A comparison of linear consistent correction methods for first-order SPH derivatives

Lukas Westhofen
RWTH Aachen University
Aachen, Germany

Stefan Jeske
RWTH Aachen University
Aachen, Germany

Jan Bender
RWTH Aachen University
Aachen, Germany

Figure 1: (Left) A coupled simulation using the elasticity model of Peer et al. [2018] and the viscosity model of Takahashi et al. [2015] with Kernel Gradient Correction. In this simulation an elastic bunny is spun around in a washing machine drum, which is filled with viscous soap and water. (Right) Viscous buckling being observable when using Kernel Gradient Correction for the method of Takahashi et al. [2015]. This buckling effect could not be observed for the non-corrected version using the same parameters.

ABSTRACT

A well-known issue with the widely used Smoothed Particle Hydrodynamics (SPH) method is the neighborhood deficiency. Near the surface, the SPH interpolant fails to accurately capture the underlying fields due to a lack of neighboring particles. These errors may introduce ghost forces or other visual artifacts into the simulation.

In this work we investigate three different popular methods to correct the first-order spatial derivative SPH operators up to linear accuracy, namely the Kernel Gradient Correction (KGC), Moving Least Squares (MLS) and Reproducing Kernel Particle Method (RKPM). We provide a thorough, theoretical comparison in which we remark strong resemblance between the aforementioned methods. We support this by an analysis using synthetic test scenarios. Additionally, we apply the correction methods in simulations with boundary handling, viscosity, surface tension, vorticity and elastic solids to showcase the reduction or elimination of common numerical artifacts like ghost forces. Lastly, we show that incorporating the correction algorithms in a state-of-the-art SPH solver only incurs a negligible reduction in computational performance.

CCS CONCEPTS

• **Computing methodologies → Physical simulation**.

KEYWORDS

smoothed particle hydrodynamics, computer animation, moving least squares, reproducing kernel particle method

ACM Reference Format:
Lukas Westhofen, Stefan Jeske, and Jan Bender. 2023. A comparison of linear consistent correction methods for first-order SPH derivatives. In *Proceedings of The ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA '23)*. ACM, New York, NY, USA, Article 1, 12 pages. https://doi.org/10.1145/3606933

1 INTRODUCTION

Smoothed Particle Hydrodynamics established itself in the computer graphics community as an essential method due to its flexibility in simulating various physical phenomena. Recent research include the simulation of fluids, deformable solids, granular materials, snow and their interaction with each other. As a meshless, Lagrangian method, SPH determines the required properties like density or pressure by interpolating between unstructured sampling points which we call particles. These particles represent a certain volume of the underlying continuum and thus also carry our field quantities, which we need to solve the desired differential equations. The interpolated value is determined by summation over neighboring particles weighted by their volume and by the

SCA '23, August 04–06, 2023, Los Angeles, CA

so-called kernel function which both together form the SPH shape function.

In practice, the SPH interpolation works well in densely sampled regions. However, in sparse regions where there are less particles, the accuracy of the SPH interpolant may deteriorate. This phenomenon is known as the *neighborhood deficiency problem*. Its origin lies in the shape function employed by SPH, which fails to accurately restore the underlying fields if not enough particles are present to meet the normalization condition of the kernel function. As a consequence, various spurious boundary effects like ghost forces may appear if the inherent problem is not addressed. These ghost forces can lead to visual artifacts, artificial damping or hinder a physical accurate depiction altogether. These issues are especially prominent in the first-order derivatives from which many forces are derived.

Different approaches have been proposed to address the neighborhood deficiency problem. Among others, this includes the Kernel Gradient Correction which reweights the existing sampling points to reduce the introduced error. Thus, it can be viewed as a correction function for the SPH interpolant. Additionally, other interpolation methods have been included into the SPH pipeline like Moving Least Squares or the Reproducing Kernel Particle Method. Usually, these are used as drop-in replacements for the SPH interpolant.

In our work, we analyze three important interpolation methods for first-order derivatives, namely Kernel Gradient Correction (KGC), Moving Least Squares (MLS) and Reproducing Kernel Particle Method (RKPM). While many authors already employed these methods to, e.g., accurately capture rotational motion, our work provides an in-depth comparison of the aforementioned methods. This includes the following contributions:

- A mathematical investigation of KGC, MLS and RKPM as correction methods to the SPH interpolant.
- A derivation of linear consistent, first-order SPH operators with the aforementioned methods.
- A reduction of visual artifacts stemming from the neighborhood deficiency in application examples of boundary handling, viscosity, surface tension, vorticity and elastic solids with negligible, additional computational effort.

2 RELATED WORK

Originally formulated by Gingold and Monaghan [1977], Smoothed Particle Hydrodynamics is a meshless, Lagrangian simulation technique which was introduced into the field of computer graphics by Desbrun and Gascuel [1996]. Since then, many physical effects have been successfully simulated solely using SPH which include fluids [Bender and Koschier 2015; Ihmsen et al. 2014a], highly viscous materials [Peer et al. 2015; Takahashi et al. 2015; Weiler et al. 2018], surface tension effects [Akinci et al. 2013; Becker and Teschner 2007], granular materials [Alduán and Otaduy 2011; Ihmsen et al. 2013], snow [Gissler et al. 2020], turbulent flows [Bender et al. 2017], and elastic solids [Kugelstadt et al. 2021; Peer et al. 2018]. An extensive overview over the current research in SPH can be found in the State-of-the-Art reports of Ihmsen et al. [2014b] and Koschier et al. [2022].

While SPH works great for simulating the aforementioned effects, visual artifacts can occur due to the error-prone interpolation in sparsely sampled regions. An overview of the errors accompanying the different state-of-the-art SPH interpolants can be found in the work of Price [2012]. Furthermore, Violeau and Fonty [2019] provide an investigation of the error introduced by the smoothing in relation to the standard deviation of the kernel. To mitigate these issues, different correction methods and alternative, meshless shape functions have been proposed. One approach is to sample the void region and extrapolate the fluid's properties on the new sampling points, which is used by Schechter and Bridson [2012]. However, this method introduces a significant computational overhead by the necessity to introduce and distribute additional particles instead of just using the existing ones. Apart from the correction methods we will investigate in this work — namely KGC, MLS & RKPM — there also exists Corrected SPH (CSPH) by Bonet and Lok [1999] which directly corrects the kernel function instead of its gradient to be first-order consistent in a similar manner as Liu et al. [1995] propose for RKPM. Constant rank MLS, also known as Shepard interpolation [Shepard 1968], has been applied to the density computation in SPH by Reinhardt et al. [2019]. Extensions have also been proposed like Generalized Moving Least Squares (GMLS) [Atluri et al. 1999] to MLS and hermite RKPM [Liu et al. 1996] to RKPM. Finally, the works of Belytschko et al. [1998] and Fries and Matthies [2004] provide a great overview over other meshless approaches using the aforementioned shape functions. We focus on using the KGC, MLS & RKPM since they have efficiently computable, linearized versions and can be directly integrated into the SPH pipeline.

In computer graphics, many approaches already use the presented techniques. In SPH, MLS has been used to correct the spurious interface artifacts towards particle-based boundaries [Akinci et al. 2012]. Band et al. [2017] propose its application for the normal calculation of the boundary. Thus, when a particle is close to the boundary, a prototype particle pattern can be constructed for a smooth pressure force independent of the underlying particle sampling. Likewise, Band et al. [2018] use MLS to extrapolate the fluid's pressure onto boundary particles for a more accurate pressure force. However, their method requires an additional neighborhood search on the boundary particles. We will later show that MLS can be directly integrated into particle-based boundaries without the need of prototype particle patterns or additional neighborhood information while retaining the same advantages.

Spurious boundary effects also hinder the simulation of elastic solids and solid-fluid interaction using SPH. Most authors employ KGC like in the works of Ganzenmüller [2015] and Peer et al. [2018]. These propose a linear elasticity model which uses a corrected kernel function for calculating the deformation gradient using SPH. As we show later, KGC is used to eliminate ghost forces appearing for rigid-body deformations like rotations. Likewise, Kugelstadt et al. [2021] apply the same kernel gradient correction matrix to their corotated elasticity model. For the simulation of snow, Gissler et al. [2020] follow another approach in advecting the deformation gradient through time. This requires the estimation of the velocity gradient which the authors correct using KGC as well. Coupling fluids and solids, Joubert et al. [2020] also employ KGC to apply Neumann boundary conditions to couple a DEM simulation with SPH using pressure fields.

Lastly, MLS and RKPM have already been used in conjunction with other discretization methods for simulating various effects in computer animation. Müller et al. [2004] use MLS for the simulation of deformable solids and incorporate effects like melting. Pauly et al. [2005] extend this model to allow the simulation of fracturing. Additionally, Adams and Wicke [2009] provide a tutorial on meshless methods which focuses on the applications of MLS and comparison to SPH. Outside the research of SPH, first- and second-order MLS has been used by Wang et al. [2020] for the simulation of surface tension. Hu et al. [2018] extend the Material Point Method (MPM) with MLS to simulate effects like cutting and fracturing while being faster than traditional MPM. Finally, Chen et al. [2020] use RKPM with a Galerkin discretization for the simulation of various effects like wetting, fracturing, mixing and for the simulation of snow and deformable solids.

3 SPH DISCRETIZATION

In this paper we are interested in the simulation of physical phenomena using Smoothed Particle Hydrodynamics. We will introduce SPH at the example of incompressible fluids and in this context discuss its issues at the free surface. Note that these effects also appear when simulating, e.g. , deformable solids which we will also analyze over the course of this paper.

For the simulation of free-surface flows, we will employ the Navier-Stokes equations as our mathematical model. These consist of the continuity equation and the momentum equation. The former is given for incompressible fluids by

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot \mathbf{v} = 0, \tag{1}$$

where ρ is the density of the fluid and $\mathbf{v}$ the velocity. Further, Eq. (1) is given in Lagrangian coordinates, thus $\frac{D}{Dt}$ denotes the material derivative. The second Navier-Stokes equation can be derived from the Cauchy momentum equation by inserting the stress tensor for Newtonian fluids

$$\rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \mathbf{P} + \mathbf{f} \tag{2}$$

$$\mathbf{P} = p\mathbb{I} + 2\mu \mathbf{E} \tag{3}$$

$$\mathbf{E} = \frac{1}{2}\left(\nabla \mathbf{v} + (\nabla \mathbf{v})^T\right). \tag{4}$$

Here $\mathbf{P}$ denotes the Piola-Kirchhoff stress tensor, $\mathbf{f}$ external body forces, p the internal pressure, μ the dynamic viscosity and $\mathbf{E}$ the viscous strain rate tensor. Inserting Eq. (3) into Eq. (2) yields the well-known second Navier-Stokes equation for incompressible fluids

$$\frac{D\mathbf{v}}{Dt} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{v} + \frac{\mathbf{f}}{\rho}, \tag{5}$$

where $\nu = \mu/\rho$ is the kinematic viscosity. The advection of the fluid is governed by the acceleration terms on the right hand side of Eq. (5), which represent from left to right the pressure force, the viscous force and external forces like gravity.

For simulating the fluid flow according to Eqs. (1) and (5), we use Smoothed Particle Hydrodynamics (SPH) [Gingold and Monaghan 1977]. Discretizing the continuum, SPH samples the desired domain by particles which represent the body's volume. Each particle i contains data like mass m_i, density ρ_i or position $\mathbf{x}_i$. A quantity A for particle i can be determined by using the SPH interpolation

$$\langle A(\mathbf{x}_i)\rangle = \sum_{j \in \mathcal{N}_i} \frac{m_j}{\rho_j} A(\mathbf{x}_j)\, W(\mathbf{x}_i - \mathbf{x}_j, h). \tag{6}$$

In Eq. (6), the value of $A(\mathbf{x}_i)$ is determined as the weighted sum of the property A at neighboring particle positions $\mathbf{x}_j$, where the weighting is determined by the product of the volume the particle represents, i.e. $\frac{m_j}{\rho_j}$, and the kernel function $W(\mathbf{x}_i - \mathbf{x}_j, h)$. The latter can be understood as a Gaussian-like function with smoothing length h. $\mathcal{N}_i$ denotes the set of particles which are inside the compact support region of particle i which is called the neighborhood. In the following, we will use the abbreviations $A_i = A(\mathbf{x}_i)$ and $W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j, h)$.

Note that apart from the density, which is usually computed using Eq. (6), the Navier-Stokes equations (see Eqs. (1) and (5)) and most of the employed non-pressure forces (e.g., surface tension forces) predominantly contain derivative terms. Focusing on the spatial derivative operators, the SPH interpolant of the gradient of a property A is given by

$$\langle \nabla A_i \rangle = \sum_{j \in \mathcal{N}_i} \frac{m_j}{\rho_j} A_j \nabla W_{ij}. \tag{7}$$

Similarly, one can derive the divergence and curl operator in the same fashion [Koschier et al. 2019].

The SPH gradient gives a good estimate for densely sampled regions as Koschier et al. [2019] show, but at sparse regions, e.g. near the free surface, the lack of particles in the neighborhood reduces the interpolation accuracy. While this is hardly a problem for most SPH simulations, there are use cases where neglecting the neighborhood deficiency problem can lead to ghost forces and therefore to visual artifacts. Mathematically, one can derive the interpolation accuracy according to Price [2012] by using a Taylor expansion of A_j around $\mathbf{x}_j$ in Eq. (7):

$$\nabla A_i \approx \langle \nabla A_i \rangle = \sum_{j \in \mathcal{N}_i} \frac{m_j}{\rho_j} A_i \nabla W_{ij} + \sum_{j \in \mathcal{N}_i} \frac{m_j}{\rho_j} \frac{\partial A_i}{\partial \mathbf{x}_i}^T (\mathbf{x}_j - \mathbf{x}_i)\nabla W_{ij} + O\left(||\mathbf{x}_j - \mathbf{x}_i||^2\right). \tag{8}$$

From Eq. (8), it becomes apparent that in order to guarantee an accurate interpolation estimate, the first and last term of the right hand side must vanish while the middle term should equal ∇A_i. The interpolation is called k-th order consistent if it is exact for an input field containing a polynomial with maximum degree of k. Analyzing Eq. (8), Eq. (7) may not even be 0-th order consistent depending on the choice of the kernel function and the particle arrangement. By enforcing

$$\sum_{j \in \mathcal{N}_i} \frac{m_j}{\rho_j} A_i \nabla W_{ij} = \mathbf{0} \tag{9}$$

one can eliminate the constant error, which can be done by subtracting Eq. (9) from Eq. (7), which yields the so-called **difference formula**

$$\nabla A_i = \langle \nabla A_i \rangle - A_i \langle \nabla 1 \rangle = \sum_{j \in \mathcal{N}_i} \frac{m_j}{\rho_j} (A_j - A_i)\nabla W_{ij}. \tag{10}$$

Eq. (10) works well in practice and improves the accuracy.

Another prominent alternative to Eq. (10) is the **symmetric formula**

$$\nabla A_i = \rho_i \sum_{j \in \mathcal{N}_i} m_j \left(\frac{A_i}{\rho_i^2} + \frac{A_j}{\rho_j^2}\right)\nabla W_{ij}. \tag{11}$$

As Price [2012] shows, Eq. (11) has the advantage of conserving linear and angular momentum as well as energy.

4 LINEAR CONSISTENT CORRECTION METHODS

4.1 Correcting the SPH interpolant

In the following we investigate correcting SPH with the approaches KGC, MLS and RKPM which aim to reduce the errors at the free surface introduced by the classic SPH interpolation. In detail, SPH uses the shape function

$$\Phi_j^{\mathrm{SPH}}(\mathbf{x}_i) = \frac{m_j}{\rho_j} W_{ij} \tag{12}$$

for computing interpolated values $A_i = \sum_{j \in \mathcal{N}_i} A_j \Phi_j^{\mathrm{SPH}}(\mathbf{x}_i)$. However, Eq. (12) fails to build a partition of unity as shown by Dilts [1999]. Thus, SPH categorically underdetermines values near the boundary. This introduces spurious boundary effects into the simulation which mostly manifest themselves as ghost forces. The previously mentioned correction algorithms focus on correcting or replacing the basis functions to achieve a partition of unity with linear consistency for the gradient operators. One can reformulate the SPH gradient using its shape function as:

$$\nabla A_i = \sum_{j \in \mathcal{N}_i} A_j \nabla \Phi_j^{\mathrm{SPH}}(\mathbf{x}_i) \tag{13}$$

$$\nabla \Phi_j^{\mathrm{SPH}}(\mathbf{x}_i) = \frac{m_j}{\rho_j} \nabla W_{ij}. \tag{14}$$

Thus, we can apply many shape functions of other meshless methods as drop-in replacements for the SPH interpolant as long as we can formulate the gradient in the same fashion. Furthermore, we can also write the difference (Eq. (10)) and symmetric formula (Eq. (11)) using shape functions

$$\nabla A_i = \sum_{j \in \mathcal{N}_i} \left(A_j - A_i\right) \nabla \Phi_j(\mathbf{x}_i) \tag{15}$$

$$\nabla A_i = \rho_i \sum_{j \in \mathcal{N}_i} m_j \left(\frac{A_i \frac{\rho_j}{m_j} \nabla \Phi_j(\mathbf{x}_i)}{\rho_i^2} - \frac{A_j \frac{\rho_i}{m_i} \nabla \Phi_i(\mathbf{x}_j)}{\rho_j^2} \right). \tag{16}$$

Note that we derived Eq. (16) to be symmetric and thus it conserves linear and angular momentum. As possible shape functions, we will employ KGC for correcting the original SPH interpolant and MLS and RKPM as replacements.

4.2 Linear consistent shape functions

4.2.1 *Kernel Gradient Correction.*

The kernel gradient correction as proposed by Bonet and Lok [1999] aims to eliminate the linear error in SPH formulations while also preserving angular momentum. Analyzing Eq. (8), one can see that for linear accuracy the middle term of the right hand site should be equal to ∇A_i. The corrected interpolant is based on enforcing this condition by multiplication of the kernel with the matrix $\mathbf{L}$ defined by

$$\mathbf{L}(\mathbf{x}_i) = \left[\sum_{j \in \mathcal{N}_i} \frac{m_j}{\rho_j} \left(\mathbf{x}_j - \mathbf{x}_i\right) \otimes \nabla W_{ij} \right]^{-1}. \tag{17}$$

This yields a first-order consistent shape function gradient $\nabla\Phi^{\mathrm{KGC}}$ compliant to Eq. (14)

$$\nabla \Phi_j^{\mathrm{KGC}}(\mathbf{x}_i) = \frac{m_j}{\rho_j} \mathbf{L}_i \nabla W_{ij}. \tag{18}$$

In 3D, the kernel gradient correction requires to store a 3 × 3-matrix per particle. Furthermore, a matrix inversion is required which fails if the matrix in Eq. (17) becomes singular. This is usually the case if the particles are arranged in a plane or a line or for single particles in, e.g., splashes. An approach for tackling this issue which works well in practice is to perform the inversion only on the non-singular matrix spanning its associated subspace. Therefore, we use a singular value decomposition and only calculate the reciprocal values for singular values σ which are bigger than a threshold ε. If a singular value is smaller than ε, then we replace it with a 1 thus leaving the corresponding directions uncorrected.

4.2.2 *Moving Least Squares.*

First introduced by Lancaster and Salkauskas [1981], Moving Least Squares replaces the error-prone interpolants by a shape function which form a partition of unity. An advantage of MLS over SPH is that the interpolant can be derived to have the desired consistency given the dimension of the polynomial basis functions $\mathbf{b}$. Thus, even in sparsely sampled regions, MLS can reconstruct a reasonable solution. MLS has been widely adopted in the graphics community by many authors [Adams and Wicke 2009; Band et al. 2018; Hu et al. 2018; Müller et al. 2004].

In detail, the MLS interpolant of a property A can be derived as

$$\mathbf{A}_i = \sum_{j \in \mathcal{N}_i} \Phi_j^{\mathrm{CMLS}}(\mathbf{x}_i) \mathbf{A}_j \tag{19}$$

$$\Phi_j^{\mathrm{CMLS}}(\mathbf{x}_i) = W_{ij} \mathbf{b}(\mathbf{x}_i)^T \mathbf{M}(\mathbf{x}_i) \mathbf{b}(\mathbf{x}_j) \tag{20}$$

$$\mathbf{M}(\mathbf{x}_i) = \left[\sum_{j \in \mathcal{N}_i} W_{ij} \mathbf{b}(\mathbf{x}_j) \mathbf{b}(\mathbf{x}_j)^T \right]^{-1}, \tag{21}$$

where Φ_j^{CMLS} denotes the (classic) MLS weighting function and $\mathbf{M}$ the so-called *momentum matrix*. Note that W_{ij} denotes the same kernel function as for SPH. For linear consistency in 3D, the polynomial basis is given by $\mathbf{b}(\mathbf{x}_i) = [1\ x_i\ y_i\ z_i]$, where $\mathbf{x}_i = (x_i\ y_i\ z_i)^T$.

Similarly to SPH, the gradient operator can be computed by taking the gradient of the weighting function:

$$\nabla \mathbf{A}_i = \sum_{j \in \mathcal{N}_i} \nabla \Phi_j^{\mathrm{CMLS}}(\mathbf{x}_i) \mathbf{A}_j^T. \tag{22}$$

The lengthy expression for the gradient of the MLS shape function as well as the MLS interpolants for divergence and curl can be derived from Eq. (19) and found in the work of Adams and Wicke [2009].

Computing the MLS gradient requires a substantial amount of operations and therefore is computationally expensive. Thus for computer graphics, Müller et al. [2004] propose a linear consistent alternative:

$$\nabla \mathbf{A}_i = \mathbf{K}(\mathbf{x}_i) \sum_{j \in \mathcal{N}_i} W_{ij} \left(\mathbf{A}_j - \mathbf{A}_i\right) \mathbf{x}_{ij} \tag{23}$$

$$\mathbf{K}(\mathbf{x}_i) = \left[\sum_{j \in \mathcal{N}_i} W_{ij} \mathbf{x}_{ij} \mathbf{x}_{ij}^T \right]^{-1}, \tag{24}$$

where $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$. Both versions of MLS in Eqs. (22) & (23) visually yield the same result for linear accuracy. Thus, for the rest of this work we will only investigate the latter version for performance reasons. Its shape function gradient $\nabla\Phi^{\text{MLS}}$ is given by:

$$\nabla\Phi_j^{\text{MLS}} = -\mathbf{K}\left(\mathbf{x}_i\right)\mathbf{x}_{ij}W_{ij}. \tag{25}$$

4.2.3 Reproducing Kernel Particle Method. Instead of replacing the SPH interpolant completely, the Reproducing Kernel Particle Method by Liu et al. [1995] can be viewed as an extension to the classic SPH formulation. Based on the wavelet analysis of SPH, the authors suggest a corrected kernel function in a continuous sense. Discretizing it and choosing a polynomial basis yields the (classic) RKPM interpolation

$$\mathbf{A}_i = \sum_{j\in\mathcal{N}_i}\Phi_j^{\text{CRKPM}}\left(\mathbf{x}_i\right)\mathbf{A}_j \tag{26}$$

$$\Phi_j^{\text{CRKPM}}\left(\mathbf{x_i}\right) = \frac{m_j}{\rho_j}W_{ij}\mathbf{b}\left(\mathbf{x}_i\right)^T\mathbf{N}\left(\mathbf{x}_i\right)\mathbf{b}\left(\mathbf{x}_j\right) \tag{27}$$

$$\mathbf{N}\left(\mathbf{x}_i\right) = \left[\sum_{j\in\mathcal{N}_i}\frac{m_j}{\rho_j}W_{ij}\mathbf{b}\left(\mathbf{x}_j\right)\mathbf{b}\left(\mathbf{x}_j\right)^T\right]^{-1}. \tag{28}$$

According to Fries and Matthies [2004], RKPM offers the same advantages as MLS. Note that the choice of nodal volumes V_j is free in RKPM. For seamless integration into the SPH pipeline, we choose it to be the same as the SPH volume. The same considerations for stability and the derivative operators as for MLS also apply to RKPM. Additionally, we derive the same linearized formulation for the gradient as we did for MLS. To our knowledge, this formulation has not been used in computer graphics and offers the same performance benefits while still being linearly correct as we will see later. Analogous to the derivation of Müller et al. [2004] we get

$$\nabla\mathbf{A}_i = \mathbf{J}\left(\mathbf{x}_i\right)\sum_{j\in\mathcal{N}_i}\frac{m_j}{\rho_j}W_{ij}\left(\mathbf{A}_j - \mathbf{A}_i\right)\mathbf{x}_{ij} \tag{29}$$

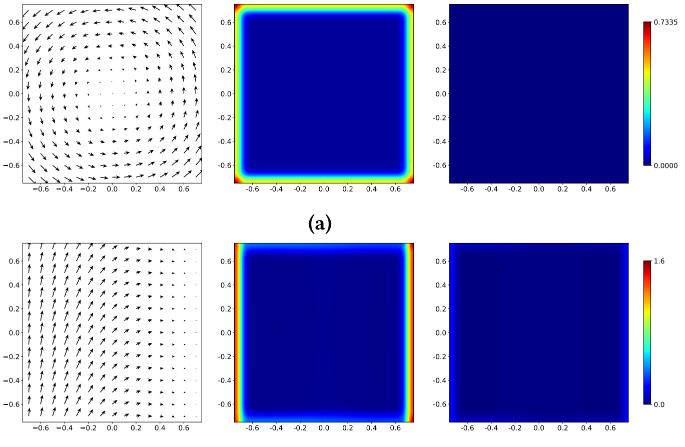
$$\mathbf{J}\left(\mathbf{x}_i\right) = \left[\sum_{j\in\mathcal{N}_i}\frac{m_j}{\rho_j}W_{ij}\mathbf{x}_{ij}\mathbf{x}_{ij}^T\right]^{-1}, \tag{30}$$

which again is our preferred option for linear consistent interpolation and what we will investigate in the following sections. Incorporating it into Eq. (13) yields

$$\nabla\Phi_j^{\text{RKPM}} = -\mathbf{J}\left(\mathbf{x}_i\right)\frac{m_j}{\rho_j}\mathbf{x}_{ij}W_{ij}. \tag{31}$$

4.3 Comparison of KGC, MLS & RKPM

Inserting these shape functions gradients to, e.g., the difference formula in Eq. (10) directly yields a first-order consistent solution. Taking a deeper look at the correction matrices $\mathbf{L}$, $\mathbf{K}$ and $\mathbf{J}$ for KGC, MLS and RKPM in Eqs.(17), (24) and (30), respectively, one can directly draw parallels between the different formulations.



(a)

(b)

Figure 2: Interpolation errors when computing the gradient of a linear (Fig. 2a) and non-linear velocity field (Fig. 2b). Each subfigure shows the following: Left: 2D plot of the velocity field $\mathbf{v}^{\text{rot.}}$ / $\mathbf{v}^{\text{trigonom.}}$. Middle: Frobenius norm of the error between the analytical solution and the results of the difference formula (Eq. (10)). Right: Frobenius norm of the error between the analytical solution and the results of the corrected difference formula (Eq. (15)).

First, when comparing RKPM and MLS, we see that they only differ by the volume factor used in SPH, i.e. $V_i = \frac{m_i}{\rho_i}$. Note that we explicitly chose to use the standard SPH volume since the original formulation of RKPM of Liu et al. [1995] leaves the question for nodal volumes open. We can also choose $V_i = 1$ which results in MLS and RKPM to be identical. In fact, Aluru [2000] denotes the following. For any collocation method — which SPH belongs to as well — the nodal volumes can be chosen completely arbitrarily since they do not deteriorate the solution. When using different nodal volumes both methods will yield different corrections, but they still yield linear correct results.

Second, we compare KGC to RKPM. Inspecting Eqs. (30) and (17), they differ in the computation of the kernel gradient. While KGC employs the standard kernel gradient ∇W_{ij}, RKPM uses a first-order approximation, i.e. $\nabla W_{ij} \approx \mathbf{x}_{ij}W_{ij}$. This is not a significant change since we strive for linear correctness anyway and the approximation as well as the kernel gradient both point in exactly the same direction. Thus, both methods also correct the gradient in a similar way.

Conclusively, all methods perform equally well from a theoretical viewpoint. Although KGC, MLS and RKPM all yield slightly different results for correcting the underlying fields, in our experiments they all produced mathematically similar and visually indistinguishable results.

To validate the effectiveness of the aforementioned methods, we compare them in a synthetic scenario. This consists of a square domain filled with particles arranged in a dense, square lattice. We use the difference formula to calculate the values of the gradient of different velocity fields $\nabla\mathbf{v}\left(x,y\right)$.

First, we analyze the rotational field $\mathbf{v}^{\text{rot.}}\left(x,y\right) = \left(-y,x\right)^T$ depicted in Fig. 2a. As already outlined by Weiler et al. [2018], SPH fails to accurately reconstruct the field at the boundaries, which can be seen from the plotted error. Using KGC the underlying velocity field can be perfectly recreated up to numerical precision. Applying MLS or RKPM yields similar results and validates the first-order correctness of the approaches.

Table 1: Timings for the computation of the correction matrices T_{mat} and the evaluation of the gradient T_{grad} of 1 million particles using the difference formula with SPH, KGC and the classic and linearized versions of MLS and RKPM.

	SPH	KGC	CMLS	MLS	CRKPM	RKPM
T_{mat} [ms]	–	11.23	28.98	11.29	29.18	12.31
T_{grad} [ms]	12.08	13.00	41.95	12.42	44.80	13.94

Second, we want to investigate whether the correction methods also offer an advantage for higher-order functions since fluid behavior is usually highly non-linear. Thus, we use $\mathbf{v}^{\text{trigonom.}}(x, y) = (\cos(2x), -\sin(3x) + \cos(y))$ which can be seen in Fig. 2b. Comparing the results of the standard SPH interpolant with the corrected one using KGC, one can remark that the errors at the boundary are drastically reduced. Still, there remains a small, higher-order error at the boundary. Likewise, using the proposed MLS or KGC variants produce similar results. For further reducing the error at the boundary one would need to use a higher-order interpolation method. These could be achieved by using classic MLS as in Eq. (19) or RKPM in Eq. (26) with higher-order polynomial basis functions at the cost of additional computational effort.

Note that for the linear consistency, the classic MLS and RKPM approaches and their linearized versions produce comparable results. However, in terms of runtime, the classic formulations of MLS and RKPM are expected to be significantly slower than their linearized equivalents. We benchmark their performance using a synthetic scenario consisting of a cube sampled with 1 million particles arranged in a regular cubic lattice. The average timings for evaluating the gradient computation for all particles using the difference formula in Eq. (10) and its corrected versions can be seen in Tab. 1. CMLS is 3.22 times slower than the linearized MLS and CRKPM is 3.61 times slower than the linearized RKPM. Additionally, we remark that there is hardly any runtime difference between KGC, MLS and RKPM both in terms of the matrix computation and the gradient evaluation. In particular, the gradient evaluation is even not significantly more expensive than the standard SPH interpolant.

Overall, our analysis shows that for correcting the linear error all investigated methods perform equally well. This result also corresponds to our mathematical investigation which shows that all corrections work in a similar fashion.

5 APPLICATIONS

Apart from theoretical and mathematical improvements we analyzed in the previous section, we now investigate the visual benefits of correcting the derivative operations in SPH to be first-order correct. We show that correcting the errors arising through the neighborhood deficiency improves visual fidelity in the application examples of boundary handling, viscosity, surface tension, vorticity and elastic solids.

5.1 Boundary Handling

Different boundary handling methods have been proposed for SPH. These include particle-based approaches [Adami et al. 2012; Akinci et al. 2012; Band et al. 2018, 2017] and implicit boundary representations [Bender et al. 2019b; Koschier and Bender 2017]. A popular, particle-based approach is given by Akinci et al. [2012]. Their method samples boundary models and rigid bodies with particles which interact with the fluid by being included into the density and pressure force computation. The authors derive the boundaries contribution to the pressure force as

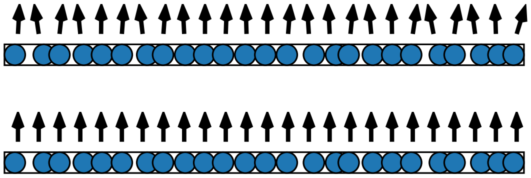


Figure 3: Normal computations of an irregularly sampled boundary. Top: SPH normal computation using Eq. (33). Bottom: Corrected normal computation using MLS.

$$\mathbf{f}^{p}_{i\leftarrow k} = -\Psi_k \frac{p_i}{\rho_i^2} \nabla W_{ik}, \tag{32}$$

where k denotes a boundary particle, $\Psi_k = \rho_0 V_k$ its pseudo-mass and $V_k = 1/\sum_{k'} W_{kk'}$ the boundary particle's volume. While this directly allows for rigid-fluid coupling, using the same error-prone SPH interpolant leads to artifacts in the computation. These errors become apparent when taking a look at the direction of the pressure force contribution, which is indicated by the normal field. As Band et al. [2017] derive from Eq. (32), the normal of the boundary is computed as

$$\mathbf{n}_i = \sum_k \Psi_k \nabla W_{ik}. \tag{33}$$

In a regularly ordered, symmetric scenario, SPH calculates the correct normal using Eq. (33), but even a small misalignment or non-regular particle sampling of the boundary will lead to errors. This behavior can be seen in Fig. 3 which depicts the computation of the normal of an exemplary boundary sampling using Eq. (33). The introduced errors lead to a normal estimate deviating from the correct normal of the boundary and thus create ghost forces leading to drift or bouncing [Bender et al. 2019b; Koschier and Bender 2017]. We now replace the SPH interpolation with the MLS shape function. To account for the contributions of the boundary, we subsequently have to include the boundary particles into the computation of the correction matrix Eq. (24) and the MLS interpolant . As a linear, least-squares method, the MLS interpolation can be treated like fitting a hyperplane through the boundary sampling. Since all particles of the boundary object lie in one plane, the fit exactly equals the surface. Thus, the resulting normal direction also equals the normal of the underlying geometry. To be specific, the accurate normal field in Fig. 3 is constant for any particle on top of it. Since the corrected interpolants have linear consistency, we receive the correct normal estimate.

Applying the shape function in Eq. (14) to Eq. (32) yields

$$\mathbf{f}^{p}_{i\leftarrow k} = -\rho_0 \frac{p_i}{\rho_i^2} \nabla\Phi_k(\mathbf{x}_i). \tag{34}$$

On flat surfaces, this completely eliminates any non-physical drift or jumping introduced by the particle-based sampling.

5.2 Viscosity

To simulate viscous flow, one needs to solve the viscous force in the second term of Eq. (5). There are two state-of-the-art approaches in computing this force. These include either evaluating the divergence of the viscous strain rate tensor, i.e. $\mathbf{a}^{\text{visco}} = 2\nu\nabla\cdot\mathbf{E}$ [Takahashi et al. 2015] or directly computing the Laplacian of the velocity field, i.e. $\mathbf{a}^{\text{visco}} = \nu\nabla^2\mathbf{v}$ [Müller et al. 2003; Weiler et al. 2018]. Weiler et al. [2018] report that the former approach may introduce visual artifacts since the strain rate tensor is often computed with an error-prone SPH formulation.

As an example, we will analyze the implicit, strain rate-based approach of Takahashi et al. [2015]. Their model solves the backward Euler equation for the current velocities advected by the viscous acceleration. Let $\mathbf{v}_i^*$ denote the velocities after applying external forces like gravity, then the implicit system is given by

$$\mathbf{v}_i^{(t+\Delta t)} = \mathbf{v}_i^{*,(t+\Delta t)} + 2\nu\frac{\Delta t}{\rho_i}\nabla\cdot\mathbf{E}_i\left(\mathbf{v}^{(t+\Delta t)}\right), \tag{35}$$

where the viscous stress rate tensor $\mathbf{E}$ is given by Eq. (4). To fully build the linear system, $\nabla\cdot\mathbf{E}$ and $\nabla\mathbf{v}$ must be determined for which the authors employ the symmetric and difference formula, respectively:

$$\nabla\cdot\mathbf{E}_i = \rho_i\sum_{j\in\mathcal{N}_i} m_j\left(\frac{\mathbf{E}_i}{\rho_i^2}+\frac{\mathbf{E}_j}{\rho_j^2}\right)\nabla W_{ij} \tag{36}$$

$$\nabla\mathbf{v}_i = \frac{m_j}{\rho_j}\left(\mathbf{v}_j-\mathbf{v}_i\right)\nabla W_{ij}^T. \tag{37}$$

Both equations are subject to errors at the free surface as we have shown in Sec. 4.1. Using the corrected formulations in Eqs. (16), (15) and applying them on Eqs. (36) and (37) yields

$$\nabla\cdot\mathbf{E}_i = \rho_i\sum_{j\in\mathcal{N}_i} m_j\left(\frac{\rho_j\mathbf{E}_i\nabla\Phi_j\left(\mathbf{x}_i\right)}{m_j\rho_i^2}-\frac{\rho_i\mathbf{E}_j\nabla\Phi_i\left(\mathbf{x}_j\right)}{m_i\rho_j^2}\right) \tag{38}$$

$$\nabla\mathbf{v}_i = \sum_{j\in\mathcal{N}_i}\left(\mathbf{v}_j-\mathbf{v}_i\right)\nabla\Phi_j\left(\mathbf{x}_i\right)^T. \tag{39}$$

Since both formulations are first-order consistent, they eliminate all ghost forces introduced by rotational motion, which makes the resulting model momentum conserving. Additionally, the double smoothing employed by Takahashi's model propagates and amplifies any error present after the first interpolation. Further reducing these errors by the correction reduces numerical artifacts like tearing which have also been reported by Weiler et al. [2018].

5.3 Surface Tension

Another important phenomena to capture while simulating fluids is surface tension. This effect is particularly interesting for simulating fluid droplets, soap bubbles or adhesion effects between a fluid and a solid surface. Since surface tension forces arise near the free surface, the neighborhood deficiency problem also affects their computation using SPH. Many popular approaches use the normal field for calculation [Akinci et al. 2013; He et al. 2014; Morris 2000], which is needed to, e.g., calculate the mean curvature [Müller et al. 2004]. As we have seen in Sec. 5.1, the normal field computation using SPH is particularly error-prone. Here, we will analyze the surface energy approach at the example of the method of He et al. [2014]. This uses the surface tension energy created by the local surface area approximation:

$$E^{\text{surface}} = \int_V \frac{\kappa}{2}\parallel\nabla c\parallel^2 dV, \tag{40}$$

where κ denotes the surface tension coefficient and c the color field, i.e. a field that is 1 inside and 0 outside the fluid domain. The authors propose to compute the color field as the interpolation of the constant field of unity:

$$c_i = \sum_{j\in\mathcal{N}_i}\frac{m_j}{\rho_j}W_{ij}. \tag{41}$$

Note that this representation of the color field relies on the neighborhood deficiency problem. Taking the gradient of it is similar to calculating the normal with Eq. (33) and thus also inherits all of its accompanying issues. Thus, the authors propose to include a renormalization factor in computing the gradient:

$$\nabla c_i = \frac{\sum_{j\in\mathcal{N}_i}\frac{m_j}{\rho_j}c_j\nabla W_{ij}}{\sum_{j\in\mathcal{N}_i}\frac{m_j}{\rho_j}W_{ij}}, \tag{42}$$

which accounts for the missing contributions near the free surface. While this reweighting of the particles helps to adjust for the missing contribution, the direction and the norms are still erroneous. Finally, spatially deriving and discretizing Eq. (40) yields the surface tension force, for which the authors use the following, symmetrized version:

$$\mathbf{f}_i^{\text{ST}} = \frac{\kappa}{2}\sum_{j\in\mathcal{N}_i}\frac{m_j}{\rho_j}\frac{\parallel\nabla c_i\parallel^2+\parallel\nabla c_j\parallel^2}{2}\nabla W_{ij}. \tag{43}$$

In summary, the model requires two SPH gradient operations which are both subject to errors at the free surface. The authors try to mitigate these issues by the inclusion of an atmospheric pressure computation. During the rest of this work, we will exclude this correction and focus on correcting the underlying model by the introduced correction methods.

Similar to the computation of the boundary pressure force, the gradient of the color field can be computed as

$$\nabla c_i = \sum_{j\in\mathcal{N}_i} c_j\nabla\Phi_j\left(\mathbf{x}_i\right). \tag{44}$$

Note that we exclude the renormalization factor used in Eq. (42) since the corrected shape functions already account for missing particles. Thus, we achieve a linear accurate color field gradient which also propagates less errors to the second interpolation. For the force formulation we use Eq. (13) with the averaged norm of the neighboring color field gradients as in Eq. (43)

$$\mathbf{f}_i^{\text{ST}} = \frac{\kappa}{2}\sum_{j\in\mathcal{N}_i}\frac{\|\nabla c_i\|^2+\left\|\nabla c_j\right\|^2}{2}\nabla\Phi_j\left(\mathbf{x}_i\right). \tag{45}$$

These changes enable us to, e.g., recreate a fluid droplet in zero gravity without needing an additional atmospheric pressure since we account for the neighborhood deficiency in the desired formulas.

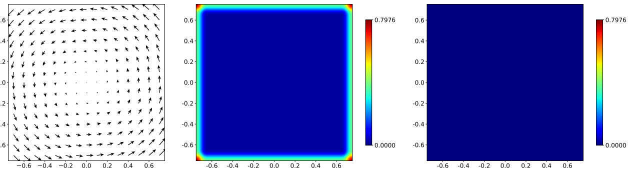


Figure 4: Interpolation errors when computing $\nabla \times \mathrm{v}$ of a linear velocity field. Left: 2D plot of the velocity field $\mathrm{v}^{\text{rot.}}$. Middle: Vector norm of the error between the analytical solution and the results of the difference formula (Eq. (46)). Right: Vector norm of the error between the analytical solution and the results of the corrected difference formula (Eq. (47)).

5.4 Vorticity

Recent advancements in SPH also include the preservation and augmentation of turbulent motion created by unsteady vorticies. These are needed since SPH suffers from numerical diffusion and thus looses vorticity over time. Therefore, models have been proposed that employ an additional, velocity adjustment based on the current vorticity [Liu et al. 2021; Peer and Teschner 2016] or replace the stress tensor of the Navier-Stokes equations with a micropolar one [Bender et al. 2017]. Both models have in common that the vorticity $\boldsymbol{\omega}$ is explicitly computed using SPH:

$$\boldsymbol{\omega}_i = \nabla \times \mathbf{v}_i = \frac{1}{\rho_i} \sum_{j \in \mathcal{N}_i} m_j \left(\mathbf{v}_i - \mathbf{v}_j\right) \times \nabla W_{ij}. \tag{46}$$

Note that the vorticity directly relates to the angular velocity by a scalar factor. As Bender et al. [2019a] denote, Eq. (46) and its corresponding symmetrically discretized formulation using Eq. (11) are subject to errors at the free surface. With the introduced correction methods, we can account for these errors by replacing the shape function with the linear accurate ones, which yields

$$\nabla \times \mathbf{v}_i = \sum_{j \in \mathcal{N}_i} \left(\mathbf{v}_i - \mathbf{v}_j\right) \times \nabla \Phi_i \left(\mathbf{x}_j\right). \tag{47}$$

This is shown in Fig. 4.

5.5 Elastic Solids

Exchanging the Piola-Kirchhoff stress tensor $\mathbf{P}$ for fluids in Eq. (2) by

$$\mathbf{P} = 2\mu\boldsymbol{\varepsilon} + \lambda \operatorname{tr}(\boldsymbol{\varepsilon})\mathbb{I} \tag{48}$$

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\mathbf{F} + \mathbf{F}^T\right) - \mathbb{I}, \tag{49}$$

where ε is the Cauchy strain measure and $\mathbf{F}$ the deformation gradient, yields the formulation for linear elastic materials. This enables us to simulate deformable solids using SPH. However, as we have seen in the case of viscosity in Sec. 5.2 the inability of the SPH interpolant to capture rotational motion introduces the same ghost forces into the simulation of elastic solids. There have been multiple methods proposed for elastic solids. Popular approaches include a corotated material model which allows the usage of a linear material model for better computational efficiency. Since extracting the rotation using solely SPH yields erroneous results, previous methods either compute the rotation using shape matching [Becker et al. 2009] or extract it from the deformation gradient while already applying KGC as a correction method [Kugelstadt et al. 2021; Peer et al. 2018]. We will investigate the latter using the approach of Peer et al. [2018]. In particular, we investigate which errors appear if the correction step is omitted.

In each time step, the authors extract the rotation from the deformation gradient by a polar decomposition and then calculate the elastic force from the linear Cauchy strain measure using an updated, co-rotated deformation gradient. Thus, for the first step the deformation gradient must be calculated from the deviation of the rest configuration $\mathbf{x}^0$ to the current one $\mathbf{x}$

$$\mathbf{F}_i = \sum_{j \in \mathcal{N}_i^0} V_j^0 \left(\mathbf{x}_{ji}\right) \left(\nabla W_{ij}^0\right)^T, \tag{50}$$

where A^0 denotes a property A in rest configuration. Since Eq. (50) cannot capture rigid body rotations, which requires first-order consistency, one can apply our shape functions which yields

$$\mathbf{F}_i = \sum_{j \in \mathcal{N}_i^0} \mathbf{x}_{ji} \nabla \Phi_j^0 \left(\mathbf{x}_i^0\right)^T, \tag{51}$$

from which the rotation matrix $\mathbf{R}$ can be successfully extracted. Next, the evaluation of the strain measure requires the updated deformation gradient $\mathbf{F}_i^*$. A generalization of Peer et al.'s method with the generalized corrected kernel gradients can be determined as

$$\mathbf{F}_i^* = \mathbb{I} + \sum_{j \in \mathcal{N}_i^0} \left(\mathbf{x}_{ji} - \mathbf{R}_i \mathbf{x}_{ji}^0\right) \left(\mathbf{R}_i \nabla \Phi_j^0 \left(\mathbf{x}_i^0\right)\right)^T. \tag{52}$$

With the evaluated strain measure, the Piola-Kirchhoff stress tensor can be evaluated for the final force computation. Inserting Eq. (48) into Eq. (2) yields the elastic body force which the authors discretize in accordance to Ganzenmüller [2015]. Using the corrected kernel gradients yields the general formulation:

$$\mathbf{f}_i^{\text{elastic}} = \sum_{j \in \mathcal{N}_i^0} \left(\mathbf{P}_i \mathbf{R}_i \nabla \Phi_j^0 \left(\mathbf{x}_i^0\right) - \mathbf{P}_j \mathbf{R}_j \nabla \Phi_i^0 \left(\mathbf{x}_j^0\right)\right). \tag{53}$$

6 RESULTS

In this section we will analyze and compare the efficacy of applying any correction algorithm in regard to the neighborhood deficiency problem. Thus, we provide practical insights by showcasing the reduction of various spurious artifacts appearing in the computation of boundary handling, viscosity, surface tension, vorticity and elastic solids. Furthermore, we show that the increased runtime of computing the correction matrices is negligible in comparison to the whole SPH pipeline.

The following example scenes and benchmarks have been simulated on an AMD Ryzen Threadripper PRO 5975WX CPU with 32 cores and 3.60 GHz equipped with 256 GB of RAM. Further, we used the open-source SPH framework SPlisHSPlasH [Bender 2023], and the contained implementations of the previously presented methods, in conjunction with TreeNSearch [Fernández-Fernández et al. 2022] for the neighborhood search. We also optimized the computation of the correction algorithms using OpenMP for parallelization and AVX2 for vectorization. If not otherwise stated, we employed Divergence-Free SPH (DFSPH) [Bender and Koschier 2017] as our pressure solver and the viscosity model of Weiler et al. [2018] for every scene.

Grid Model. To showcase the efficacy of the corrected boundary pressure force in Sec. 5.1, we analyze the trajectories of single fluid particles using Akinci's boundary handling approach [Akinci et al. 2012]. The test scene consists of fluid particles arranged in a 20×20 grid dropping onto an inclined plane which can be seen in Fig. 5. Apart from itself, the neighborhood of each particle consists of only boundary particles. As we have already outlined, the normal computation using the standard SPH interpolant leads to tiny imperfections in the direction of the boundaries normals which manifest themselves in drift and small particle jumps which can be seen in the leftmost image. Applying any of the aforementioned correction algorithms yields a perfect recreation of the flat surface and thus the grid model stays intact while sliding down. Note that we see no difference between KGC, MLS and RKPM in the remaining images since although the correction methods all work slightly different, the resulting plane normals are identical.

Rotating Cubes. In this experiment we showcase the effect of ghost forces caused by the neighborhood deficiency. Thus, we placed a particle-sampled cube in a rotational velocity field and only applied Takahashi et al.'s viscosity model which we analyzed and corrected in Sec. 5.2. Since the cube is advected inside a linear velocity field, any linear errors introduced by the evaluation of the gradient and the divergence operator will cause a force to be applied to the particles. Fig. 6 shows the results of this experiment. As can be seen in the left most image, the uncorrected viscosity model introduces non-physical drift which leads to a disintegration of the cube's shape over time. The corrected versions in the remaining images correctly reconstruct the underlying velocity field. This yields a viscous strain rate which is zero and thus no force is applied during the simulation. As a result for all corrected methods, the cube spins indefinitely while conserving angular momentum.

Buckling. This scene shows the buckling behavior exhibited by highly viscous fluids. We again employ the model of Takahashi et al. [2015]. Note that the authors show that simulating the buckling effect is entirely possible with their method. However, we deliberately chose a set of parameters and input geometry for which this effect fails to be simulated due to the errors arising at the free surface. The buckling is dependent on the ability to fold the highly viscous fluid stream which includes rotational motions. As we mentioned before, these get damped using an uncorrected model. Thus by applying, e.g., KGC we can restore the buckling effect and thus extend the parameter range of Takahashi et al.'s model. A comparison between corrected and uncorrected results can be seen in Fig. 7.

Zero Gravity Droplet. In this experiment we show that we can form a water droplet in zero gravity while using He's surface tension model by solely using corrected SPH interpolants. The original method — as we have discussed in Sec. 5.3 — includes a renormalization of the color field gradient and an atmospheric pressure computation to reduce errors stemming from the free surface. This allows the original authors to also simulate droplets as well as other surface tension effects. However, omitting the additional, atmospheric pressure computation reveals the errors introduced by the underlying SPH operators. Fig. 8 shows the uncorrected solution which indicates numerical errors introduced by the erroneous color field gradient estimation as well as double interpolation. Applying any correction algorithm significantly reduces these errors and thus permits the model to form a sphere, even without atmospheric pressure.

Table 2: Average timings per step for the computation of the washing machine scene. This includes the timings for the neighborhood search T_{ns}, the non-pressure forces T_{np}, the pressure solver T_{p}, the computation of the correction matrices T_{mat} and the total time for a single simulation step T_{total}.

$T_{\text{ns}}\,[ms]$	$T_{\text{np}}\,[ms]$	$T_{\text{p}}\,[ms]$	$T_{\text{mat}}\,[ms]$	$T_{\text{total}}\,[ms]$
8.86	58.52	5.74	0.23	83.19

Spinning Top. While Peer et al. [2018] already show the influence of their co-rotated, corrected kernel function, we are particularly interested in the function of the kernel gradient correction applied throughout the model. Thus, we simulate a spinning top visible in Fig. 9 with and without applying the kernel gradient correction. The results are depicted in Figs. 9a & 9b. When omitting the correction, we notice that the top loses angular momentum and will eventually stop rotating. The reason for this behavior is twofold. Firstly, the deformation gradient computed in Eq. (50) fails to capture the rotational motion near the free surface. This leads to an erroneous, extracted rotation matrix which secondly is propagated to the computation of the updated deformation gradient and the force formulation. Altogether, the resulting ghost forces counteract the spinning motion which violates the conservation of angular momentum. In contrast to the results of the aforementioned investigations of Peer et al. [2018] and also Becker et al. [2009], the rotation of the elastic bodies is not completely hindered since in densely sampled regions, the rotation matrix is still correctly computed.

Washing machine. Finally, we showcase the corrected methods of Peer et al. [2018] and Takahashi et al. [2015] in a coupled multiphase simulation, as seen in Fig. 1. This simulation contains an elastic bunny, viscous soap, water and a rotating washing machine drum. The bunny is dropped into a pool of water and subsequently the washing machine starts rotating while water and soap are continuously added. This shows that it is still possible to use the corrected methods in complex and versatile ways. Furthermore, the additional computational overhead through the matrix computation only accounts for 0.27% of the total runtime for this particular scenario as can be seen in Tab. 2. Note that the computation is heavily dominated by the non-pressure force evaluations. However, we notice that the computation of the correction matrices only take around 4% of the time of the pressure solver. With an average of 3.77 iterations for the constant density and 1 iteration for the divergence-free solver of DFSPH in this scene, we deduce that the overhead of our first-order corrections is negligible in comparison to the full SPH simulation loop.

7 CONCLUSION & FUTURE WORK

We analyzed the benefits of correcting the spatial first-order derivatives using KGC, MLS and RKPM up to linear consistency. In context of the neighborhood deficiency problem, we reduced the ghost

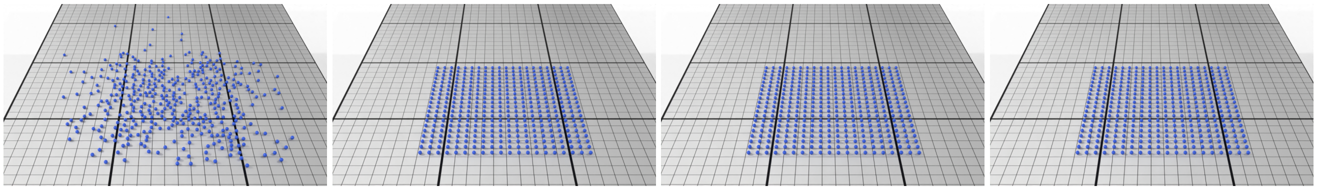


Figure 5: A grid of particles dropped on an inclined plane. Their movement is influenced by the particle-based boundary handling in Eqs. (32) and (34). From left to right, the results are shown using the uncorrected and corrected versions with KGC, MLS and RKPM, respectively. While the corrected versions compute the correct normals, the uncorrected version introduces non-physical drift.

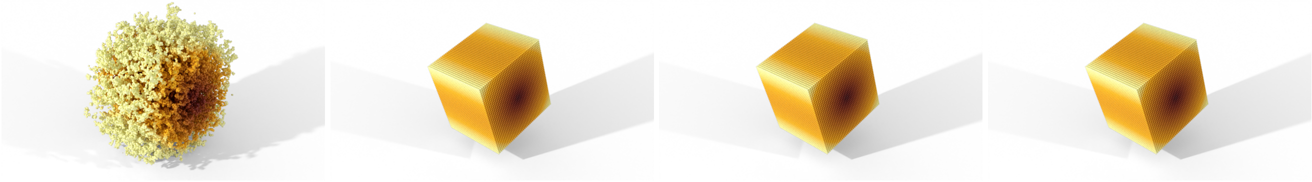


Figure 6: A particle sampled cube placed in an external, rotational velocity field. Only the viscous force model by Takahashi et al. [2015] affects the particle's motion. From left to right: uncorrected, KGC, MLS, RKPM. Only the uncorrected version introduces ghost forces which disintegrate the cube. The particles are shaded according to the velocity magnitude, with increasing values from dark to light.

forces and numerical artifacts which usually appear when using the standard SPH interpolant. Furthermore, we provided a thorough comparison showcasing the similarities between the different approaches. Analyzing the corrections in different synthetic scenarios, we concluded that all methods perform equally well in reducing the errors near the free surface, even for non-linear fields. The application of the methods showed a significant improvement in visual fidelity. Finally, we discovered that the additional overhead of computing the corrected interpolants in comparison to the uncorrected ones is negligibly small.

Since the neighborhood deficiency problem is fundamentally linked to SPH, current and future methods may benefit from adopting a linear consistent formulation. In that regard, our work provides theoretical and practical insights when and how to apply any of the aforementioned correction methods. Compared with replacing the erroneous interpolants with other meshless methods like classic MLS, the presented methods can be easily integrated into the SPH pipeline and do not require substantially more computational effort. However, investigating these methods in the context of higher-order interpolants to further reduce the remaining artifacts appearing at the free surface remains an interesting future research avenue to the graphics community.

ACKNOWLEDGMENTS

The presented investigations were carried out at RWTH Aachen University within the framework of the Collaborative Research Centre SFB1120-236616214 "Bauteilpräzision durch Beherrschung von Schmelze und Erstarrung in Produktionsprozessen" and funded by the Deutsche Forschungsgemeinschaft e.V. (DFG, German Research Foundation). The sponsorship and support is gratefully acknowledged.

Additionally, we would like to thank the Stanford Computer Graphics Laboratory for providing the model of the Stanford Bunny.

REFERENCES

Stefan Adami, XY Hu, and Nikolaus A Adams. 2012. A generalized wall boundary condition for smoothed particle hydrodynamics. *J. Comput. Phys.* 231, 21 (2012), 7057–7075.

Bart Adams and Martin Wicke. 2009. Meshless Approximation Methods and Applications in Physics Based Modeling and Animation. In *Proceedings of the Eurographics conference (EG '09)*. Eurographics Association, 213–239.

Nadir Akinci, Gizem Akinci, and Matthias Teschner. 2013. Versatile surface tension and adhesion for SPH fluids. *ACM Transactions on Graphics* 32, 6 (2013), 1–8. https://doi.org/10.1145/2508363.2508395

Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. 2012. Versatile rigid-fluid coupling for incompressible SPH. *ACM Transactions on Graphics* 31, 4 (July 2012), 1–8. https://doi.org/10.1145/2185520.2335413

Iván Alduán and Miguel A. Otaduy. 2011. SPH granular flow with friction and cohesion. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM Press. https://doi.org/10.1145/2019406.2019410

N. R. Aluru. 2000. A point collocation method based on reproducing kernel approximations. *Internat. J. Numer. Methods Engrg.* 47, 6 (2000), 1083–1121.

Satya N Atluri, JY Cho, and H-G Kim. 1999. Analysis of thin beams, using the meshless local Petrov–Galerkin method, with generalized moving least squares interpolations. *Computational mechanics* 24, 5 (1999), 334–347.

Stefan Band, Christoph Gissler, Andreas Peer, and Matthias Teschner. 2018. MLS pressure boundaries for divergence-free and viscous SPH fluids. *Computers & Graphics* 76 (nov 2018), 37–46. https://doi.org/10.1016/j.cag.2018.08.001

Stefan Band, Christoph Gissler, and Matthias Teschner. 2017. Moving least squares boundaries for SPH fluids. In *Proceedings of the 13th Workshop on Virtual Reality Interactions and Physical Simulations*. Eurographics Association, 21–28.

Markus Becker, Markus Ihmsen, and Matthias Teschner. 2009. Corotated SPH for deformable solids. In *Proceedings of Eurographics Conference on Natural Phenomena*. 27–34. https://doi.org/10.2312EG/DL/conf/EG2009/nph/027-034

Markus Becker and Matthias Teschner. 2007. Weakly compressible SPH for free surface flows. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 1–8. http://portal.acm.org/citation.cfm?id=1272690.1272719{%}5Cnhttp://dl.acm.

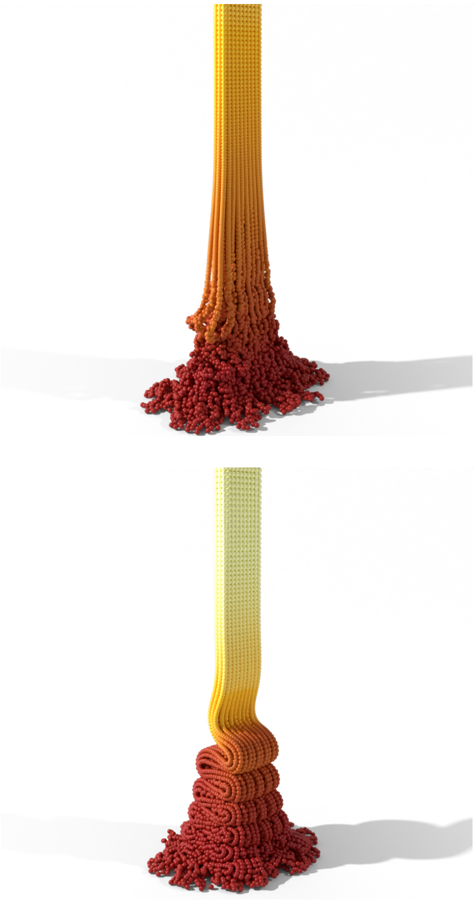


Figure 7: Showcasing the buckling effect of highly viscous fluids using Takahashi et al.'s viscosity model. The fluid on the left fails to buckle given the set of input parameters. Using, e.g., KGC as depicted in the right image, can extend the range of admissible parameters by reducing the errors near the free surface. The particles are shaded according to the velocity magnitude, with increasing values from dark to light.

org/citation.cfm?id=1272719

Ted Belytschko, Yury Krongauz, John Dolbow, and C16509140939 Gerlach. 1998. On the completeness of meshfree particle methods. *Internat. J. Numer. Methods Engrg.* 43, 5 (1998), 785–819.

Jan Bender. 2023. SPlisHSPlasH Library. https://github.com/InteractiveComputerGraphics/SPlisHSPlasH.

Jan Bender and Dan Koschier. 2015. Divergence-Free Smoothed Particle Hydrodynamics. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* 1–9.

Jan Bender and Dan 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

Jan Bender, Dan Koschier, Tassilo Kugelstadt, and Marcel Weiler. 2017. A Micropolar Material Model for Turbulent SPH Fluids. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* 1–8.

Jan Bender, Dan Koschier, Tassilo Kugelstadt, and Marcel Weiler. 2019a. Turbulent Micropolar SPH Fluids with Foam. *IEEE Transactions on Visualization and Computer Graphics* 25, 6 (2019), 2284–2295. https://doi.org/10.1109/TVCG.2018.2832080

Jan Bender, Tassilo Kugelstadt, Marcel Weiler, and Dan Koschier. 2019b. Volume Maps: An Implicit Boundary Representation for SPH. In *Motion, Interaction and Games.* ACM, 26.

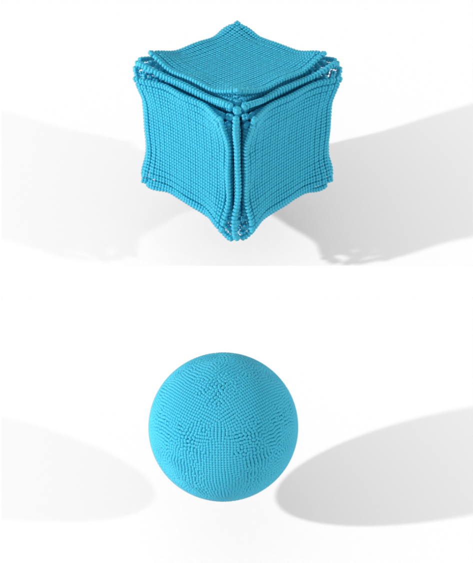
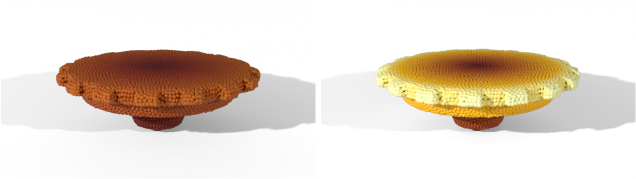


Figure 8: Droplet formation in zero gravity using the surface tension model of He et al. without atmospheric pressure. The uncorrected version on the left suffers from artifacts through the double interpolation and the erroneous normal estimation while the corrected version on the right successfully forms a sphere.



(a) Uncorrected **(b) Corrected**

Figure 9: A spinning top is simulated using Peer et al.'s elasticity model with a high initial angular velocity. Not accounting for the errors estimating the rotation matrix and in the force computation (a) stops the rotation after a short time. Including KGC in the computations (b) lets the top spin indefinitely. The particles are shaded according to the velocity magnitude, with increasing velocity values from dark to light.

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

Xiao-Song Chen, Chen-Feng Li, Geng-Chen Cao, Yun-Tao Jiang, and Shi-Min Hu. 2020. A moving least square reproducing kernel particle method for unified multiphase continuum simulation. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–15.

Mathieu Desbrun and Marie-Paule Gascuel. 1996. Smoothed Particles: A new paradigm for animating highly deformable bodies. In *Eurographics Workshop on Computer Animation and Simulation.* 61–76. https://doi.org/10.1007/978-3-7091-7486-9_5

Gary A. Dilts. 1999. Moving-least-squares-particle hydrodynamics-I. Consistency and stability. *Internat. J. Numer. Methods Engrg.* 44, 8 (1999), 1115–1155. https://doi.org/10.1002/(SICI)1097-0207(19990320)44:8<1115::AID-NME547>3.0.CO;2-L

José Antonio Fernández-Fernández, Lukas Westhofen, Fabian Löschner, Stefan Rhys Jeske, Andreas Longva, and Jan Bender. 2022. Fast Octree Neighborhood Search for

SPH Simulations. *ACM Transactions on Graphics (SIGGRAPH Asia)* 41, 6 (2022), 13.

Thomas-Peter Fries and Hermann G. Matthies. 2004. Classification and Overview of Meshfree Methods. *Informatik-Berichte der Technischen Universität Braunschweig* 2003-03 (2004). https://publikationsserver.tu-braunschweig.de/receive/dbbs_mods_00001418

Georg C Ganzenmüller. 2015. An hourglass control algorithm for Lagrangian smooth particle hydrodynamics. *Computer Methods in Applied Mechanics and Engineering* 286 (2015), 87–106.

R. a. Gingold and J.J. Monaghan. 1977. Smoothed Particle Hydrodynamics: Theory and Application to Non-Spherical Stars. *Monthly Notices of the Royal Astronomical Society* 181 (1977), 375–389. https://doi.org/10.1093/mnras/181.3.375

Christoph Gissler, Andreas Henne, Stefan Band, Andreas Peer, and Matthias Teschner. 2020. An Implicit Compressible SPH Solver for Snow Simulation. *ACM Transactions on Graphics* 39, 4 (Aug. 2020), 1–16.

Xiaowei He, Huamin Wang, Fengjun Zhang, Hongan Wang, Guoping Wang, and Kun Zhou. 2014. Robust Simulation of Sparsely Sampled Thin Features in SPH-Based Free Surface Flows. *ACM Transactions on Graphics* 34, 1 (2014), 7:1–7:9. https://doi.org/10.1145/2682630

Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. 2018. A Moving Least Squares Material Point Method with Displacement Discontinuity and Two-way Rigid Body Coupling. *ACM Transactions on Graphics* 37, 4, Article 150 (July 2018), 14 pages. https://doi.org/10.1145/3197517.3201293

Markus Ihmsen, Jens Cornelis, Barbara Solenthaler, Christopher Horvath, and Matthias 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

Markus Ihmsen, Jens Orthmann, Barbara Solenthaler, Andreas Kolb, and Matthias Teschner. 2014b. SPH Fluids in Computer Graphics. *Eurographics (State of the Art Reports)* (2014), 21–42. https://doi.org/10.2312/egst.20141034

Markus Ihmsen, Arthur Wahl, and Matthias Teschner. 2013. A Lagrangian framework for simulating granular material with high detail. *Computers & Graphics* 37, 7 (nov 2013), 800–808. https://doi.org/10.1016/j.cag.2013.04.010

Johannes C Joubert, Daniel N Wilke, Nicolin Govender, Patrick Pizette, Ugur Tuzun, and Nor-Edine Abriak. 2020. 3D gradient corrected SPH for fully resolved particle–fluid interactions. *Applied Mathematical Modelling* 78 (2020), 816–840.

Dan Koschier and Jan Bender. 2017. Density Maps for Improved SPH Boundary Handling. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 1–10.

Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. 2019. Smoothed Particle Hydrodynamics for Physically-Based Simulation of Fluids and Solids. In *EUROGRAPHICS 2019 Tutorials*. Eurographics Association.

Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. 2022. A Survey on SPH Methods in Computer Graphics. *Computer Graphics Forum* 41, 2 (2022). https://doi.org/10.1111/cgf.14508

Tassilo Kugelstadt, Jan Bender, José Antonio Fernández-Fernández, Stefan Rhys Jeske, Fabian Löschner, and Andreas Longva. 2021. Fast Corotated Elastic SPH Solids with Implicit Zero-Energy Mode Control. *Proc. ACM Comput. Graph. Interact. Tech.* 4, 3, Article 33 (Sept. 2021), 21 pages. https://doi.org/10.1145/3480142

Peter Lancaster and Kes Salkauskas. 1981. Surfaces generated by moving least squares methods. *Mathematics of computation* 37, 155 (1981), 141–158.

Sinuo Liu, Xiaokun Wang, Xiaojuan Ban, Yanrui Xu, Jing Zhou, Jiri Kosinka, and Alexandru C. Telea. 2021. Turbulent Details Simulation for SPH Fluids via Vorticity Refinement. *Computer Graphics Forum* 40, 1 (2021), 54–67. https://doi.org/10.1111/cgf.14095 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14095

Wing Kam Liu, Yijung Chen, R Aziz Uras, and Chin Tang Chang. 1996. Generalized multiple scale reproducing kernel particle methods. *Computer Methods in Applied Mechanics and Engineering* 139, 1-4 (1996), 91–157.

Wing Kam Liu, Sukky Jun, and Yi Fei Zhang. 1995. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids* 20, 8-9 (1995), 1081–1106. https://doi.org/10.1002/fld.1650200824 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.1650200824

Joseph P Morris. 2000. Simulating surface tension with smoothed particle hydrodynamics. *International journal for numerical methods in fluids* 33, 3 (2000), 333–353.

Matthias Müller, David Charypar, and Markus Gross. 2003. Particle-Based Fluid Simulation for Interactive Applications. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 154–159. https://doi.org/citation.cfm?id=846298

M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa. 2004. Point based animation of elastic, plastic and melting objects. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 141. https://doi.org/10.1145/1028523.1028542

Mark Pauly, Richard Keiser, Bart Adams, Philip Dutré, Markus Gross, and Leonidas J Guibas. 2005. Meshless Animation of Fracturing Solids. *ACM Transactions on Graphics* 24, 3 (2005), 957–964. https://doi.org/10.1145/1186822.1073296

Andreas Peer, Christoph Gissler, Stefan Band, and Matthias Teschner. 2018. An Implicit SPH Formulation for Incompressible Linearly Elastic Solids. *Computer Graphics Forum* 37, 6 (2018), 135–148. https://doi.org/10.1111/cgf.13317

A. Peer, M. Ihmsen, J. Cornelis, and M. Teschner. 2015. An Implicit Viscosity Formulation for SPH Fluids. *ACM Transactions on Graphics* 34, 4 (2015), 1–10.

Andreas Peer and Matthias Teschner. 2016. Prescribed Velocity Gradients for Highly Viscous SPH Fluids with Vorticity Diffusion. *IEEE Transactions on Visualization and Computer Graphics* (2016), 1–9. https://doi.org/10.1109/tvcg.2016.2636144

Daniel J. Price. 2012. Smoothed Particle Hydrodynamics and Magnetohydrodynamics. *J. Comput. Phys.* 231, 3 (Feb. 2012), 759–794. https://doi.org/10.1016/j.jcp.2010.12.011

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 (nov 2019), 11 pages. https://doi.org/10.1145/3355089.3356503

Hagit Schechter and Robert Bridson. 2012. Ghost SPH for animating water. *ACM Transactions on Graphics* 31, 4 (2012), 61:1–61:8. https://doi.org/10.1145/2185520.2335412

Donald Shepard. 1968. A Two-Dimensional Interpolation Function for Irregularly-Spaced Data. In *Proceedings of the 1968 23rd ACM National Conference (ACM '68)*. Association for Computing Machinery, New York, NY, USA, 517–524. https://doi.org/10.1145/800186.810616

T. Takahashi, Y. Dobashi, I. Fujishiro, T. Nishita, and M.C. Lin. 2015. Implicit Formulation for SPH-based Viscous Fluids. *Computer Graphics Forum* 34, 2 (2015), 493–502.

Damien Violeau and Thomas Fonty. 2019. Calculating the smoothing error in SPH. *Computers & Fluids* 191 (2019), 104240.

Hui Wang, Yongxu Jin, Anqi Luo, Xubo Yang, and Bo Zhu. 2020. Codimensional Surface Tension Flow Using Moving-Least-Squares Particles. *ACM Trans. Graph.* 39, 4, Article 42 (aug 2020), 16 pages. https://doi.org/10.1145/3386569.3392487

Marcel Weiler, Dan Koschier, Magnus Brand, and Jan Bender. 2018. A Physically Consistent Implicit Viscosity Solver for SPH Fluids. *Computer Graphics Forum* 37, 2 (2018).