cube in a rotational force field. The cube consists of 8k particles. The particle spacing is 50mm. The timestep is 5ms which corresponds to 4 substeps, i.e. simulation steps per frame. ξ is set to zero. Fig. 2 shows the scenario and Tab. 1 shows performance details. The scenario illustrates that the correct handling of spin is an issue for viscosity with vorticity damping and vorticity preservation, and that our proposed approach with added vorticity diffusion significantly improves the spin handling.

	vorticity damping	vorticity preservation	vorticity diffusion
meshing	6 ms	6 ms	6 ms
ext. forces	4 ms	4 ms	4 ms
pressure	6 ms	6 ms	6 ms
pre-processing	12 ms	16 ms	16 ms
viscosity	36 ms	36 ms	30 ms
vorticity	-	-	30 ms
overall	64 ms	68 ms	92 ms
pressure	1 it.	1 it.	1 it.
viscosity	19 it.	17 it.	14 it.
vorticity	-	-	14 it.

TABLE 1
Cube in a rotational force field. Performance measurements are given per frame. The solver iterations for pressure, viscosity and vorticity are average values per substep.

The measurements in Tab. 1 indicate the significant computational overhead in case of vorticity diffusion. The overhead is difficult to specify in a general way. On one

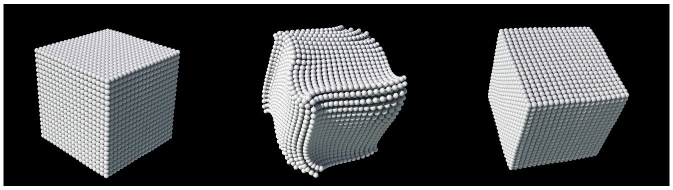

(a) Vorticity damping (b) Vorticity preservation (c) Vorticity diffusion

Fig. 2. Cube in a rotational force field. With vorticity damping, the cube does not rotate. With vorticity preservation, the cube rotates, but deforms. With the proposed vorticity diffusion, the cube rotates with minimal distortion.

hand, there is an additional system solve. On the other hand, the meshing, the pressure solve and the pre-processing for the viscosity solver is similar in all variants. The pre-processing for vorticity preservation and vorticity diffusion is more expensive as it requires the computation of the velocity gradient per particle. We have also observed that the handling of the vorticity diffusion seems to slightly improve the convergence of the viscosity solver as indicated by the solver iterations for viscosity.

6.2 Cube on an inclined plane

This scenario illustrates a highly viscous fluid cube under gravity on an inclined plane. The cube consists of 8k particles. The particle spacing is 50mm. The timestep is 0.5ms which corresponds to 40 substeps per frame. ξ is set to zero. Fig. 1 shows the scenario and Tab. 2 shows performance details. In contrast to the previous scenario, fluid-solid interaction is added which requires a smaller timestep. Again, spin handling is an issue for existing techniques [22], while the proposed vorticity diffusion improves the spin handling.

	vorticity damping	vorticity preservation	vorticity diffusion
meshing	60 ms	60 ms	60 ms
ext. forces	40 ms	40 ms	40 ms
pressure	60 ms	60 ms	60 ms
pre-processing	160 ms	190 ms	190 ms
viscosity	800 ms	530 ms	400 ms
vorticity	-	-	750 ms
overall	1120 ms	880 ms	1500 ms
pressure	1 it.	1 it.	1 it.
viscosity	44 it.	26 it.	20 it.
vorticity	-	-	37 it.

TABLE 2
Cube on an inclined plane. Performance measurements are given per frame. The solver iterations for pressure, viscosity and vorticity are average values per substep.

Comparing Tab. 1 and Tab. 2 indicates that the boundary handling negatively affects the convergence of the viscosity and vorticity solvers. The tables also show that the relative computational overhead of the additional vorticity solver varies compared to the variants without vorticity diffusion.

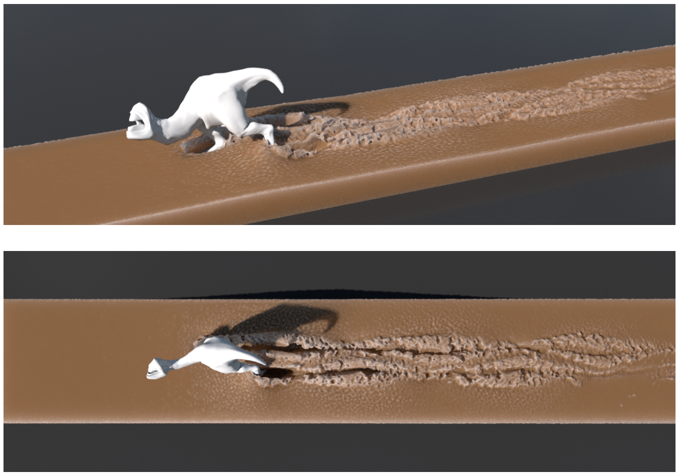


Fig. 3. A creature walking in mud. The creature model is courtesy of Eric Mootz.

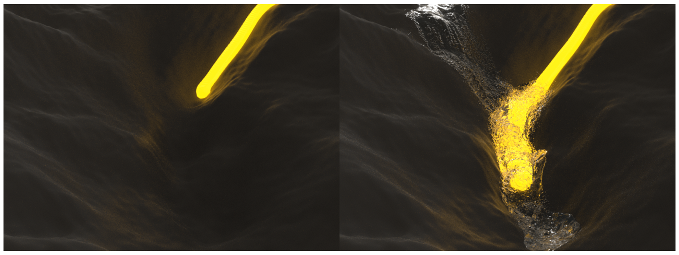


Fig. 4. Multiple phases. Lava mixes with water.

6.3 Interaction with scripted geometry

The scenario in Fig. 3 is simulated with the proposed vorticity diffusion. It shows the interaction of a highly viscous fluid with scripted geometry. The particle count is 2.9M, the spacing is 35mm and the timestep is 1ms. ξ is set to 0.5. The average computation time per frame is 43s. On average, 2.3 pressure iterations, 1 vorticity iteration and 7 viscosity iteration are computed. The low number of vorticity iterations indicates that spin handling is less challenging in this scenario.

6.4 Multiple phases

Fig. 4 illustrates a scenario where a highly viscous fluid interacts with a solid and a low viscous fluid. The scene consists of 800k particles with a spacing of 20mm. The timestep is 2ms. The scenario is simulated with vorticity diffusion. ξ is set to 0.4 for the lava. The average computation time per frame is 12s. On average, the scenario requires 12.9 pressure iterations, 18.3 vorticity iterations and 31.4 viscosity iterations.

6.5 Large-scale scenario

Fig. 5 shows a large-scale scenario simulated with vorticity diffusion. It consists of 8M particles with a spacing of 100mm. ξ is set to 0.2. The timestep is 4ms and the average computation time per frame is 393s. 8 pressure iteration, 160 vorticity iterations and 45 viscosity iterations have been computed on average.

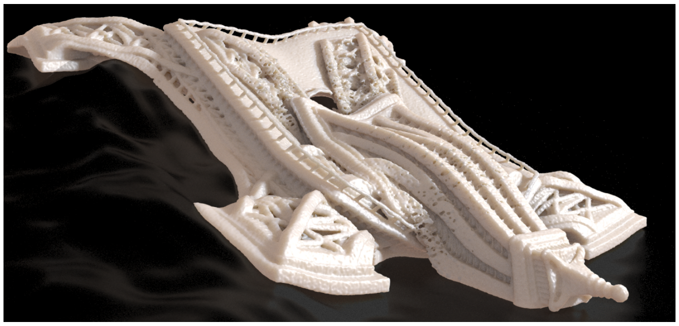


Fig. 5. Melting Eiffel tower. The model is courtesy of Pranav Panchal.

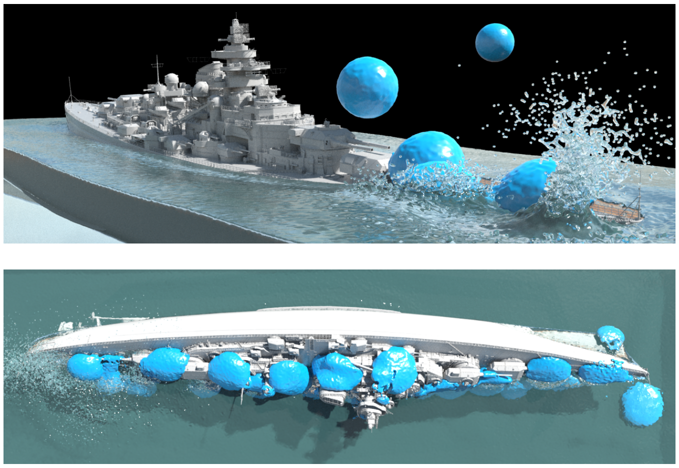


Fig. 6. Multiple phases with two-way coupled solids. A swimming ship is hit by highly viscous spheres. The ship model is courtesy of splatypi at www.blendswap.com

The scene starts with an undeformed model under gravity on an inclined plane. A significant spin is introduced during the deformation of the object resulting in a large number of iterations for diffusing the vorticity.

6.6 Multiple phase with two-way coupled solids

In the scenario in Fig. 6, highly viscous material interacts with a ship that swims in water. The scene consists of 24M particles including water and viscous spheres. The particle spacing is 60mm. The scene is simulated with a timestep of 1ms. ξ is set to 0.7 for the viscous spheres. The computation time per frame is 109s. The pressure computation required 5 iterations, vorticity diffusion 3.6 iterations and viscosity took 9.4 iterations on average.

6.7 Comparison of vorticity preservation, damping and diffusion for the Eiffel tower

The Eiffel tower in Fig. 5 has been simulated with the proposed vorticity diffusion. To illustrate the practical relevance of our approach, we have simulated the same scenario with vorticity preservation and vorticity damping as illustrated in Fig. 7. The three variants lead to significantly different results, while the vorticity diffusion in Fig. 5 shows the most plausible behavior. Please refer to the accompanying video to assess the differences in the dynamics for all three variants. Analogous to the scenario in Fig. 5, the scenarios in Fig. 7 consist of 8M particles with a spacing of 100mm. ξ is set to 0.2. Tab. 3 summarizes the solver iterations for all three variants.

	vorticity diffusion	vorticity damping	vorticity preservation
viscosity iter.	45	78	62
vorticity iter.	160	-	-

TABLE 3
Solver iterations for the scenarios in Figs. 5 and 7.

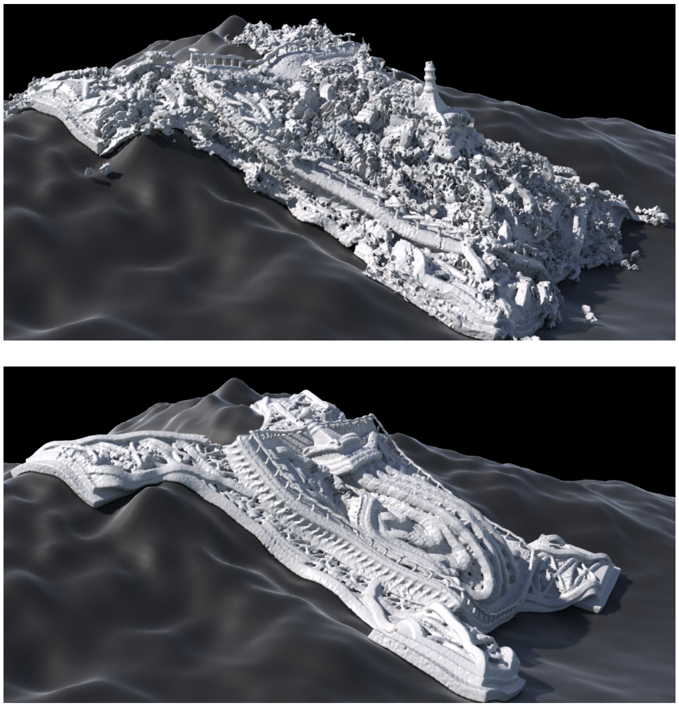


Fig. 7. Top: Vorticity damping. Bottom: Vorticity preservation.

6.8 Comparison of vorticity preservation, damping, and diffusion for the same value of ξ using Armadillos

The comparisons in the top row in Fig. 8 illustrate the relevance of vorticity diffusion in the case of time-varying spin. If spin changes over time, simulations with the same value of ξ differ for vorticity damping, preservation and diffusion. While vorticity damping and in particular vorticity preservation result in artificial, less plausible distortions, such errors are minimized with vorticity diffusion. This is particularly visible at the tail and the ears of the Armadillo in Fig. 8. Please also refer to the accompanying video to assess the differences in the dynamics. The Armadillo consists of 50k particles with a spacing of 25mm. ξ is set to 0.1. Tab. 4 summarizes the solver iterations for all three variants.

6.9 Comparison of vorticity preservation, damping, diffusion for the same value of ξ using non-rotating cubes

In this scenario, we compare vorticity damping, preservation and diffusion for a falling cube without a significant

	vorticity damping	vorticity preservation	vorticity diffusion
viscosity iter.	30	27	28
vorticity iter.	-	-	8

TABLE 4
Solver iterations for the scenario in the top row of Fig. 8.

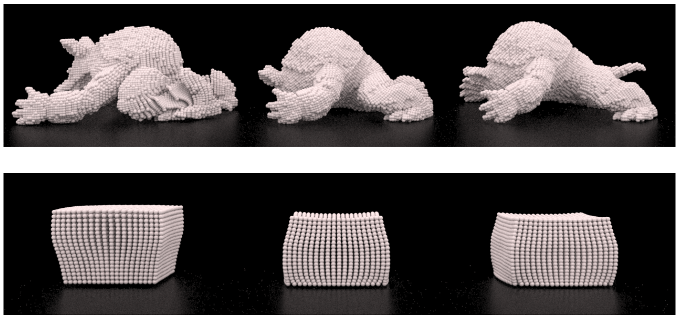


Fig. 8. Top: Vorticity damping, preservation, diffusion (left to right) for the Armadillo. Bottom: Vorticity damping, preservation, diffusion (left to right) for a non-rotating cube.

change of spin over time. As expected and illustrated in the bottom row of Fig. 8, there are less relevant differences in the simulation results. The cube consists of 8k particles with a spacing of 50mm. ξ is set to 0.5. Tab. 5 summarizes the solver iterations for all three variants. As spin does not change over time, the vorticity diffusion requires a minimum number of iterations.

6.10 Comparisons of three viscous fluids with varying values of ξ

The simulations in Fig. 9 show that our approach can be used to simulate a wide range of viscous materials. Each Armadillo consists of 50k particles with a particle spacing of 25mm. Tab. 6 summarizes the solver iterations. Analogous to [22], our approach is less appropriate for low viscous fluids such as water and also for rigid-like materials for which the number of solver iterations would significantly grow.

As a rule of thumb, values of ξ below 0.1 result in rigid-like materials, values of ξ around 0.4 represent highly viscous, but melting fluids such as mud, and values of ξ above 0.7 should be used for medium viscous fluids that dissolve fast, like honey.

	vorticity damping	vorticity preservation	vorticity diffusion
viscosity iter.	20	14	12
vorticity iter.	-	-	4

TABLE 5
Solver iterations for the scenario in the bottom row of Fig. 8.

ξ	0.1	0.5	0.9
viscosity iter.	28	16	10
vorticity iter.	8	7	7

TABLE 6
Solver iterations for the scenario in Fig. 9.

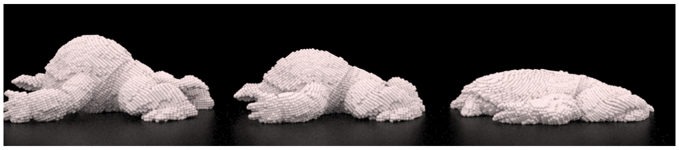


Fig. 9. Different values of ξ. From left to right: $\xi = 0.1$, $\xi = 0.5$, $\xi = 0.9$.

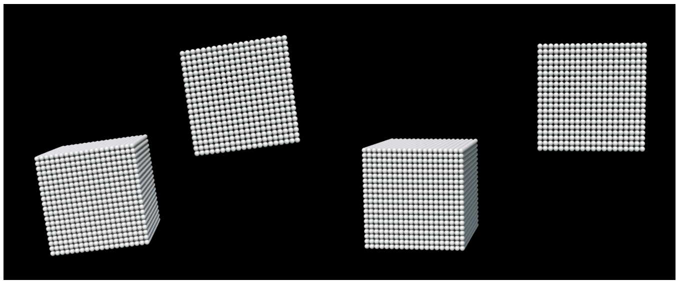


Fig. 10. Colliding cubes. Left: Our approach. Right: Takahashi's approach [17].

6.11 Comparison with Takahashi [17], cubes

We have outlined in Sec. 5.1 that Takahashi's formulation [17] and Peer's formulation [22] converge towards each other for extremely viscous fluids. This can be an indicator that [17] has problems with vorticity diffusion. The two experiments in Fig. 10 show two colliding fluids. While the fluids start rotating after the collision using our approach, the fluids do not rotate with [17]. Please refer to the accompanying video to assess the differences in the dynamics. Each scenario consists of 16k particles with a particle spacing of 50mm. Our approach requires 13 vorticity iterations and 4 viscosity iterations for $\xi = 0$, while Takahashi's technique [17] requires 32 iterations for $\mu = 10^9$ and $\rho_0 = 1000$.

6.12 Comparison with Takahashi [17], buckling and coiling

Fig. 11 shows that both approaches, Takahashi's [17] and ours, can capture coiling effects. Takahashi's method seems to produce "nicer" patterns for lower viscosities, while the differences to our method are less obvious for higher viscosities. As already discussed in Sec. 5.2, our method is faster. Further, Takahashi's scheme seems to introduce an artificial melting effect at the boundary which does not occur with our approach. This might be due to the fact that pressure and viscosity solver do not interfere in our approach. Each scenario in Fig. 11 consists of 5k particles with a spacing of 50mm. The top row of Fig. 11 shows Takahashi's approach with $\mu = 10^4$ and $\rho_0 = 1000$, while our approach runs with $\xi = 0.5$. Takahashi's approach requires 20 iterations, while our approach requires 3 vorticity iterations and 8 viscosity iterations. The bottom row of Fig. 11 shows Takahashi's approach with $\mu = 10^5$ and $\rho_0 = 1000$, while our approach runs with $\xi = 0.1$. Takahashi's approach requires 38 iterations, while our approach requires 3 vorticity iterations and 17 viscosity iterations.

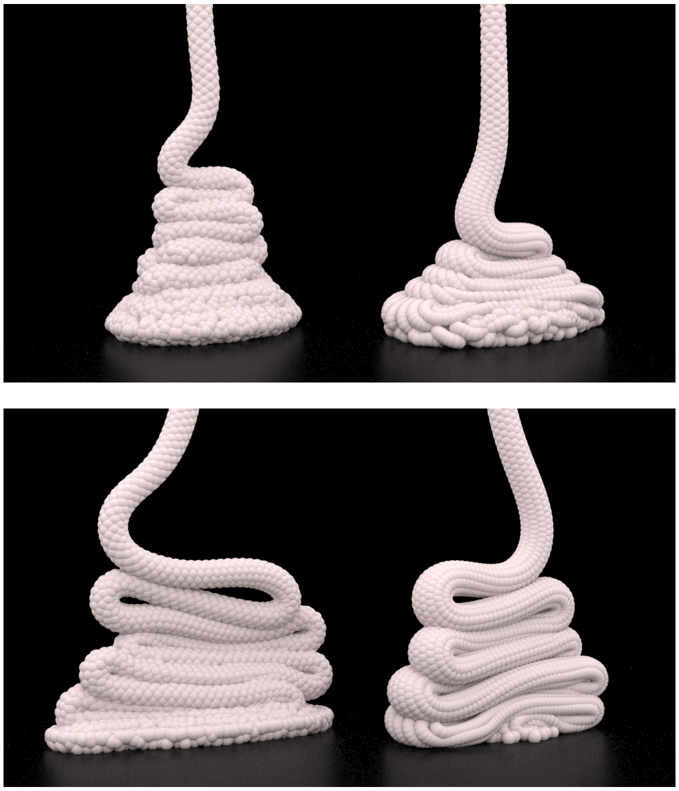


Fig. 11. Buckling and coiling for fluids with varying viscosity. Top: low viscosity. Bottom: high viscosity. Left: Takahashi's approach [17]. Right: Our approach.

7 Conclusion

We have presented a novel variant of prescribed velocity gradients for highly viscous SPH fluids. In contrast to alternative variants that damp or preserve vorticity, the proposed variant diffuses vorticity in a physically motivated way. Therefore, a diffusion process is employed to compute the target vorticity. This target vorticity is encoded in the target velocity gradient and the final velocity field is reconstructed accordingly. We have performed comparisons of three different variants for the vorticity processing in order to illustrate artifacts in case of damping and preservation and to show the plausible handling of rotations in case of the proposed vorticity diffusion. Additional scenarios have shown the flexibility of the approach. It can be used in combination with one- or two-way coupled solids and for interacting multiphase fluids.

References

[1] J. Monaghan, "Smoothed particle hydrodynamics," *Annual review of astronomy and astrophysics*, vol. 30, pp. 543–574, 1992.

[2] M. Ihmsen, J. Orthmann, B. Solenthaler, A. Kolb, and M. Teschner, "SPH fluids in computer graphics," in *Eurographics 2014 - State of the Art Reports*, S. Lefebvre and M. Spagnuolo, Eds. The Eurographics Association, 2014.

[3] J. Cornelis, M. Ihmsen, A. Peer, and M. Teschner, "IISPH-FLIP for incompressible fluids," in *Computer Graphics Forum*, vol. 33, no. 2. Wiley Online Library, 2014, pp. 255–262.

[4] B. Solenthaler and R. Pajarola, "Predictive-corrective incompressible SPH," in *ACM transactions on graphics (TOG)*, vol. 28, no. 3. ACM, 2009, p. 40.

[5] M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, and M. Teschner, "Implicit incompressible SPH," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 426–435, Mar. 2014.

[6] J. Bender and D. Koschier, "Divergence-free SPH for incompressible and viscous fluids," *IEEE Transactions on Visualization and Computer Graphics*.

[7] M. Ihmsen, N. Akinci, M. Gissler, and M. Teschner, "Boundary handling and adaptive time-stepping for PCISPH," in *Proceedings VRIPHYS*. VRIPHYS, 2010, pp. 79–88.

[8] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, and M. Teschner, "Versatile rigid-fluid coupling for incompressible SPH," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 62, 2012.

[9] H. Schechter and R. Bridson, "Ghost SPH for animating water," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 61, 2012.

[10] B. Solenthaler and R. Pajarola, "Density contrast SPH interfaces," in *Proceedings of the 2008 ACM SIGGRAPH/Eurographics symposium on computer animation*. Eurographics Association, 2008, pp. 211–218.

[11] B. Ren, C. Li, X. Yan, M. C. Lin, J. Bonet, and S.-M. Hu, "Multiple-fluid SPH simulation using a mixture model," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 5, p. 171, 2014.

[12] N. Akinci, G. Akinci, and M. Teschner, "Versatile surface tension and adhesion for SPH fluids," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 182, 2013.

[13] X. He, H. Wang, F. Zhang, H. Wang, G. Wang, and K. Zhou, "Robust simulation of sparsely sampled thin features in SPH-based free surface flows," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 1, p. 7, 2014.

[14] J. Monaghan, "On the problem of penetration in particle methods," *Journal of Computational physics*, vol. 82, no. 1, pp. 1–15, 1989.

[15] J. P. Morris, P. J. Fox, and Y. Zhu, "Modeling low reynolds number incompressible flows using SPH," *Journal of computational physics*, vol. 136, no. 1, pp. 214–226, 1997.

[16] M. Müller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003, pp. 154–159.

[17] T. Takahashi, Y. Dobashi, I. Fujishiro, T. Nishita, and M. C. Lin, "Implicit formulation for SPH-based viscous fluids," in *Computer Graphics Forum*, vol. 34, no. 2. Wiley Online Library, 2015, pp. 493–502.

[18] C. Batty and R. Bridson, "Accurate viscous free surfaces for buckling, coiling, and rotating liquids," in *Proceedings of the 2008 ACM SIGGRAPH/Eurographics symposium on computer animation*. Eurographics Association, 2008, pp. 219–228.

[19] J. Stam, "Stable fluids," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 121–128.

[20] M. Carlson, P. J. Mucha, R. B. Van Horn III, and G. Turk, "Melting and flowing," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 2002, pp. 167–174.

[21] F. Losasso, T. Shinar, A. Selle, and R. Fedkiw, "Multiple interacting liquids," in *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 812–819.

[22] A. Peer, M. Ihmsen, J. Cornelis, and M. Teschner, "An implicit viscosity formulation for SPH fluids," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 114, 2015.

[23] J. Steinhoff and D. Underhill, "Modification of the Euler equations for vorticity confinement: application to the computation of interacting vortex rings," *Physics of Fluids (1994-present)*, vol. 6, no. 8, pp. 2738–2744, 1994.

[24] A. Selle, N. Rasmussen, and R. Fedkiw, "A vortex particle method for smoke, water and explosions," in *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 910–914.

[25] X. Zhang, R. Bridson, and C. Greif, "Restoring the missing vorticity in advection-projection fluid solvers," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 52, 2015.

[26] M. Macklin and M. Müller, "Position based fluids," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 104, 2013.

[27] C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin, "The affine particle-in-cell method," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 51:1–51:10, 2015.

[28] N. Rasmussen, D. Enright, D. Nguyen, S. Marino, N. Sumner, W. Geiger, S. Hoon, and R. Fedkiw, "Directable photorealistic liquids," in *Proceedings of the 2004 ACM SIGGRAPH/Eurographics*

symposium on Computer animation. Eurographics Association, 2004, pp. 193–202.

[29] D. Ram, T. Gast, C. Jiang, C. Schroeder, A. Stomakhin, J. Teran, and P. Kavehpour, "A material point method for viscoelastic fluids, foams and sponges," in *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2015, pp. 157–163.

[30] B. Zhu, M. Lee, E. Quigley, and R. Fedkiw, "Codimensional non-Newtonian fluids," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 115, 2015.

[31] J. Monaghan, "Smoothed particle hydrodynamics," *Reports on progress in physics*, vol. 68, no. 8, p. 1703, 2005.

[32] T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of hermite data," *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, pp. 339–346, 2002.

[33] M. Ihmsen, N. Akinci, M. Becker, and M. Teschner, "A parallel SPH implementation on multi-core CPUs," in *Computer Graphics Forum*, vol. 30, no. 1. Wiley Online Library, 2011, pp. 99–112.

Andreas Peer received his MSc degree in computer science from the University of Freiburg in 2013. He is a research assistant and a PhD candidate in the computer graphics group at the University of Freiburg. His research interests include physically based animations, with a particular focus on fluid simulations.

Matthias Teschner received the PhD degree in electrical engineering from the University of Erlangen-Nuremberg in 2000. He is professor of computer science and head of the computer graphics group at the University of Freiburg. From 2001 to 2004, he was research associate at Stanford University and at the ETH Zurich. His research interests include physically based simulation, computer animation, rendering, and computational geometry with applications in robotics, medical simulation and entertainment technology. He regularly serves on program committees of major graphics conferences including Eurographics, Pacific Graphics, IEEE Vis, and ACM SIGGRAPH / Eurographics SCA. He serves as an associate editor for Computers & Graphics.