Consistent Shepard Interpolation for SPH-Based Fluid Animation

STEFAN REINHARDT, University of Stuttgart, Hochschule der Medien, Germany
TIM KRAKE, University of Stuttgart, Hochschule der Medien, Germany
BERNHARD EBERHARDT, Hochschule der Medien, Germany
DANIEL WEISKOPF, University of Stuttgart, Germany

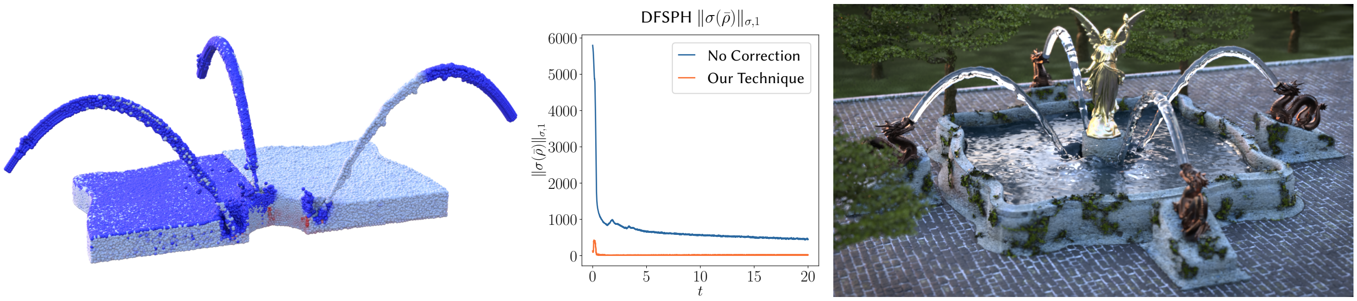

Fig. 1. In SPH-based fluid animation, the continuous fluid domain is sampled with particles. This discretization procedure induces errors resulting in a noisy density field (left part of the particle rendering shown on the left). We propose a correction method to reduce this noise (right part of the particle rendering). The observations are quantified using a suitable norm (center). A production rendering is shown on the right.

We present a novel technique to correct errors introduced by the discretization of a fluid body when animating it with smoothed particle hydrodynamics (SPH). Our approach is based on the Shepard correction, which reduces the interpolation errors from irregularly spaced data. With Shepard correction, the smoothing kernel function is normalized using the weighted sum of the kernel function values in the neighborhood. To compute the correction factor, densities of neighboring particles are needed, which themselves are computed with the uncorrected kernel. This results in an inconsistent formulation and an error-prone correction of the kernel. As a consequence, the density computation may be inaccurate, thus the pressure forces are erroneous and may cause instabilities in the simulation process. We present a consistent formulation by using the corrected densities to compute the exact kernel correction factor and, thereby, increase the accuracy of the simulation. Employing our method, a smooth density distribution is achieved, i.e., the noise in the density field is reduced by orders of magnitude. To show that our method is independent of the SPH variant, we evaluate our technique on weakly compressible SPH and on divergence-free SPH. Incorporating the corrected density into the correction process, the problem cannot be stated explicitly anymore. We propose an efficient and easy-to-implement algorithm to solve the implicit problem by applying the power method. Additionally, we demonstrate how our model can be applied to improve the density distribution on rigid bodies when using a well-known rigid-fluid coupling approach.

CCS Concepts: • **Computing methodologies → Modeling methodologies**; **Physical simulation**; *Computer graphics*.

Additional Key Words and Phrases: Smoothed Particle Hydrodynamics, Physically Based Animation

ACM Reference Format:
Stefan Reinhardt, Tim Krake, Bernhard Eberhardt, and Daniel Weiskopf. 2019. Consistent Shepard Interpolation for SPH-Based Fluid Animation. *ACM Trans. Graph.* 38, 6, Article 189 (November 2019), 11 pages. https://doi.org/10.1145/3355089.3356503

1 INTRODUCTION

Smoothed particle hydrodynamics (SPH) has evolved into a well-established method for computer animation. It is a Lagrangian technique to animate fluid flow with free surfaces and complex interactions. The fluid quantities are evaluated at given particle positions by using neighboring particles only.

The demand for visual quality and degree of realism is constantly increasing for computer-generated animations. Higher quality is not only achieved by increasing the particle numbers but also by improving the accuracy of the underlying models to simulate the physical behavior as best as possible. Due to the Lagrangian nature of SPH, the incompressibility constraint is typically violated as the fluid density is a function of time. This leads to oscillations of the fluid volume and affects the simulation quality [Ihmsen et al. 2014b]. Modern SPH variants overcome this issue and enforce the incompressibility constraint by adjusting the pressure forces accordingly.

The fluid quantities, such as the density, are not only a function of time but also of space. Due to the spatial discretization of the

Authors' addresses: Stefan Reinhardt, University of Stuttgart, Hochschule der Medien, Nobelstr. 10, 70569, Stuttgart, Germany, stefan.reinhardt@visus.uni-stuttgart.de; Tim Krake, University of Stuttgart, Hochschule der Medien, Nobelstr. 10, 70569, Stuttgart, Germany, tim.krake@visus.uni-stuttgart.de; Bernhard Eberhardt, Hochschule der Medien, Nobelstr. 10, 70569, Stuttgart, Germany, eberhardt@hdm-stuttgart.de; Daniel Weiskopf, University of Stuttgart, Allmandring 19, 70569, Stuttgart, Germany, weiskopf@visus.uni-stuttgart.de.

0730-0301/2019/11-ART189 $15.00
https://doi.org/10.1145/3355089.3356503

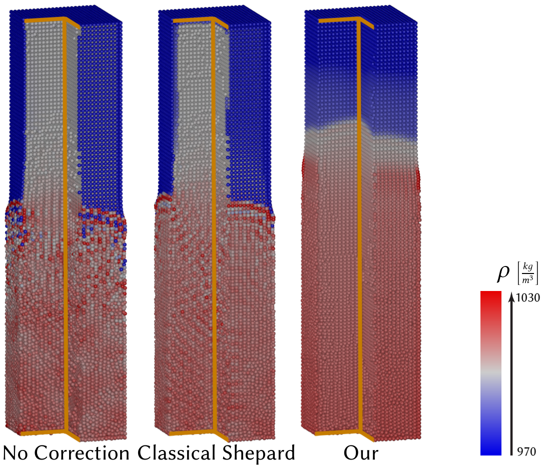

Fig. 2. Snapshot of the fluid pillar scenario consisting of 60 k particles simulated with WCSPH. From left to right: simulation without kernel correction, with classical Shepard correction, and with our method. Particles are colored with respect to density. We sliced out the front left quarter of the tower to show the inside of the fluid. With our method, we obtain a significantly smoother density field compared to simulations conducted without kernel correction or with classical Shepard correction.

fluid body, inevitable errors are introduced as well. The Shepard correction [Shepard 1968] (also referred to as Shepard kernel) is often used to reduce this kind of error when animating the fluid with SPH. To this end, the smoothing kernel is normalized when computing the fluid quantities. This approach has one drawback as the correction factors depend on the volume represented per particle. Since the particles' mass is constant and the density changes over time and space, the represented volume changes and, more importantly, depends on the density itself. Shepard correction employs the uncorrected particle volumes to scale the kernel and, therefore, results in an inconsistent and error-prone correction scheme.

We resolve this issue and present a consistent method to correct the kernel. We propose an iterative technique that includes the corrected densities to compute the Shepard kernel. As Figure 2 demonstrates, we achieve a smoother density distribution throughout the fluid. The main technical contributions of this paper are:

- A consistent and stable method to compute the kernel correction.
- A linear formulation of the problem solved iteratively by using the power method. Furthermore, we prove that our algorithm is unconditionally stable.
- A kernel gradient correction scheme that accounts for discretization errors in the kernel gradient approximation.
- To model rigid boundaries, we use the rigid-fluid coupling model of Akinci et al. [2012]. We adapt this model and adjust the local number densities with corrected kernels.

Employing the power method is very efficient, e.g., for the fountain scenario (Figure 1), we need on average only 2.1 iterations to converge. Even in highly dynamic scenes with very large time steps, our solver needs on average less than five iterations to converge. Therefore, the introduced overhead can be neglected. We evaluate our model on weakly compressible SPH (WCSPH) [Becker and Teschner 2007] and divergence-free SPH (DFSPH) [Bender and Koschier 2017]. With a suitable norm, we show that the density distribution is smoother with our method compared to Shepard correction and a simulation with uncorrected kernel.

2 PREVIOUS WORK

Since Desbrun and Gascuel [1996] introduced SPH to the field of computer graphics, many extensions to its original formulation have been presented. Several authors examine the density preservation issue, either based on the equation of state [Becker and Teschner 2007; Bender and Koschier 2017; Solenthaler and Pajarola 2009] or pressure projection schemes [Cummins and Rudman 1999; Ellero et al. 2007; Ihmsen et al. 2014a]. Schechter and Bridson [2012] address the sampling deficiency near solid and air boundaries by adding a narrow layer of ghost particles. We propose a method that reduces discretization errors all over the fluid body.

Shepard [1968] presents an operational solution to reduce discretization errors and to produce a continuous surface from the interpolation of irregularly spaced data using weighted averages. This kernel normalization, often referred to as Shepard kernel, is applied by many authors to accurately preserve the discontinuity at interfaces (e.g., by Johnson and Beissel [1996] or Colagrossi [2001]), such as liquid-liquid or liquid-air interfaces. Grenier et al. [2009] additionally use a gradient normalization based on the Shepard kernel. The Shepard kernel is also referred to zeroth-order correction (or constant completeness) [Belytschko et al. 1998], i.e., constant functions are exactly represented by the approximation. We propose a method based on the Shepard kernel and obtain constant completeness but also resolve inconsistencies of the original formulation.

When simulating a fluid, methods that satisfy higher-order completeness conditions are often applied (a completeness condition to the order k is satisfied if any polynomial function to the degree k is represented exactly) [Belytschko et al. 1998]. An overview of completeness methods for SPH can be found in the work of Belytschko et al. [1998] and Vaughan et al. [2008]. They examine different approximation techniques that restore various levels of completeness. Krongauz and Belytschko [1996] achieve linear (and higher-level) completeness by using a moving least squares approximation. This approach is not a kernel correction in a classical sense but computes the approximation directly, achieving a specified level of completeness. Liu et al. [1997] use the moving least squares interpolation scheme to correct the kernel function, the so-called moving least-square reproducing kernel. Similar to the approach by Krongauz and Belytschko, a polynomial of arbitrary degree is exactly represented. The moving least-square reproducing kernel approach is applicable to any meshless approximation for Galerkin procedures, in particular for SPH.

Johnson and Beissel [1996] present an algorithm that uses normalized smoothing kernel functions for SPH. They adjust the smoothing kernel so that constant normal strain rates are represented exactly. Even though their approach does not ensure linear completeness, their results show improved accuracy, especially at free boundaries. Bonet and Lok [1999] and Bonet et al. [2004] present a variational formulation of SPH that ensures linear completeness and works for both fluids and solids. Their approach reduces density noise remarkably, as discussed by Vaughan et al. [2008]. Similar to our model, higher-order kernel corrections reduce the noise in the density field. Nonetheless, the mentioned methods neglected that the volume represented by a particle changes over space and time. Since the mass is usually set to a constant value and the density changes, the volume has to change and, therefore, needs to be included in the kernel correction. We resolve this issue and propose an approach that accounts for this fact.

Monaghan [1988] proposes a skew-symmetric form of the kernel gradient. This symmetrization results in constant completeness of the gradient [Belytschko et al. 1998]. Morris [1996] proposes the same formulation to compute the velocity changes due to pressure forces. This formulation is stable and independent of background pressure. Note that it does not conserve momentum exactly [Monaghan 1992]. Therefore, alternative variants are often used (see, e.g., Morris et al. [1997] or Monaghan [1992]). Randles and Libersky [1996] propose a kernel gradient correction scheme to model generalized boundary conditions. It can be understood as a generalization of the method by Johnson and Beissel [1996] and ensures that the gradient of any linear tensor field is approximated exactly. Bonet et al. [2004] also use modified kernel gradients and achieve linear completeness. Nonetheless, these modifications lead to the problem that the gradient is no longer antisymmetric and, hence, linear and angular momentum are not necessarily preserved.

All the mentioned methods alter the gradient of the smoothing kernel itself to obtain constant completeness. We propose a method for constant completeness that does not alter the kernel gradient but implicitly adds a ghost particle to compensate the discretization errors. A similar idea is used by Ganzenmüller [2015]. Instead of adding an integration point (particle) to correct the kernel gradient, he computes a correction force to suppress zero-energy modes (the deformation of an element without change of its potential energy). This is similar to our concept in the sense that we virtually place a particle and use the resulting force that would occur. However, our approach has a different goal: While Ganzenmüller suppresses zero-energy modes, we ensure that the kernel gradient sums up to zero. Note that our method does not replace the mentioned kernel and gradient correction schemes but can be combined with them.

3 BACKGROUND

Before describing our method in detail, it is necessary to recapitulate the fundamentals of SPH and the classical Shepard correction. Additionally, we outline the SPH variants into which we integrated our method to evaluate our technique, namely WCSPH and DFSPH.

3.1 Basics of SPH

The motion of a fluid can be described by the equation of momentum of the Navier-Stokes equations:

$$\frac{D\mathbf{v}}{Dt} = \frac{\mathbf{F}^b}{m} - \frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{v}, \tag{1}$$

where $\mathbf{v}$ denotes the fluid velocity, ρ its density, p the pressure, ν the kinematic viscosity, and $\mathbf{F}^b$ the body forces (e.g., gravity $\mathbf{g}$). SPH evaluates the fluid properties $A(\mathbf{x}(t))$ at specific locations $\mathbf{x}(t)$ at time t (for simplicity, we will skip the variable t in the following unless it is explicitly needed). The quantities $A(\mathbf{x})$ are smoothed over the domain $\Omega(\mathbf{x}, h) = \{\mathbf{y}\,; \|\mathbf{x}-\mathbf{y}\| \le h\}$, which is a compact area around position $\mathbf{x}$. To this end, a smoothing kernel function $W(r, h)$ with compact support h is employed. The fluid quantities are then approximated via

$$A(\mathbf{x}) = \int_{\Omega(\mathbf{x},h)} A(\mathbf{x}')W(\|\mathbf{x}-\mathbf{x}'\|, h)\, dV(\mathbf{x}'), \tag{2}$$

where $dV(\mathbf{x}')$ is the differential volume element at $\mathbf{x}'$. Besides the compact support, other requirements are demanded on W to be suitable as a kernel function. An important one is that W must be normalized [Monaghan 1988], i.e.,

$$\int_{\Omega(\mathbf{x},h)} W(\|\mathbf{x}-\mathbf{x}'\|, h)\, dV(\mathbf{x}') = 1. \tag{3}$$

This ensures that the constant function $A(\mathbf{x}) = 1$ is exactly reproduced. The gradient ∇A of the fluid quantity A is evaluated via

$$\nabla A(\mathbf{x}) = \int_{\Omega(\mathbf{x},h)} A(\mathbf{x}')\nabla W(\|\mathbf{x}-\mathbf{x}'\|, h)\, dV(\mathbf{x}'). \tag{4}$$

Equation 4 lets us specify another desired property of the smoothing kernel W: To exactly reproduce the derivative of a constant function, such as $A(\mathbf{x}) = 1$, the integral of the kernel W over the smoothing domain Ω must be zero:

$$\mathbf{0} = \nabla 1 = \int_{\Omega(\mathbf{x},h)} \nabla W(\|\mathbf{x}-\mathbf{x}'\|, h)\, dV(\mathbf{x}'). \tag{5}$$

To animate the fluid, the equation of momentum (Equation 1) needs to be discretized over space and time. To this end, the fluid domain is sampled with n particles at discrete positions $\mathbf{x}_i$ with $i \in N = \{1, \ldots, n\}$ and the fluid quantities $A(\mathbf{x}_i)$ are evaluated at $\mathbf{x}_i$. Subsequently, particle accelerations are computed by Equation 1. To evaluate the fluid quantities with SPH, the formulation in Equation 2 is discretized, approximating the integral. Therefore, a specific volume is linked with each particle i, obtaining

$$A(\mathbf{x}_i) = \sum_{j\in N_i} \frac{m(\mathbf{x}_j)}{\rho(\mathbf{x}_j)} A(\mathbf{x}_j)W(\|\mathbf{x}_i - \mathbf{x}_j\|, h), \tag{6}$$

where $\frac{m(\mathbf{x}_j)}{\rho(\mathbf{x}_j)}$ describes the volume represented by particle j and $N_i \subset N$ the set of the discrete evaluation positions of the smoothing domain $\Omega(\mathbf{x}_i, h)$. For simplicity, we will write $A_i = A(\mathbf{x}_i)$ and $W_{ij} = W(\|\mathbf{x}_i - \mathbf{x}_j\|, h)$ in the following. Similar to the computation of attribute A_i, the integral in Equation 4 needs to be approximated when computing the gradient of the quantity of the fluid ∇A_i. Equation 4 then reads as

$$\nabla A_i = \sum_{j\in N_i} \frac{m_j}{\rho_j} A_j \nabla W_{ij}. \tag{7}$$

This formulation has some drawbacks, especially when computing the pressure gradient ∇p_i to determine the pressure forces. Modern SPH solvers use an antisymmetric reformulation of ∇p_i to conserve momentum [Morris et al. 1997]. To this end, Monaghan [2005] rewrites the pressure gradient via

$$\frac{\nabla p}{\rho} = \nabla\left(\frac{p}{\rho}\right) + \frac{p}{\rho^2}\nabla\rho. \tag{8}$$

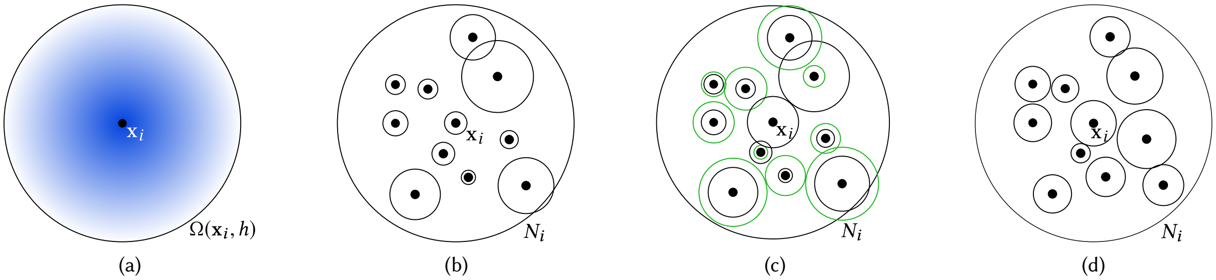

Fig. 3. Illustration of the fluid body in the neighborhood of particle i. In the continuous case, the kernel normalization property is satisfied (a), whereas this property gets lost in the discretization process (b). Classical Shepard correction (c) addresses this problem by adjusting the volume of particle i. To compute the correction factor the volumes of the neighboring particles are needed. However, classical Shepard correction neglects that they undergo a correction (indicated by the green circles in (c)) and uses the uncorrected volumes (black circles) to compute the correction factors. Consequently, the correction factor and, therefore, the volume is calculated inconsistently. We propose a consistent formulation and compute the correction factors simultaneously (d). This results in a smoother density distribution and, hence, more uniform volumes.

This results in an antisymmetric pressure gradient formulation and exactly conserves linear and angular momentum [Monaghan 2005; Morris et al. 1997]. The acceleration of particle i induced through pressure can then be computed via

$$\frac{1}{\rho_i}\nabla p_i = \sum_{j \in N_i} m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right)\nabla W_{ij}. \tag{9}$$

To obtain the pressure itself, a pressure Poisson equation or an equation of state (EOS) can be employed. WCSPH uses the EOS to compute the pressure:

$$p_i = k\frac{\rho_0}{\gamma}\left(\left(\frac{\rho_i}{\rho_0}\right)^{\gamma} - 1\right). \tag{10}$$

Commonly, $\gamma = 7$ is used and, to enforce incompressibility, the stiffness factor needs to be set to $k = c_s^2$, where c_s is the speed of sound. In practice, k is a stiffness constant that scales the pressure [Ihmsen et al. 2014b]. DFSPH employs the EOS to derive the stiffness constants for the constant-density and the divergence-free solver. It uses the linear form of Equation 10 (i.e., $\gamma = 1$).

3.2 Shepard Correction

The Shepard [1968] correction of the smoothing kernel W addresses errors introduced by the SPH discretization process due to irregularly distributed particles inside the fluid domain. Especially near open boundaries (such as fluid-air interfaces), the computation of the fluid quantity A_i is error-prone due to the lack of neighboring particles. The following correction factor c_i^{sh} is computed to account for this fact:

$$c_i^{\text{sh}} = \frac{1}{\sum_{k \in N_i} \frac{m_k}{\rho_k} W_{ik}}. \tag{11}$$

The factor c_i^{sh} can then be used to correct the smoothing kernel W_{ij} via

$$\widetilde{W}_{ij} = c_i^{\text{sh}} W_{ij} = \frac{W_{ij}}{\sum_{k \in N_i} \frac{m_k}{\rho_k} W_{ik}}. \tag{12}$$

This correction shall ensure that the property described in Equation 3 still holds for the discrete case, i.e., the kernel is normalized (note that this only holds if uncorrected densities ρ_k are considered). The fluid attributes are then computed with the corrected kernel $\widetilde{A}_i = \sum_{j \in N_i} \frac{m_j}{\rho_j} A_j \widetilde{W}_{ij}$. The classical Shepard correction is illustrated in Figure 3 (c).

4 CORRECTED CORRECTION

In this section, we present our novel approach to compute consistent correction factors c_i by considering the corrected densities. As pointed out earlier, the densities are fluid quantities themselves and need to undergo the correction step as well. They are computed using Equation 6, i.e., $\rho_i = \sum_{j \in N_i} m_j W_{ij}$. In essence, this means that with the classical Shepard correction, a different smoothing kernel function (the kernel W) is used to compute the correction factors than for the computation of the fluid quantities (the kernel $\widetilde{W}$). This is an inconsistent use of kernel functions and introduces new errors. Instead of smoothing the density field, the Shepard correction may distort it even more compared to simulating without a kernel correction. We resolve this inconsistency and use the Shepard kernel $\widetilde{W}$ to compute the correction factors c_i.

4.1 Kernel Correction

It is worth to consider another view on the problem: On account of the fluid discretization, the particle i represents a certain amount $V_i = \frac{m_i}{\rho_i}$ of the total fluid volume V, as illustrated in Figure 3 (b). Using the classical Shepard correction, the density ρ_i and, therefore, the volume element V_i is corrected (Figure 3 (c)). This correction process neglects that the volume elements V_j of the neighboring particles j change as well. We account for this fact and include the corrected neighboring volume elements $V_j^{\text{c}} = \frac{m_j}{c_j \rho_j}$ (Figure 3 (d)). Equation 11 then becomes

$$c_i = \frac{1}{\sum_{j \in N_i} \frac{m_j}{\sum_{k \in N_j} m_k c_j W_{jk}} W_{ij}} = \frac{1}{\sum_{j \in N_i} \frac{m_j}{c_j \rho_j} W_{ij}}. \tag{13}$$

We cannot compute c_i explicitly anymore as the correction factors depend on each other. This problem can, however, be solved iteratively, e.g., using fixed-point iteration. By $c = (c_1, \ldots, c_n)$ we denote the vector of the correction factors c_i, where n is the total number of particles. Let $F(c) = (f_1(c), ..., f_n(c))$ be a vector of functions f_i defined as

$$f_i(c) = \frac{1}{\sum_{j\in N_i} \frac{m_j}{c_j \rho_j} W_{ij}}. \tag{14}$$

To solve Equation 13 we search for a c that satisfies $F(c) = c$. To this end, we need to repeat the following steps:

(1) For all particles i, set $c^0 = (1, ..., 1)$.
(2) Repeat until converged: $c^{r+1} = F(c^r)$.

Note that, if we perform only one iteration in step (2), we end up with the classic Shepard correction. Therefore, our method could be interpreted as a generalization of the Shepard correction.

Employing a fixed-point iteration to solve Equation 13 has some drawbacks though. First, convergence of the method is not ensured and, second, we cannot guarantee temporal coherence of the solution. This induces discontinuities in the density field and, therefore, causes instabilities in the simulation. To reduce the likelihood of incoherent solutions, we can initialize c with the previous solution, i.e., $c^0(t + \Delta t) = c(t)$, but we can still end up with instabilities, as we may have obtained a repulsive fixed point. Besides the instability issues, the speed of convergence is typically very slow for fixed-point iterations.

We account for these issues and present an unconditionally stable and fast converging algorithm employing the power method. To this end, we use an equivalent reformulation of this problem. We substitute c_i with $\tilde{c}_i = \frac{1}{c_i}$:

$$\tilde{c}_i = \frac{1}{c_i} = \sum_{j\in N_i} \tilde{c}_j \underbrace{\frac{m_j}{\rho_j} W_{ij}}_{a_{ij}} . \tag{15}$$

If we use Equation 15 instead of Equation 14, we obtain a fixed-point problem $A\tilde{c} = \tilde{c}$ with the linear function A. Solving this is equivalent to finding an eigenvector $\tilde{c}$ to the eigenvalue $\lambda = 1$ of the matrix $A = (a_{ij})_{(i,j)\in N\times N}$. For this purpose, we examine A in detail. First, we need to ensure that $\lambda = 1$ exists. Fortunately, the column sum of A equals one, i.e., $\sum_{i\in N} \frac{m_j}{\rho_j} W_{ij} = 1$ and the entries a_{ij} of A satisfy the condition $0 \leq a_{ij} \leq 1$. This implies that A is a column (or left) stochastic matrix and we can conclude that the eigenvalue $\lambda = 1$ exists. In addition, for all other eigenvalues λ, it holds true that $|\lambda| < 1$ (see Lemma A.3). With these insights into matrix A, we are able to use the power method [Mises and Pollaczek-Geiringer 1929] to find a solution vector $\tilde{c}$ by considering the recurrence relation

$$\tilde{c}^{r+1} = \frac{A\tilde{c}^r}{\|A\tilde{c}^r\|}. \tag{16}$$

Applying this method, we can ensure that our algorithm converges and returns a solution for $A\tilde{c} = \tilde{c}$. Furthermore, we can guarantee a unique solution if we have one connected fluid body. By the term connected fluid body we understand that every particle is a direct or indirect neighbor of any other particle of the fluid (see Definition A.1). For more details and a mathematical derivation, we refer the reader to Appendix A. If splashes occur, we do not have a connected fluid body. For a non-fully connected fluid body, the matrix A is not irreducible and, hence, $\lambda = 1$ is not necessarily a simple eigenvalue. However, in our experiments, we obtained temporal coherence when initializing the algorithm with $\tilde{c}_0(t + \Delta t) = \tilde{c}(t)$.

Algorithm 1 Compute $\tilde{c}$ using the power method

```
for particle $i$ do
    Neighborhood search
    If first step then $\tilde{c}_i^0 = 1$
    $\rho_i = \sum_{j\in N_i} m_j W_{ij}$
end for
do
    Compute $\|A\tilde{c}^r\|$
    for particle $i$ do
        $\tilde{c}_i^{r+1} = \frac{1}{\|A\tilde{c}^r\|} \sum_{j\in N_i} \tilde{c}_j^r a_{ij}$
    end for
while $\frac{1}{n}\sum_{i\in N} |\tilde{c}_i^{r+1} - \tilde{c}_i^r| > \varepsilon_1$ or $\max_{i\in N} \{|\tilde{c}_i^{r+1} - \tilde{c}_i^r|\} > \varepsilon_2$
```

Algorithm 1 describes the outline of our method. At least one iteration is always performed. The algorithm stops if the average change rate of $\tilde{c}_i$ is smaller or equal than ε_1 and the maximum rate is smaller than ε_2. Since only direct neighbors are needed to compute $\tilde{c}_i^{r+1}$, the problem can be solved efficiently in a matrix-free way. Having computed the kernel correction factors, we correct the kernel via

$$\widetilde{W}_{ij} = \frac{W_{ij}}{\tilde{c}_i} \tag{17}$$

and then perform the SPH simulation step.

4.2 Kernel Gradient Correction

To determine the fluid quantities, the kernel gradient ∇W_{ij} is needed as well. Since the kernel corrections (Section 4.1) alter the kernel function, the computation of the gradient ∇W_{ij} has to be adjusted. To obtain a consistent formulation for the kernel gradient, we take Equation 17 into account and compute the gradient $\nabla \widetilde{W}_{ij}$ via

$$\nabla \widetilde{W}_{ij} = \nabla \left(\frac{W_{ij}}{\tilde{c}_i} \right) = \frac{\nabla W_{ij} - W_{ij} \frac{\nabla \tilde{c}_i}{\tilde{c}_i}}{\tilde{c}_i}, \tag{18}$$

where $\nabla \tilde{c}_i$ is obtained from Equation 7 but considering the corrected densities:

$$\nabla \tilde{c}_i = \sum_{j\in N_i} \frac{m_j}{\frac{1}{\tilde{c}_j}\rho_j} \nabla W_{ij}. \tag{19}$$

Even though Equation 18 provides a consistent formulation of the gradient and, hence, increases the accuracy, it does not necessarily satisfy the reproducing condition stated in Equation 5. In other words, to exactly reproduce the gradient of constant functions we need to ensure that $\sum_{j\in N_i} \frac{m_j}{c_i \rho_j} \nabla \widetilde{W}_{ij} = 0$. In general, this condition is violated. By

$$\xi_{s_i} = \sum_{j\in N_i} \frac{m_j}{c_i \rho_j} \nabla \widetilde{W}_{ij}, \tag{20}$$

we denote the error of the kernel gradient sum. If we subtract ξ_{s_i} from the right-hand side of Equation 20, it sums up to zero: $0 = \sum_{j\in N_i} \frac{m_j}{c_i \rho_j} \nabla \widetilde{W}_{ij} - \xi_{s_i}$. One element in the sum corresponds to exactly one neighboring particle. We think of ξ_{s_i} as an additional addend to the sum, considering the value ξ_{s_i} to correspond to a ghost particle s_i at an unknown location $\mathbf{x}_{s_i}$. In other words, we add a ghost particle s_i into the neighborhood of particle i to correct the error. Therefore, ξ_{s_i} can be expressed in the form

$$\xi_{s_i} = \frac{m_{s_i}}{c_{s_i}\rho_{s_i}} \nabla \widetilde{W}_{is_i}. \tag{21}$$

To place the ghost particle s_i we need to know its location $\mathbf{x}_{s_i}$. To this end, we define the function $\boldsymbol{\kappa}_i(\mathbf{x}) = \frac{m(\mathbf{x})}{c(\mathbf{x})\rho(\mathbf{x})}\nabla\widetilde{W}(\|\mathbf{x}_i - \mathbf{x}\|, h)$. The location $\mathbf{x}_{s_i}$ is then given by

$$\mathbf{x}_{s_i} = \boldsymbol{\kappa}_i^{-1}\left(-\xi_{s_i}\right). \tag{22}$$

Knowing the position $\mathbf{x}_{s_i}$ of our ghost particle, we can compute the fluid quantity A_{s_i} using standard SPH interpolation as stated in Equation 6 and correct the kernel gradient sum.

Although this would provide an exact solution, computing $\boldsymbol{\kappa}_i^{-1}$ is computationally expensive. Furthermore, it depends on the choice of kernel function, is not always well defined, or may even not exist[1]. Fortunately, we can compute A_{s_i} without explicit knowledge of $\mathbf{x}_{s_i}$. As mentioned, we aim to reproduce constant functions, which can be ensured if we approximate A_{s_i} with $A_{s_i} = A_i$. The corrected gradient ∇A_i of the fluid quantity A is then computed as:

$$\nabla A_i = \sum_{j\in N_i} \frac{m_j}{c_j\rho_j} A_j \nabla\widetilde{W}_{ij} - A_i \xi_{s_i}. \tag{23}$$

When animating the fluid using WCSPH, we use an antisymmetric kernel as stated in Equation 9. In this case, ξ_{s_i} becomes

$$\xi_{s_i} = \sum_{j\in N_i} m_j \left(\frac{1}{\rho_j^2} + \frac{1}{\rho_i^2}\right) \nabla\widetilde{W}_{ij}, \tag{24}$$

and the gradient of the fluid quantity A is then adjusted accordingly.

4.3 Rigid Boundary Correction

We employ the approach of Akinci et al. [2012] to model rigid boundary objects. With this model, the surface of the rigid object is sampled with particles that interact with the fluid. These boundary particles contribute to the density computation of the fluid particles. The density ρ_i of particle i then reads:

$$\rho_i = \sum_{j\in N_i^f} m_j \widetilde{W}_{ij} + \sum_{k\in N_i^b} \Psi_{b_k}(\rho_0) W_{ik}, \tag{25}$$

where $\Psi_{b_k}(\rho_0)$ is the so-called local number density, N_i^f is the set of fluid neighbors of particle i, and N_i^b the set of boundary neighbors. In essence, $\Psi_{b_k}(\rho_0)$ is a 'pseudo mass' that is computed based on the configuration of the neighborhood particles:

$$\Psi_{b_k}(\rho_0) = \frac{\rho_0}{\sum_{j\in N_k^b} W_{b_k j}}. \tag{26}$$

To maintain a consistent approach, we include the boundary particles into the computation of the kernel correction factors, i.e., we determine correction factors for rigid bodies as well and the kernel W_{ik} in Equations 25 and 26 is replaced by $\widetilde{W}_{ik}$.

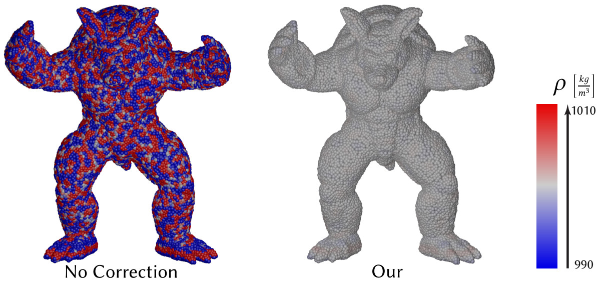

Fig. 4. Calculating the density on rigid boundaries results in a noisy density field (left). In contrast, a smooth density distribution is achieved (right) by applying our kernel correction to the rigid boundary objects as a preprocessing step to the simulation correcting the volumes accordingly.

This boundary model has the advantage that only one layer of boundary particles is needed, and the local number densities are computed only with the boundary neighbors, as stated in Equation 26. However, the discretization of the smoothing kernel function tends to be error prone, especially at free surfaces where only few neighboring particles are available. Therefore, we adjust the computation of $\Psi_{b_k}(\rho_0)$ and apply our approach to calculate correction factors, i.e., we alter Equation 26 accordingly using corrected kernels. This is done as a preprocessing step to the simulation and, hence, produces overhead only in the initialization but not during simulation. For our approach, we need to compute the density of a boundary particle b_i, via $\rho_{b_i} = \sum_{j\in N_i^b} \Psi_{b_j}(\rho_0)\widetilde{W}_{b_i j}$. Without our kernel correction, we obtain a noisy density field, as the example in Figure 4 (left) indicates. When adjusting the boundary volumes and computing the densities by employing the corrected kernel, we achieve a smooth density distribution over the complete rigid body (Figure 4, right) with barely any deviations from the reference density ρ_0.

5 IMPLEMENTATION

We implemented our approach in the open-source framework *SPlisHSPlasH*, which is also used by several other authors, e.g., Bender and Koschier [2017], Bender et al. [2018], or Weiler et al. [2018]. For all simulations, the cubic spline kernel function [Monaghan 2005] is employed. In most examples, we exclude surface tension, but if included we use a modification of the IIF model [Becker and Teschner 2007] as presented by Huber et al. [2015]. For WCSPH, the pressure is computed via Equation 10, where we set $k = 5.250$ and $\gamma = 7$, if not stated otherwise. We apply fixed time stepping for WCSPH with $\Delta t = 1$ ms and adaptive time stepping for DFSPH as presented by Monaghan [1992], where we restrict the maximum time step to $\Delta t \leq 5$ ms, except for measuring volume preservation. In this case, we use a fixed time step of $\Delta t = 2$ ms for DFSPH. The gradient correction scheme is always incorporated when using our method as well as with the classic Shepard kernel.

When applying our model to DFSPH, we observe a little roughness at free surfaces. We attribute this to the interplay of our approach and the constant-density solver. With our approach, densities are corrected, especially at free surfaces, and we observe a slight overestimation of the pressure forces in these areas. To overcome this issue, we scale the stiffness constant with c_i at free surfaces, which

[1] For example, when using the cubic spline kernel, ∇W is only injective on the interval $[0, h/3]$. In addition, it cannot be ensured that ξ_{s_i} is in the value range of function $\boldsymbol{\kappa}_i$.

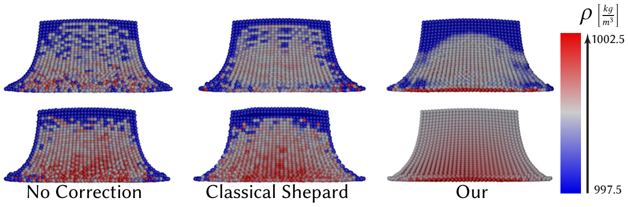

Fig. 5. The collapsing fluid block simulated with DFSPH (upper row) and WCSPH (lower row). From left to right: no kernel correction, classical Shepard correction, and our method. For both uncorrected and classical Shepard correction, we observe fine-grained noise in the density field. Using our technique, we achieve a completely smooth density field for WCSPH (lower right). Considering DFSPH, our method still significantly improves the smoothness of the density field.

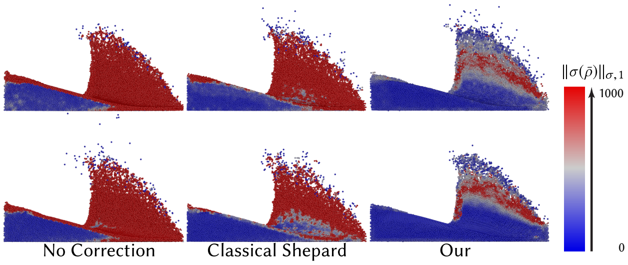

Fig. 6. A snapshot of the corner dam break scenario simulated with DFSPH (upper row) and WCSPH (lower row). The particles are color-coded with respect to the local density variance. On the left, the simulations without kernel corrections are shown. We observe high local density variances at the surface but also beneath. With classical Shepard correction (center), there are only slight improvements recognizable. With our method (right), there are only few regions with a high local density variance (where we would expect them) and the majority of the fluid body has a variance close to zero.

leads to a smooth appearance of the fluid surfaces. For a side-by-side comparison, we refer the reader to the supplemental material.

6 EVALUATION

To show the versatility of our method we consider scenarios of different characteristics and compare our technique to simulations conducted with uncorrected as well as with classical Shepard correction. One class of examples are almost steady fluids, such as the fluid pillar example depicted in Figure 2. To complement the evaluation, we explore the behavior of our method in highly dynamic scenes like the corner dam break shown in Figure 6. To reduce the influence of the boundary model we examine the density distribution of a collapsing fluid block standing on the ground (see Figure 5).

We evaluate our method considering two important aspects for the animation of a fluid: the smoothness of the density field and the volume the fluid occupies. A smooth density distribution is crucial for the stability because most fluid quantities depend on the density, especially the pressure, which usually produces the dominating forces in the system and, hence, is crucial for the stability of the simulation. With our method, we observe increased volume, which indicates that we obtain a more accurate simulation. Besides the density and volume discussion, we investigate convergence and stability characteristics of our technique.

6.1 Density Field Evaluation

We inspect the density distribution visually and substantiate the observations by measuring the smoothness of the density field locally and globally.

Figure 2 depicts a fluid pillar consisting of 60 k particles simulated with WCSPH. Without any kernel correction (left), we observe a quite noisy density distribution throughout the fluid. This is slightly improved by applying classical Shepard correction (center), but we still recognize large fluctuations in the density field. As WCSPH employs a state equation to compute the pressure, we will end up with oscillations in the pressure field and large pressure forces in diverse directions, which cause instabilities in the simulation process. Our method (right) achieves a smoother density field. We observe almost no density fluctuations in the horizontal direction. When simulating the fluid pillar using DFSPH, the density field is already comparatively smooth without any correction and the effects of our method are less significant. Nonetheless, we observe slight density fluctuations throughout the fluid body, which become worse when employing classical Shepard correction. In contrast, when using our method, they are less significant and we achieve a smoother overall density distribution (see Section 3 of the supplemental material).

In all tested scenarios, we obtain a smooth density field, including the collapsing fluid block (Figure 5), where a smooth density distribution over the whole fluid domain can be observed for WCSPH (lower row). For the simulation with DFSPH (upper row), we obtain a smoother density distribution as well. Although the effect is less prominent, it is still a significant improvement.

We confirm our observations by quantifying the noise in the density field. To that end, we define a seminorm $\sigma_i(\bar\rho)$ to measure local density variance:

$$\sigma_i(\bar\rho) = \left(\sum_{j\in N_i}\left(\frac{\rho_{ij}}{\|\mathbf{x}_{ij}\|}\right)^2\right)^{\frac{1}{2}}, \tag{27}$$

where $\bar\rho = (\rho_1, ..., \rho_n)$ is the vector of the particles' densities, $\rho_{ij} = \rho_i - \rho_j$, and $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$. This could be interpreted as a local L_2-norm of finite differences. In addition, we define two norms

$$\|\sigma(\bar\rho)\|_{\sigma,1} = \frac{1}{n}\sum_{i\in N}|\sigma_i(\bar\rho)| \quad\text{and}\quad \|\sigma(\bar\rho)\|_{\sigma,\infty} = \max_{i\in N}|\sigma_i(\bar\rho)|, \tag{28}$$

where $\sigma(\bar\rho) = (\sigma_1(\bar\rho), ..., \sigma_n(\bar\rho))$. While the seminorm $\sigma_i(\bar\rho)$ characterizes local smoothness, the norms $\|\sigma(\bar\rho)\|_{\sigma,1}$ and $\|\sigma(\bar\rho)\|_{\sigma,\infty}$ provide a global characterization.

A snapshot of the dam break scenario is depicted in Figure 6, where the particles are color-coded with respect to $\sigma_i(\bar\rho)$. We observe that our method reduces the local density variance, both at the surface and inside the fluid.

Figure 7 shows the norm $\|\sigma(\bar\rho)\|_{\sigma,1}$ over time. This plot confirms that we achieve a smoother density field for both pressure models, i.e., we reduce the local density variance by orders of magnitudes. In all examples, we obtain a reduced mean and max density variance. In particular, in the fluid pillar scenario, the mean density variance is almost zero at all times in the simulation, as Figure 8 shows. While

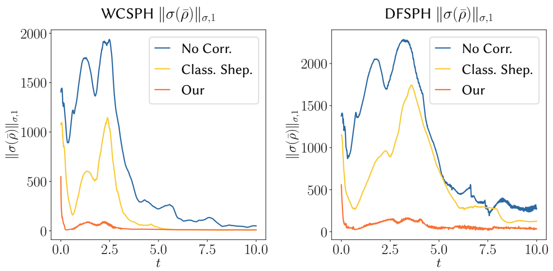

Fig. 7. The mean local density variance $\|\sigma\|_{\sigma,1}$ over time for the corner dam break scenario (Figure 6) simulated with WCSPH (left) and DFSPH (right). Our method results in a much lower density variance, which is consistent with the visualization of the local density variance in Figure 6.

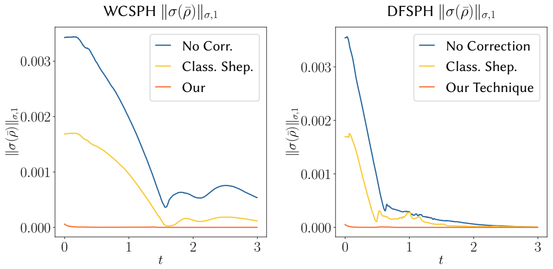

Fig. 8. Measurements of the mean local density variance for the fluid pillar scenario (Figure 2) simulated with WCSPH (left) and DFSPH (right) comparing the density smoothness with the norm $\|\sigma(\bar{\rho})\|_{\sigma,1}$. Classical Shepard correction reduces the local density variance compared to the simulation conducted without correction. Yet, we outperform both and achieve almost zero local density variance.

a noisy density field occurs for the uncorrected and the Shepard correction (at least until the fluid comes to rest), we obtain a smooth density gradient using our correction. The same behavior can be observed for $\|\sigma(\bar{\rho})\|_{\sigma,\infty}$. Measurements for other scenarios and plots of $\|\sigma(\bar{\rho})\|_{\sigma,\infty}$ can be found in the supplemental material.

6.2 Volume Evaluation

We recognized a slightly increased volume of the fluid when applying our method. To further investigate this fact, we measure the volume occupied by the fluid for two scenarios. The first one is the fluid pillar depicted in Figure 9. Additionally, we simulate a fluid cube with an edge length of 2 m. In this scenario, we use hard-coded reflective boundaries (i.e., particle positions are clipped at the domain boundaries and velocities are reflected) to exclude distortion of the measurement due to the boundary model. Both scenarios are simulated until the fluid comes to rest, then the enclosed volume is computed. To this end, we generate an iso-surface of the fluid with an offset of 0.025 m and measure its volume. Table 1 provides an overview of the measurements. In both scenarios, all methods slightly underestimate the fluid volume. With our method, this underestimation is reduced by a factor of about 2.9 for DFSPH in the fluid block scenario. In the fluid pillar, there is a volume loss of 0.6 % with our method compared to 4.1 % without kernel correction. For WCSPH, we reduce the volume loss by a factor of 4.33.

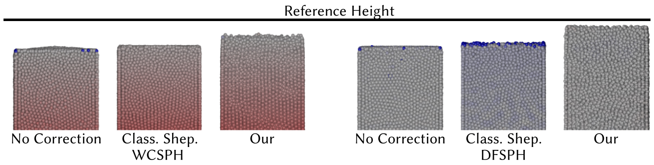

Fig. 9. Volume preservation tests. The rest states of simulations of the fluid pillar scenario with WCSPH (left three) and DFSPH (right three) are shown. When using our method, we almost reach the analytically computed height. As shown in Table 1, we decrease the volume loss compared to simulations using classical Shepard and no correction.

6.3 Convergence and Performance

We analyze our algorithm regarding convergence rate and investigate the stability and performance of the simulation. The study is conducted on the corner dam break scenario consisting of 125 k particles with an initial distance of 0.05 m and a smoothing kernel size of $h = 0.1$ m. When simulating with WCSPH, the stiffness constant is set to $k = 75\,000$. For DFSPH, we set the maximum allowed density error to 0.01 % and the maximum allowed divergence error to 0.1 %. We increase the time step from $\Delta t = 1$ ms up to $\Delta t = 7.5$ ms by steps of 0.5 ms, measuring the needed iterations of our algorithm to converge. The convergence criteria are set $\varepsilon_1 = 0.0001$ and $\varepsilon_2 = 0.0005$. On the left of Figure 10, the measurements considering the convergence rate of our method are given. Our method shows a very fast average convergence rate. For DFSPH and $\Delta t = 4$ ms, we are able to maintain an average convergence rate of 1.45 iterations, and even with very large time steps such as $\Delta t = 7.5$ ms, our method only needs on average 4.7 iterations to converge. Note that the convergence of our technique is independent of the particle count. In Figure 12 (right), we plot the average convergence rate for the collapsing fluid block, while increasing the particle count. The time step was set to $\Delta t = 1$ ms. It is observable that we maintain an almost constant convergence rate, and that our method scales well with increasing particle count.

Since DFSPH itself employs two iterative solvers, we investigate their convergence rate when combining them with our method. To this end, we measure the needed iterations the divergence-free and constant-density solver when including our technique as well as without kernel corrections (Figure 10, right). We observe that the

Table 1. Volume preservation comparison. We conduct our test on two different scenarios. The fluid pillar scenario (Figure 9, 60 k particles) and a fluid block of size 2 m × 2 m × 2 m (64 k particles) are simulated until the rest state is reached, then the enclosed volume is measured. For WCSPH and DFSPH, we decrease the volume loss. Measurements are given in m^3.

		Ref.	WCSPH	DFSPH
Fluid Block	**No Corr.**	8	7.49 (6.3%)	7.6 (4.9%)
	Shepard		7.53 (5.9%)	7.73 (3.3%)
	Our		7.69 (3.8%)	7.85 (1.7%)
Fluid Pillar	**No Corr.**	7.5	7.01 (6.5%)	7.19 (4.1%)
	Shepard		7.09 (5.5%)	7.29 (2.9%)
	Our		7.39 (1.5%)	7.45 (0.6%)

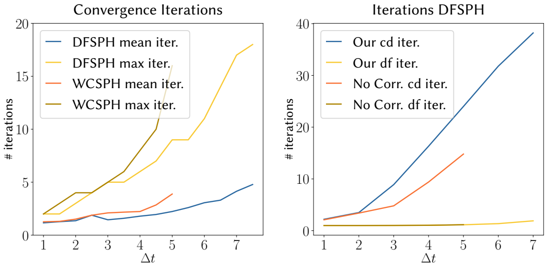

Fig. 10. Convergence analysis. We measured max and mean iterations of our method in the corner dam break scenario, while increasing Δt (left). Additionally, for DFPSH we measured the mean iterations that the constant-density (cd) and the divergence-free (df) solver needed to converge (right).

constant-density solver needs more iterations to converge when using our approach, e.g., the constant-density solver of DFSPH requires about 10 iterations without kernel corrections and 16 with our method. Nonetheless, we are able to maintain a stable simulation up to $\Delta t = 7.5$ ms, whereas the largest possible time step for the simulation with uncorrected kernel is $\Delta t = 5$ ms.

Our method introduces a little computational overhead. In the tested scenarios, we observe that the compute time per step is increased on average by about 20% when employing our method. Most of the overhead can be attributed to the fact that we need to include the boundary particles into our model and, therefore, need an extra loop over them. We observe that the overhead is reduced in scenes that incorporate only few boundary particles.

6.4 Kernel Gradient Correction

We examine the error of the kernel gradient by computing $\|\xi_{s_i}\|$ and the mean distance between the active particle and the ghost particle. The measurements are given for the corner dam break scenario. In Figure 11, the kernel gradient error is color-coded according to $\|\xi_{s_i}\|$. We observe that $\|\xi_{s_i}\|$ is considerably reduced when using classical Shepard correction (center) compared to a simulation without correction (left). Our method further reduces the error, especially in sparsely sampled regions such as within the splashes. At first glance, the error appears to be rather small, but in fact, it is not negligible. To gain a deeper understanding of the error we compute the mean distance between particle i and the virtually placed particle s_i. In Figure 12 (left), the mean distance in relation to h is plotted over time. When simulating without any kernel correction, particle s_i would be placed in a distance between $0.08h$ and $0.12h$ on average. This distance is rather small with our method and classical Shepard correction alike.

7 DISCUSSION AND FUTURE WORK

Considering DFSPH, we recognize varying run-time overhead. When incorporating our method, the constant-density solver of DFSPH sometimes needed fewer iterations and in others more. Unfortunately, we could not find a pattern when these cases occur. We assume that in some cases the pressure solver is badly preconditioned when including our model into DFSPH. Choosing a different preconditioner could help solve this issue. A more elaborative study on this fact might be an interesting future research direction.

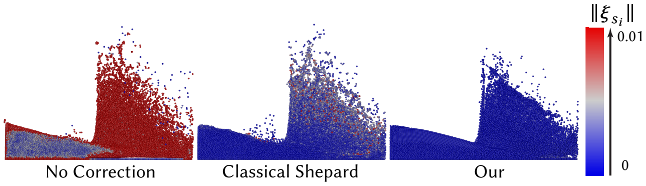

Fig. 11. Kernel gradient correction study. The particles are color-coded with $\|\xi_{s_i}\|$. From left to right: simulation without kernel correction, with classical Shepard, and with our correction.

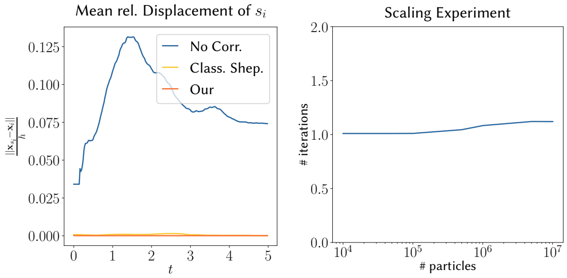

Fig. 12. Left: Gradient correction showing the mean distance between the current and the ghost particle over time. Right: Convergence study showing the average number of iterations for increasing number of particles.

Since the boundary particles contribute to the density computation, we include them into our solver to compute correction factors. This is a performance issue if the boundary objects dominate the scenario. However, this issue can be addressed by incorporating other state-of-the-art boundary models, such as pressure-based boundaries [Band et al. 2018] or moving least squares [Band et al. 2017].

When applying our gradient correction method, ghost forces are introduced into the simulation. When additionally using our kernel correction method, these forces become subtle. These ghost forces alter the simulation in a way that the discretization becomes more accurate and that the gradient kernel sum equals zero (at least for any constant field). Combining our method with other kernel and gradient correction schemes would be interesting to investigate.

As documented in Table 1, we observe a relatively high volume loss for DFSPH. At first sight, this seems to be a contradiction to the employed constant-density condition. However, when considering the fluid volume as a sum of individual particle volumes, i.e., $V^{\text{tot}} = \sum_{i=1}^{N} \frac{m_i}{\rho_i}$, there is no observable volume loss. This means that the volume loss is not reflected in density errors. Furthermore, it indicates that there might be a systematic error hidden in the SPH model itself, in the discretization process, or in other approximations made during the simulation. A more detailed investigation of this fact is a goal for future research.

We compare our model with classical Shepard correction. As mentioned in Section 2, it also works with higher-order kernel correction schemes and it would be interesting to combine our technique with them.

8 CONCLUSION

We presented a novel kernel correction technique that accounts for the error introduced by the discretization of the fluid. Our method is unconditionally stable and efficient, as it converges very fast, i.e., our solver needs on average fewer than 5 iterations to converge, even in dynamic scenes with very large time steps such as $\Delta t = 7\,\text{ms}$.

Compared to classical Shepard correction, we improve the smoothness of the density field by reducing the noise in it and, therefore, improve the stability of the simulation. This also improves the configuration of the pressure field and pressure force field. A suitable norm to measure both local and global smoothness was presented. We show that our technique improves the volume preservation of the fluid and, hence, increases the accuracy of the simulation. Overall, our method complies with the SPH concept, is efficient to compute, and can be used independently of the pressure model.

ACKNOWLEDGMENTS

This work is partly supported by “Kooperatives Promotionskolleg Digital Media” at Hochschule der Medien and the University of Stuttgart. Special thanks go to Yannic Schoof for the final renderings and Jonathan Ziegler for the voice-over in the video.

REFERENCES

M. P. Tulin A. Colagrossi, M. Landrini. 2001. A Lagrangian meshless method for free-surface flows. In *Proceedings of the 4th Numerical Towing Tank Symposium.* https://doi.org/10.1016/S0021-9991(03)00324-3

N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, and M. Teschner. 2012. Versatile rigid-fluid coupling for incompressible SPH. *ACM Transactions on Graphics* 31, 4 (2012), 62:1–62:8. https://doi.org/10.1145/2185520.2185558

S. Band, C. Gissler, M. Ihmsen, J. Cornelis, A. Peer, and M. Teschner. 2018. Pressure boundaries for implicit incompressible SPH. *ACM Transactions on Graphics* 37, 2 (2018), 14:1–14:11. https://doi.org/10.1145/3180486

S. Band, C. Gissler, and M. Teschner. 2017. Moving least squares boundaries for SPH fluids. In *Proceedings of the 13th Workshop on Virtual Reality Interactions and Physical Simulations.* 21–28. https://doi.org/10.2312/vriphys.20171080

M. Becker and M. Teschner. 2007. Weakly compressible SPH for free surface flows. In *Eurographics/SIGGRAPH Symposium on Computer Animation.* 209–218. https://doi.org/10.2312/SCA/SCA07/209-218

T. Belytschko, Y. Krongauz, J. Dolbow, and C. Gerlach. 1998. On the completeness of meshfree particle methods. *International Journal for Numerical Methods in Engineering* 43, 5 (1998), 785–819. https://doi.org/10.1002/(SICI)1097-0207(19981115)43:5<785::AID-NME420>3.0.CO;2-9

J. Bender and D. Koschier. 2017. Divergence-free SPH for incompressible and viscous fluids. *IEEE Transactions on Visualization and Computer Graphics* 23, 3 (2017), 1193–1206. https://doi.org/10.1109/TVCG.2016.2578335

J. Bender, D. Koschier, T. Kugelstadt, and M. Weiler. 2018. Turbulent micropolar SPH fluids with foam. *IEEE Transactions on Visualization and Computer Graphics* 25, 6 (2018), 2284–2295. https://doi.org/10.1109/TVCG.2018.2832080

J. Bonet, S. Kulasegaram, M. X. Rodriguez-Paz, and M. Profit. 2004. Variational formulation for the smooth particle hydrodynamics (SPH) simulation of fluid and solid problems. *Computer Methods in Applied Mechanics and Engineering* 193, 12 (2004), 1245–1256. https://doi.org/10.1016/j.cma.2003.12.018

J. Bonet and T.-S. L. Lok. 1999. Variational and momentum preservation aspects of Smooth Particle Hydrodynamic formulations. *Computer Methods in Applied Mechanics and Engineering* 180, 1 (1999), 97–115. https://doi.org/10.1016/S0045-7825(99)00051-1

S. J. Cummins and M. Rudman. 1999. An SPH projection method. *Journal of Computational Physics* 152, 2 (1999), 584–607. https://doi.org/10.1006/jcph.1999.6246

M. Desbrun and M.-P. Gascuel. 1996. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Proceedings of the Eurographics Workshop on Computer Animation and Simulation ’96.* 61–76.

M. Ellero, M. Serrano, and P. Español. 2007. Incompressible smoothed particle hydrodynamics. *Journal of Computational Physics* 226 (2007), 1731–1752. https://doi.org/10.1016/j.jcp.2007.06.019

F.-J. Fritz, B. Huppert, and W. Willems. 1979. *Stochastische Matrizen.* Springer.

G. Frobenius. 1912. Über Matrizen aus nicht negativen Elementen. *Sitzungsberichte der Preußischen Akademie der Wissenschaften zu Berlin* 26 (1912), 456–477. https://doi.org/10.3931/e-rara-18865

G. C. Ganzenmüller. 2015. An hourglass control algorithm for Lagrangian Smooth Particle Hydrodynamics. *Computer Methods in Applied Mechanics and Engineering* 286 (2015), 87–106. https://doi.org/10.1016/j.cma.2014.12.005

N. Grenier, M. Antuono, A. Colagrossi, D. Le Touzé, and B. Alessandrini. 2009. An Hamiltonian interface SPH formulation for multi-fluid and free surface flows. *Journal of Computational Physics* 228, 22 (2009), 8380–8393. https://doi.org/10.1016/j.jcp.2009.08.009

R. A. Horn and C. R. Johnson. 2012. *Matrix Analysis.* Cambridge University Press.

M. Huber, S. Reinhardt, D. Weiskopf, and B. Eberhardt. 2015. Evaluation of surface tension models for SPH-based fluid animations using a benchmark test. In *Proceedings of Workshop on Virtual Reality Interaction and Physical Simulation.* 41–50. https://doi.org/10.2312/vriphys.20151333

M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, and M. Teschner. 2014a. Implicit incompressible SPH. *IEEE Transactions on Visualization and Computer Graphics* 20, 3 (2014), 426–435. https://doi.org/10.1109/TVCG.2013.105

M. Ihmsen, J. Orthmann, B. Solenthaler, A. Kolb, and M. Teschner. 2014b. SPH fluids in computer graphics. In *EG 2014 - STARs.* 21–42. https://doi.org/10.2312/egst.20141034

G. R. Johnson and S. R. Beisel. 1996. Normalized smoothing functions for SPH impact computations. *International Journal for Numerical Methods in Engineering* 39, 16 (1996), 2725–2741. https://doi.org/10.1002/(SICI)1097-0207(19960830)39:16<2725::AID-NME973>3.0.CO;2-9

Y. Krongauz and T. Belytschko. 1996. Enforcement of essential boundary conditions in meshless approximations using finite elements. *Computer Methods in Applied Mechanics and Engineering* 131, 1 (1996), 133–145. https://doi.org/10.1016/0045-7825(95)00954-X

W.-K. Liu, S. Li, and T. Belytschko. 1997. Moving least-square reproducing kernel methods (I) Methodology and convergence. *Computer Methods in Applied Mechanics and Engineering* 143, 1 (1997), 113–154. https://doi.org/10.1016/S0045-7825(96)01132-2

R. V. Mises and H. Pollaczek-Geiringer. 1929. Praktische Verfahren der Gleichungsauflösung. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* 9, 2 (1929), 152–164. https://doi.org/10.1002/zamm.19290090206

J. J. Monaghan. 1988. An introduction to SPH. *Computer Physics Communications* 48, 1 (1988), 89–96. https://doi.org/10.1016/0010-4655(88)90026-4

J. J. Monaghan. 1992. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics* 30, 1 (1992), 543–574. https://doi.org/10.1146/annurev.aa.30.090192.002551

J. J. Monaghan. 2005. Smoothed particle hydrodynamics. *Reports on Progress in Physics* 68, 8 (2005), 1703–1759. https://doi.org/10.1088/0034-4885/68/8/r01

J. P. Morris. 1996. A study of the stability properties of smooth particle hydrodynamics. *Publications of the Astronomical Society of Australia* 13, 1 (1996), 97–102. https://doi.org/10.1017/S1323358000020610

J. P. Morris, P. J. Fox, and Y. Zhu. 1997. Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics* 136, 1 (1997), 214–226. https://doi.org/10.1006/jcph.1997.5776

P. W. Randles and L. D. Libersky. 1996. Smoothed Particle Hydrodynamics: Some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering* 139, 1 (1996), 375–408. https://doi.org/10.1016/S0045-7825(96)01090-0

H. Schechter and R. Bridson. 2012. Ghost SPH for animating water. *ACM Transactions on Graphics* 31, 4 (2012), 61:1–61:8. https://doi.org/10.1145/2185520.2185557

D. Shepard. 1968. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference.* 517–524. https://doi.org/10.1145/800186.810616

B. Solenthaler and R. Pajarola. 2009. Predictive-corrective incompressible SPH. *ACM Transactions on Graphics* 28, 3 (2009), 40:1–40:6. https://doi.org/10.1145/1531326.1531346

G. L. Vaughan, T. R. Healy, K. R. Bryan, A. D. Sneyd, and R. M. Gorman. 2008. Completeness, conservation and error in SPH for fluids. *International Journal for Numerical Methods in Fluids* 56, 1 (2008), 37–62. https://doi.org/10.1002/fld.1530

M. Weiler, D. Koschier, M. Brand, and J. Bender. 2018. A physically consistent implicit viscosity solver for SPH fluids. *Computer Graphics Forum* 37, 2 (2018), 145–155. https://doi.org/10.1111/cgf.13349

A CONVERGENCE PROOF

In Section 4, we derived a fixed-point problem in a linear formulation $Ac = c$, where $A \in \mathbb{R}^{N \times N}$ and $c \in \mathbb{R}^N$ with

$$a_{ij} = \frac{m_j W_{ij}}{\sum_{k \in N_j} m_k W_{jk}}.$$

From a mathematical point of view, the formulation corresponds to an eigenvalue equation to the eigenvalue 1. Therefore, it is reasonable to analyze the structure of the matrix A to prove existence and uniquness (up to scaling) of a corresponding eigenvector. These results will be the key for the convergence proof of the power method.

For our derivation, we assume that the masses are equal, i.e., $m = m_i$ for all $i = 1, \dots, N$. Before starting the investigation of the matrix A, we introduce the concept of connected fluid bodies regarding Lagrangian approaches.

Definition A.1. A fluid body with positions $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^3$ (or $\in \mathbb{R}^2$) is called connected if for all pairs $\mathbf{x}_i, \mathbf{x}_j$ there exist a $L \in \mathbb{N}$ and $\mathbf{x}_{k_1}, \dots, \mathbf{x}_{k_L}$ such that $\mathbf{x}_i \in N_{k_1} \setminus \partial N_{k_1}$, $\mathbf{x}_j \in N_{k_L} \setminus \partial N_{k_L}$, and $\mathbf{x}_{k_l} \in N_{k_{l+1}} \setminus \partial N_{k_{l+1}}$ for all $l = 1, \dots, L-1$, i.e., $\mathbf{x}_{k_1}$ and $\mathbf{x}_{k_L}$ are connected by a path $\mathbf{x}_{k_1}, \dots, \mathbf{x}_{k_L}$ of particles that are chained by overlapping smoothing kernels.

The next lemma proves algebraic properties of the matrix A and links the previous definition of connection to the matrix A.

Lemma A.2. *The matrix A has the following properties:*

1. *It is a left stochastic matrix, i.e.,*
   - $0 \leq a_{ij} \leq 1$ *for* $i, j = 1, \dots, N$,
   - $\sum_{i=1}^{N} a_{ij} = 1$ *for* $j = 1, \dots, N$.
2. *The diagonal is positive, i.e.,* $a_{ii} > 0$ *for* $i = 1, \dots, N$.
3. *It is irreducible if and only if the fluid body is connected.*

Proof. If $\mathbf{x}_j \notin N_i$, then $W_{ij} = 0$ and hence $a_{ij} = 0$. If $\mathbf{x}_j \in N_i$, we obtain

$$0 \leq \frac{m_j W_{ij}}{\sum_{k \in N_j} m_k W_{jk}} \leq \frac{m_j W_{ij}}{m_j W_{ji}} = 1,$$

as $m_j = m > 0$ and $W_{ij} \geq 0$ for $i, j = 1, \dots, N$. Furthermore, summing up the columns of the matrix we get

$$\sum_{i=1}^{N} a_{ij} = \sum_{i=1}^{N} \frac{m_j W_{ij}}{\sum_{k \in N_j} m_k W_{jk}} = \frac{\sum_{i \in N_j} m_j W_{ij}}{\sum_{k \in N_j} m_k W_{jk}} = 1.$$

The second statement follows from the properties of a smoothing kernel. To prove the last statement we consider the fluid body to be an undirected graph G, where particle positions are vertices and (open) neighborhood relations correspond to edges. Denoting the associated adjacency matrix by $\widetilde{A}$, we get the following relationship

$$\tilde{a}_{ij} = \begin{cases} 1 & \text{if } a_{ij} > 0 \text{ and } i \neq j, \\ 0 & \text{else.} \end{cases}$$

Since the irreducibility of non-negative matrices is equivalent to $(I + A)^{N-1} > 0$ [Horn and Johnson 2012, Theorem 6.2.24], we obtain the equivalence

$$A \text{ irreducible} \iff \widetilde{A} \text{ irreducible}.$$

A well-known fact from graph theory is that the adjacency matrix $\widetilde{A}$ is irreducible if and only if the corresponding graph G is connected. By construction, the graph is connected if and only if the fluid body is connected. □

The following lemma shows the spectral-theoretic properties of the matrix A, characterizing the existence and uniqueness of an eigenvector to the eigenvalue 1 and the location of other eigenvalues.

Lemma A.3. *For the matrix A with eigenvector λ, the following properties hold:*

1. *The value* 1 *is an eigenvalue.*
2. *If* $\lambda \neq 1$, *then* $|\lambda| < 1$.
3. *If the matrix A is irreducible, then the eigenvalue $\lambda = 1$ is simple and the corresponding eigenvector can be chosen to be positive, i.e., every entry of the vector is positive.*

Proof. The first statement is a direct consequences of Lemma A.2 and the second follows from the fact that A is a left stochastic matrix with a positive diagonal [Fritz et al. 1979, Theorem 2.6]. The last statement is a well-known result from the Perron-Frobenius theory for non-negative irreducible matrices [Frobenius 1912]. □

The next theorem is the final result proving the convergence of the used power method.

Theorem A.4. *For the matrix A, the following statements hold:*

1. *The power method converges for every initial value, i.e., the limit* $c^* = \lim_{k \to \infty} \frac{A^k c_0}{\|A^k c_0\|}$ *exists for every* $c_0 \in \mathbb{R}^N$.
2. *For a non-negative initial value, the power method converges to a non-negative eigenvector to the eigenvalue* 1, *i.e., for* $c_0 \geq 0$, *the power method converges to an eigenvector* $c^* \geq 0$ *to the eigenvalue* 1.
3. *Let the fluid body be connected. For a non-negative initial value, the power method converges to the unique positive eigenvector (up to scaling).*

Proof. By Lemma A.2 and Lemma A.3, the matrix A is a stochastic matrix with $\lambda = 1$ as the only eigenvalue with $|\lambda| = 1$. Hence, the series A^k converges [Fritz et al. 1979, Theorem 3.4] and, therefore, the limit of the power method $\lim_{k \to \infty} \frac{A^k c_0}{\|A^k c_0\|}$ exists for every initial value c_0. For $c = \lim_{k \to \infty} A^k c_0$, we obtain

$$Ac = A \lim_{k \to \infty} A^k c_0 = \lim_{k \to \infty} A^{k+1} c_0 = c,$$

which implies that the power method converges to a (normalized) eigenvector c^* to the eigenvalue 1. As the initial value is $c_0 \geq 0$, the limit has to be $c^* \geq 0$.

If the fluid body is connected, we obtain by Lemma A.2 that the matrix A is irreducible. Lemma A.3 implies the simplicity of the eigenvalue 1 and the existence of a positive eigenvector. Therefore, the limit of the power method has to be positive because the initial value is non-negative and the eigenspace is one-dimensional. □