Density Maps for Improved SPH Boundary Handling

Dan Koschier
RWTH Aachen University
koschier@cs.rwth-aachen.de

Jan Bender
RWTH Aachen University
bender@cs.rwth-aachen.de

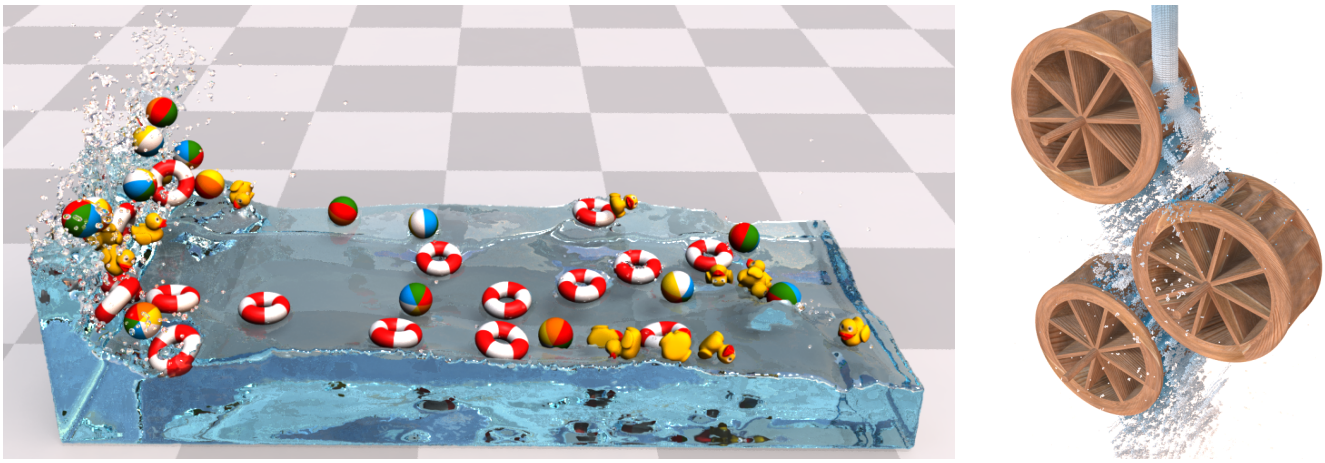


Figure 1: Left: Breaking dam scenario with 45 dynamic rigid bodies interacting with 710k fluid particles. Right: Three water wheels driven by 790k fluid particles.

ABSTRACT

In this paper, we present the novel concept of density maps for robust handling of static and rigid dynamic boundaries in fluid simulations based on Smoothed Particle Hydrodynamics (SPH). In contrast to the vast majority of existing approaches, we use an implicit discretization for a continuous extension of the density field throughout solid boundaries. Using the novel representation we enhance accuracy and efficiency of density and density gradient evaluations in boundary regions by computationally efficient lookups into our density maps. The map is generated in a preprocessing step and discretizes the density contribution in the boundary's near-field. In consequence of the high regularity of the continuous boundary density field, we use cubic Lagrange polynomials on a narrow-band structure of a regular grid for discretization. This strategy not only removes the necessity to sample boundary surfaces with particles but also decouples the particle size from the number of sample points required to represent the boundary. Moreover, it solves the ever-present problem of particle deficiencies near the boundary. In several comparisons we show that the representation is more accurate than particle samplings, especially for smooth curved boundaries. We further demonstrate that our approach robustly handles scenarios with highly complex boundaries and even outperforms one of the most recent sampling based techniques.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SCA '17, July 28-30, 2017, Los Angeles, CA, USA
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5091-4/17/07…$15.00
https://doi.org/10.1145/3099564.3099565

CCS CONCEPTS

• **Computing methodologies → Physical simulation**;

KEYWORDS

Smoothed Particle Hydrodynamics, boundary handling, implicit representation, incompressible fluids

ACM Reference format:
Dan Koschier and Jan Bender. 2017. Density Maps for Improved SPH Boundary Handling. In *Proceedings of SCA '17, Los Angeles, CA, USA, July 28-30, 2017,* 10 pages.
https://doi.org/10.1145/3099564.3099565

1 INTRODUCTION

In recent years, fluid simulations based on Smoothed Particle Hydrodynamics (SPH) have become increasingly popular. While a lot of work has been put into the development of novel models for the simulation of complex materials and efficient pressure solvers to enforce incompressibility, the general concept of discretizing boundary geometries using particle samplings remained largely untouched and is the state-of-the-art.

In this paper we present the concept of density maps; a novel model and discretization based on signed distance fields (SDFs) to represent boundary geometries and to impose boundary conditions with friction. Starting from a continuous representation extending the fluid's density field into the boundary geometry, we discretize

the function using cubic polynomials on a sparse grid without any dependence on particle sampling. The main advantage of our approach is that the higher-order approximation is able to very accurately represent curved boundary surfaces. This results in very smooth trajectories of particles in contact where traditional particle sampled boundaries produce artifacts in the dynamic movement as shown in our results. Even in the rare case of particles strongly penetrating the boundary, our implicit representation throughout the boundary enables us to correctly recover non-penetration. An additional advantage of our approach is that the discretization can be precomputed which allows us to evaluate the underlying integrals with high accuracy. Then, the discretized density map can be queried very efficiently during runtime using a simple interpolation function of constant complexity. Besides the efficiency of the density lookups the performance of neighborhood searches is greatly improved due to the vanished requirement to account for boundary particles. Additionally, our approach resolves the well-known problem of particle deficiencies in the boundary near-field while no particle sampling of the boundary is required. Furthermore, we present an approach for solid-fluid coupling with a novel Coulomb friction model. As the density maps are based on SDFs we can easily compute contact normals and friction tangents using the gradient of the map and can even reuse the SDFs for collision handling of dynamic objects.

In our results, we show that the approach seamlessly integrates into existing SPH pressure solvers. On the basis of several comparisons with particle-based boundary representations, we discuss the advantages of our implicit representation and show that it is able to produce the expected results in scenarios where particle sampled boundaries produce artifacts. We demonstrate that the method is able to robustly handle highly complex scenarios resulting in realistic dynamics (see Figure 1). Moreover, we show that our approach outperforms most recent boundary particle based approaches.

2 RELATED WORK

In recent years, a variety of SPH-based methods has been developed to simulate complex fluid phenomena and solid-fluid interaction. In this section, we will briefly discuss methods related to the presented method. For a general overview over SPH based fluid simulations we would like to refer the reader to the state-of-the-art report of Ihmsen et al. [2014b].

In an early work, Monaghan [1992] developed the first approach to simulate free surface flows with SPH using state equations. Later, the concept of SPH was introduced to the computer graphics community for the simulation of deformables [Desbrun and Gascuel 1996] and fluids [Müller et al. 2003]. In order to simulate incompressible fluids, Becker and Teschner [2007] proposed an equation of state based approach to guarantee a small maximum compression by precomputation of a suitable stiffness coefficient. In the following years, many implicit solvers were developed to enforce incompressibility, either by demanding a divergence-free velocity field or by counteracting particle compression through position corrections or impulses. Raveendran et al. [2011] proposed a hybrid method using SPH and a background grid. While the fluid was represented by particles, a divergence-free velocity field was enforced on the grid. In contrast to enforcing a divergence-free velocity field, Solenthaler and Pajarola [2009] used a predictive-corrective scheme that avoids compression on the position level. A similar, position based approach was presented by Macklin and Müller [2013]. Later, Ihmsen et al. [2014a] discretized the pressure Poisson equation to iteratively correct the density deviations, also enforcing incompressibility on the position level. In contrast to either enforcing a divergence-free velocity field or counteracting compression based on particle positions, a solver satisfying both conditions was proposed by Bender and Koschier [2017].

In order to handle static and dynamic solid boundaries traditional approaches sample the boundary surface using special boundary particles (cf. [1994]). Furthermore, specialized boundary kernel functions are frequently used during the computation of boundary pressure forces as introduced by Becker and Teschner [2007]. In contrast to counteracting boundary penetrations using forces, Becker et al. [2009] proposed a predictor-corrector scheme for boundary handling. However, the method suffers from particle stacking artifacts near the boundary. In order to solve the problem, Ihmsen et al. [2010] presented a similar method that results in smoother density distributions by accounting for the fluid pressure near the boundary. A further improved variant of particle sampled boundaries for two-way coupling was proposed by Akinci et al. [2012]. They additionally compute normalized pseudo masses for the boundary particles to account for non-uniform sampling patterns. A staggered particle approach was proposed by He et al. [2012]. They introduce auxiliary particles to enforce the desired boundary conditions during a pressure projection on a divergence-free velocity field on the staggered structure. Also adaptive sampling approaches for two-way coupling of fluids with deformable solids were developed over the years [Akinci et al. 2013; Müller et al. 2004; Yang et al. 2012].

Eventually, all previously discussed approaches have one thing in common. They all discretize solid boundaries using particles which has the following consequences. Particle samplings of smooth surfaces are only moderately smooth resulting in a 'bumpy' representation. These irregularities bias the dynamics of the fluid and cause an undesired friction effect which can also be interpreted as artificial viscosity. We also demonstrate this effect in our results. Furthermore, boundary particles produce a considerable computational overhead during neighborhood searches, force computations and pressure solves. In contrast, our novel concept of density maps is able to represent smooth surfaces very accurately using higher-order polynomials and improves the efficiency in neighborhood search algorithms and force computations.

Also approaches that avoid the use of particle samplings for boundary handling and two-way coupling were proposed. Harada et al. [2007a; 2007b] present a method for static boundary handling. They construct a particle sampling for a planar example boundary and sample the boundary's density contribution dependent on the distance of a proxy fluid particle. They further construct an SDF for the boundary geometry and access the sampling using the yielded distance values during runtime. However, the approach does not correctly account for non-planar boundaries. Bodin et al. [2012] developed a fluid solver that enforces boundary conditions using unilateral constraints. However, their approach eventually results in a mixed linear complementarity problem (MLCP) which is hard to

solve and computationally intensive as discussed by Erleben [2013]. In order to couple fluids with shells and cloth, Huber et al. [2015] detect contacts using a continuous collision detection with the polygonal surfaces and use impulses and forces to counteract penetrations. In the field of mechanical engineering, Kulasegram et al. [2004] introduced a renormalization factor to improve the SPH approximation close to the boundary. The factor considers the overlapping volume of kernel support and boundary domain. Similar gradient correction terms for weakly compressible fluids were introduced by Ferrand et al. [2013] for two-dimensional and by Mayrhofer et al. [2011] for three-dimensional domains. The renormalization approach was later adopted by Fujisawa and Miura [2015] in the field of computer graphics. They represent boundaries using triangular meshes and compute the renormalization factors during runtime. While they avoid the requirement to sample the boundary geometry with particles, they report of a significant drop in performance due to the factor computation. Even worse, for larger numbers of triangles the computational effort increases resulting in a significantly worse performance compared to sampling approaches. A semi-analytic approach that extends the density field into the boundary was proposed by Monaco et al. [2011]. While they solve the radial portion of the occurring volume integrals analytically, they have to perform a numerical integration over the remaining dimensions during runtime. Very recently, a Moving Least-Squares (MLS) based approach was proposed by Band et al. [2017]. They identify planar regions in a particle sampled boundary and fit a plane into the region using MLS. This improves the density estimate and normal computation resulting in a smoother representation. Unfortunately, their approach does not easily generalize to arbitrary boundary shapes.

In contrast to the discussed approaches, we precompute an accurate implicit representation of an extended boundary density field for arbitrary boundary geometries. Consequently, the boundary density and density gradient contributing to a fluid particle can be efficiently accessed during runtime by performing only a single query into our density map.

3 GOVERNING EQUATIONS AND STANDARD SPH DISCRETIZATION

The dynamic behavior of a wide range of fluids is represented by the incompressible Navier-Stokes equations

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

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

where $\rho, t, \mathbf{v}, p, \nu$ and $\mathbf{f}$ stand for fluid density, time, velocity, pressure, kinematic viscosity and specific external forces, respectively. While the incompressibility condition is represented by Equation (1), Equation (2) describes momentum conservation. An analytical solution to these partial differential equations with arbitrary boundary conditions is unknown. Therefore, the equations have to be discretized in order to compute a numerical solution.

A popular method for spatial discretization of the problem is the SPH formalism. Following the standard approach scalar and vector fields are approximated and rewritten in discrete form as described in the following.

Given a function $f : \Omega \to \mathbb{R}^n$ that maps a position vector $\mathbf{x}$ in the problem domain $\Omega \subset \mathbb{R}^3$ to a scalar or vectorial value, it can be rewritten using the Dirac delta identity and approximated by replacing the delta distribution with a kernel function $W : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ with compact support:

$$\begin{aligned} f(\mathbf{x}) &= \int_\Omega f(\mathbf{x}^*)\, \delta(\mathbf{x}-\mathbf{x}^*) d\mathbf{x}^* \\ &\approx \int_{\mathcal{D}_\mathbf{x}} f(\mathbf{x}^*)\, W(\|\mathbf{x}-\mathbf{x}^*\|, h) d\mathbf{x}^* \\ &= \int_{\mathcal{D}_\mathbf{x}} \frac{f(\mathbf{x}^*)}{\rho(\mathbf{x}^*)}\, W(\|\mathbf{x}-\mathbf{x}^*\|, h)\, \underbrace{\rho(\mathbf{x}^*) d\mathbf{x}^*}_{d\mathbf{m}}, \end{aligned} \tag{3}$$

where h represents the smoothing length of the kernel and $\mathcal{D}_\mathbf{x}$ the spherical domain around $\mathbf{x}$ where $W(\|\mathbf{x}-\mathbf{x}^*\|) > 0$. In the following we will refer to the radius of the domain $\mathcal{D}_\mathbf{x}$ as support radius r. More specifically, the kernel is chosen such that $\lim_{h\to 0} W(\|\mathbf{x}-\mathbf{x}^*\|, h) = \delta(\|\mathbf{x}-\mathbf{x}^*\|)$ subject to normalization $\int_\Omega W(\|\mathbf{x}-\mathbf{x}^*\|, h) d\mathbf{x}^* = 1$. The integral resulting from Equation (3) is then discretized into a sum over particle positions where each particle j carries a portion of the fluid mass m_j:

$$f(\mathbf{x}) \approx \sum_j \frac{m_j}{\rho_j} f(\mathbf{x}_j) W(\|\mathbf{x}-\mathbf{x}_j\|, h).$$

Please note that due to the fact that the kernel function W has compact support only a small subset of the fluid particles located close to the evaluation point $\mathbf{x}$ contribute to the sum. Using efficient algorithms to determine the neighboring particles the number of evaluations to compute the sum can be minimized. When the presented discretization is applied to the density field the formula reduces to

$$\rho(\mathbf{x}) \approx \sum_j m_j W(\|\mathbf{x}-\mathbf{x}_j\|, h). \tag{4}$$

In order to account for solid boundaries the boundary surface is usually sampled using so-called boundary particles which serve as a continuous extension of the density field. Due to the field extension the density of a particle approaching the boundary will increase until the resulting pressure forces stop the particle from penetrating the boundary. Then Equation (4) is extended by a second sum over the contributing density of the boundary particles k:

$$\rho(\mathbf{x}) \approx \sum_j m_j W(\|\mathbf{x}-\mathbf{x}_j\|, h) + \sum_k \tilde{m}_k \tilde{W}(\|\mathbf{x}-\mathbf{x}_k\|, h). \tag{5}$$

Often specialized boundary kernels $\tilde{W}$ or special pseudo-masses $\tilde{m}_k$ are used as presented by Becker and Teschner [2007] and Akinci et al. [2012], respectively. Again, only boundary particles in the close neighborhood of the query point $\mathbf{x}$ are considered due to the compact supports of W and $\tilde{W}$.

Discussion. Representing boundary regions with particle samplings has several disadvantages. The irregularity of the sampling patterns introduces a certain noise in the dynamics of the fluid. This also causes undesired friction between boundary and fluid which can be physically interpreted as artificial viscosity. We demonstrate these effects in Section 6. In order to alleviate the effects of irregularity the boundaries can be oversampled. This, however, results in a considerable computational overhead in neighborhood search

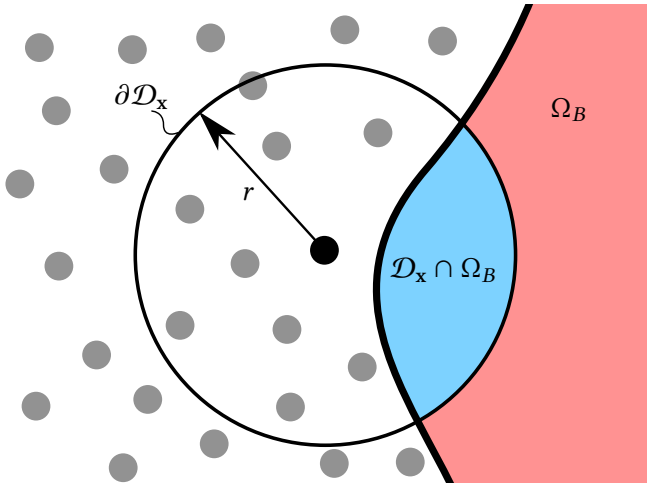


Figure 2: Domains required to generate the density map for a point located in a particle represented by a black dot. While the red area depicts the boundary domain Ω_B the blue area represents the integration domain for the evaluation of the boundary density value at the particle position.

algorithms and during pressure solves and force computations. Further, the detail with which smooth surfaces can be represented is dependent on the particle size so the number of required boundary particles increases when the size of the fluid particles is decreased.

4 CONTINUOUS BOUNDARY MODEL

In this section we will introduce our novel concept of density maps. Recalling the continuous approximation described in Equation (3), we additively decompose the integral by splitting the integration domain into boundary domain Ω_B and fluid domain $\mathcal{D}_x \setminus \Omega_B$ (cf. Figure 2) such that the density function can be rewritten as

$$\rho(\mathbf{x}) = \rho_F(\mathbf{x}) + \rho_B^c(\mathbf{x}).$$

Subsequently, we discretize the integral over the fluid domain into the usual sum resulting in

$$\begin{aligned}\rho_F(\mathbf{x}) &= \int_{\mathcal{D}_x \setminus \Omega_B} W(\|\mathbf{x}-\mathbf{x}^*\|, h) d\mathbf{m} \\ &\approx \sum_j m_j W(\|\mathbf{x}-\mathbf{x}_j\|, h).\end{aligned}$$

The basis for the boundary model is a virtual extension of the density field into solid boundaries. Given a boundary object $\mathcal{B}$ the boundary density contribution in continuous form ρ_B^c is defined by

$$\rho_B^c(\mathbf{x}) = \int_{\mathcal{D}_{\mathbf{x}} \cap \Omega_B} \gamma(\Phi(\mathbf{x}^*))\, W(\|\mathbf{x}-\mathbf{x}^*\|, h) d\mathbf{x}^* \tag{6}$$

$$\Phi(\mathbf{x}) = s(\mathbf{x}) \inf_{\mathbf{x}^* \in \partial\mathcal{B}} \|\mathbf{x}-\mathbf{x}^*\| \text{ with } s(\mathbf{x}) = \begin{cases} -1 & \text{if } \mathbf{x} \in \mathcal{B} \\ +1 & \text{otherwise} \end{cases} \tag{7}$$

$$\Omega_B = \left\{ \mathbf{x} \in \mathbb{R}^3 \mid \gamma(\Phi(\mathbf{x})) \neq 0 \right\}.$$

Here $\gamma : \mathbb{R} \to \mathbb{R}^+$ is a (non-strictly) monotonically decreasing function which yields higher values for higher penetration depth. The signed distance represented by $\Phi : \mathbb{R}^3 \to \mathbb{R}$ is then passed as an argument to γ. From a physical point of view this means that if a particle approaches or penetrates the boundary surface, the basis density function γ yields increasing non-negative values. Finally, we model the resulting density field as convolution of γ with the SPH kernel W in order to fit the smoothed discretization. While a choice of γ as constant function is generally possible, we rather model the boundary's basis density field as continuous linear function

$$\gamma(x) = \begin{cases} \rho_0 \left(1 - \frac{x}{r}\right) & \text{if } x \leq r \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

The function's domain is chosen slightly larger than the boundary object $\mathcal{B}$ and equals the rest density ρ_0 on the boundary surface. The reason for choosing a linear function is twofold. Firstly, a piecewise constant choice of γ would result in a discontinuity that makes the integral in Equation (6) much harder to evaluate numerically. Secondly, in the rare case that a particle strongly penetrates the boundary, the linear model enforces that the gradient direction of the function still points outwards in order to resolve the penetration. Obviously, the support of $\gamma(\Phi)$ on the domain Ω_B is enlarged by the kernel support radius r. The choice for defining γ on this extended boundary domain Ω_B rather than directly on $\mathcal{B}$ is motivated by practical considerations. As we visualize particles using spheres or using a reconstructed polygonal surface with a certain offset, this buffer ensures that the fluid never visually penetrates the boundary surface. Also an interesting observation is that the sampling based boundary handling approach proposed by Akinci et al. [2012] can be interpreted as particle discretization of our model with $\gamma(s) = \rho_0$.

The final question that remains is: How can we accurately and efficiently compute the function values of ρ_B^c described by the integral in Equation (6)? Not only does this involve computing a numerical solution to an integral but also evaluating the signed distance to the boundary's surface. Generally, an evaluation during runtime, e.g. using regular sampling, quadrature or Monte-Carlo sampling methods, is possible. However, an accurate numerical computation of the integral can be computationally costly as this may require a large number of function evaluations. Therefore, one either has to sacrifice accuracy for computational efficiency or vice versa.

A first step in the course of avoiding the runtime computation is the following important observation. For static boundaries ρ_B^c is, in terms of runtime variables, solely dependent on the query point $\mathbf{x}$. This also holds for rigid dynamic boundaries as the function can simply be evaluated at $\mathbf{R}^T(\mathbf{x}-\mathbf{c})$, where $\mathbf{R}$ and $\mathbf{c}$ represent the rotation of the body and its center of mass' translation, respectively. This means the correct density can be computed using a simple linear-affine transformation of the query point $\mathbf{x}$ into the rigid body's reference space and allows us to evaluate the contributing boundary density at any point in the boundary's reference space prior to the simulation. Consequently, this renders the possibility to precompute a highly accurate representation of the boundary density field. This will also allow us to perform very efficient boundary density queries during runtime.

5 DISCRETE DENSITY MAPS

The purpose of this section is to develop a suitable discretization and construction algorithm for Equation (6).

We suggest to discretize the function using cubic polynomials of Serendipity type on a regular hexahedral grid for the following reasons. As discussed by Jones et al. [2006] signed distance functions are continuous everywhere. Further, the convolution of the boundary's basis density field $\gamma(\Phi(\mathbf{x}))$ with the (usually at least cubic polynomial) SPH kernel enforces a certain smoothness of ρ_B^c. The regularity of high-order polynomials is therefore well-suited to discretize the smooth function with high accuracy. An obvious choice for C^0 continuous polynomial discretization is a tensor-product space of Lagrange polynomials. Due to the regular nodal pattern implied by the basis, the approximation can be easily constructed by simply sampling the underlying field at the node pattern. However, in three dimensions a single cubic element requires 64 nodes compared to an 8-node linear element. In order to reduce the number of nodes we use cubic Serendipity type elements. As also discussed by Arnold and Awanou [2011], this element type only requires 32 nodes and is able to maintain C^0 continuity while it has the same approximation order. A quantitative evaluation for an example boundary density field that demonstrates the superiority of the cubic Serendipity element compared to a sampling approach with trilinear interpolation is presented and discussed in Section 6. For a general introduction and discussion on Serendipity type elements for discretization we would like to refer the reader to the works of Zienkiewicz et al. [2013] and Arnold and Awanou [2011].

As previously discussed, evaluations of the boundary density can be computationally intensive since they involve the evaluation of a signed distance to a possibly high-resolution polygonal surface and numerical integration to compute the convolution with the kernel function. In order to optimize evaluations we split the discretization process into three major steps:

(1) Discretization of the signed distance function Φ,
(2) Identification of grid cells close to the boundary resulting in a sparse, narrow-band pattern,
(3) Discretization of the boundary density field ρ_B^c on the sparse grid.

We would like to stress the fact that all steps are part of the preprocessing leading to very efficient lookups in the course of the simulation. In the following paragraphs, each step will be described in detail.

Signed Distance Field Generation. In general, a signed distance field is either represented by analytic implicit functions or can be generated from explicit geometric representations. Based on the fact that complex geometries are usually represented by polygonal meshes, we will describe the algorithm to discretize the induced SDF using cubic Lagrange polynomials on a regular hexahedral grid.

Under the assumption that the input meshes are two-manifold, we first compute the unsigned distance to the mesh and determine the sign using the pseudo-normal test proposed by Bærentzen and Aanæs [2005]. In order to improve the performance of the signed distance computation, we construct a bounding sphere hierarchy beforehand and traverse the structure accordingly.

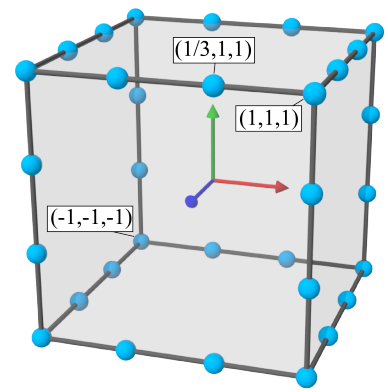


Figure 3: Cubic 32-node element of Serendipity type in isoparametric space.

As discussed earlier, we use isoparametric shape functions of Serendipity type with 32 nodes (cf. Figure 3) and sample the exact signed distance on each node in world space. We can then use the values $\Phi_i = \Phi(\mathbf{x}_i)$ according to their sample points on the grid to cubically interpolate values within each cell using the according shape functions defined as follows:

- **Corner node** ($\xi_i = \pm 1,\ \eta_i = \pm 1,\ \zeta_i = \pm 1$) :

$$N_i = \frac{1}{64}(1+\xi_i\xi)(1+\eta_i\eta)(1+\zeta_i\zeta)\left[9(\xi^2+\eta^2+\zeta^2)-19\right] \quad (9)$$

- **Representative edge node** ($\xi_i = \pm\frac{1}{3},\ \eta_i = \pm 1,\ \zeta_i = \pm 1$) :

$$N_i = \frac{9}{64}(1-\xi^2)(1+9\xi_i\xi)(1+\eta_i\eta)(1+\zeta_i\zeta), \quad (10)$$

where $\boldsymbol{\xi}_i = (\xi_i, \eta_i, \zeta_i)^T$ represents the nodal positions in isoparametric space (cf. [Zienkiewicz et al. 2013]). Here, Equation (10) describes the shape functions on a representative edge that is aligned in ξ-direction. The remaining edge nodes and according shape functions can easily be determined by permutation of coefficients ξ_i, η_i and ζ_i and coordinates ξ, η and ζ. Finally, the interpolation is described by

$$\Phi(\mathbf{x}) \approx \sum_{i=1}^{32} N_i(T(\mathbf{x}))\Phi_i$$
$$T(\mathbf{x}) = \left[\mathrm{diag}(\mathbf{b}-\mathbf{a})\right]^{-1}\left(2\mathbf{x}-(\mathbf{b}+\mathbf{a})\right),$$

where $\mathbf{a}$ and $\mathbf{b}$ are the hexahedral cell's minimum and maximum bound vectors and where $T : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ maps a coordinate in world space to a coordinate in the element's isoparametric space.

Identification of Sparse Structure. As discussed earlier, the density field according to Equations (6) and (8) for signed distances of $\Phi > 2r$ vanishes. Moreover, we assume that particles will not penetrate the boundary more than three times their smoothing length. In fact, in none of our experiments a particle the boundary at all. Consequently, we use the discretized distance field to identify unnecessary cells. This greatly reduces the amount of memory required to store the discretized fields and avoids unnecessary interpolations in far away cells. Based on the previous observation

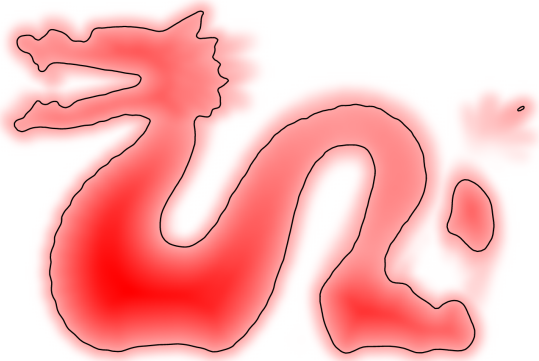


Figure 4: 2D slice of a 3D density map. The boundary density contribution is visualized using a linear scale from white to red. The black contour represents the object surface intersecting the visualized slice.

and assumption we discard all cells that do not meet the following criterion

$$\begin{cases} \text{keep cell} & \text{if } 2r > \Phi_{\min} \ \wedge \ \Phi_{\max} > -r \\ \text{discard cell} & \text{otherwise,} \end{cases}$$

where $\Phi_{\min}$ and $\Phi_{\max}$ are the minimum and maximum signed distance from the cell to the boundary surface. Although we never experienced a particle penetrating the boundary, we formulated the criterion such that cells located within the radius r around the zero contour are also kept as safety buffer.

Boundary Density Field Evaluation and Discretization. In the third and final step, we evaluate the boundary density function given by Equation (6) at each node in the remaining sparse structure. As this is all part of the precomputation, we can invest all our computational resources to compute numerical solutions to the integral with very high accuracy. We have implemented an adaptive Gauss-Kronrod quadrature algorithm that efficiently estimates the residual during computation using a nested quadrature rule (cf. [Piessens and Branders 1974]). However, the usage of any quadrature method such as Monte-Carlo or naive sampling etc. is possible. After determination of all nodal values we can again interpolate values at arbitrary positions within each cell using the functions described in Equations (9) and (10). A graphical illustration of a density map for the Stanford dragon is given in Figure 4. Please note that this image visualizes a 2D slice of a 3D density map. Therefore, also points located in close proximity to the surface in depth direction contribute to the field, e.g. in the region around the tip of the dragons tail.

5.1 Incorporation into Existing Pressure Solvers

Many types of fluids we encounter in daily life are nearly incompressible. Therefore, many implicit pressure solvers for the simulation of incompressible fluids were proposed in recent years, e.g. [Bender and Koschier 2017; Ihmsen et al. 2014a; Macklin and Müller 2013; Solenthaler and Pajarola 2009]. In this section we will discuss how our density maps can be incorporated into existing solvers.

Density and Density Gradient. All pressure solvers typically have one thing in common. They all rely on density and density gradient evaluations in order to project the velocities onto a divergence-free field and/or to project the particle positions onto an uncompressed state. Both quantities can then be evaluated using

$$\rho_B^c(\mathbf{x}) \approx \rho_B(\mathbf{x}) = \sum_{i=1}^{32} N_i(T(\mathbf{x}))\rho_{B,i} \tag{11}$$

$$\nabla_{\mathbf{x}}\rho_B^c\big|_{\mathbf{x}} \approx \nabla_{\mathbf{x}}\rho_B\big|_{\mathbf{x}} = \sum_{i=1}^{32} \nabla N_i\big|_{\xi=T(\mathbf{x})}\,\rho_{B,i}, \tag{12}$$

where $\rho_{B,i} = \rho_B^c(\xi_i)$ are the boundary density values, evaluated at the grid nodes of the hexahedral cell containing $\mathbf{x}$.

Density Advection. The mentioned pressure solvers further rely on a density evaluation after an advection based on a candidate velocity field. Solvers of predictive-corrective nature (cf. [Macklin and Müller 2013; Solenthaler and Pajarola 2009]) actually perform the advection to get a candidate position for each particle and reevaluate the density. However, they often ignore the fact that the previously computed neighborhood information is invalidated by the advection and avoid recomputing the neighborhood information. Besides the problem that the neighborhood information for fluid particles is invalidated this also holds for boundary particles in boundary sampling based approaches. Fortunately, our approach results in the exact result of the boundary density contribution after advections as our method does not rely on boundary particles.

Other pressure solvers (cf. [Bender and Koschier 2017; Ihmsen et al. 2014a]) approximate the advected density at a particle i using a Taylor series of first order in time

$$\rho(\mathbf{x}_i(t+\Delta t)) \approx \rho(\mathbf{x}_i(t)) + \frac{D\rho(\mathbf{x}_i(t))}{Dt}\,\Delta t. \tag{13}$$

Generally, this strategy is very convenient as it circumvents the problem of invalidated neighborhood information at the expense of a loss in accuracy due to the first order approximation. However, when density maps are used an actual advection of the boundary density contribution is both computationally cheap and very accurate. Therefore, we reformulate the estimation by splitting the density into fluid and boundary density

$$\begin{aligned} \rho(\mathbf{x}_i(t+\Delta t)) &= \rho_F(\mathbf{x}_i(t+\Delta t)) + \rho_B(\mathbf{x}_i(t+\Delta t)) \\ &\approx \rho_F(\mathbf{x}_i(t)) + \frac{D\rho_F(\mathbf{x}_i(t))}{Dt}\,\Delta t + \rho_B(\mathbf{x}_i(t+\Delta t)). \end{aligned} \tag{14}$$

When we aim to evaluate $\rho_B(\mathbf{x}_i(t+\Delta t))$ for a particle i, we simply advect the particle in a prediction step and access the density map yielding the desired result. If rigid dynamic boundaries are involved, we also have to account for their movement. In order to do so, we compute the position and orientation of the body at time $t+\Delta t$ in the prediction step by time integration prior to the evaluation of $\rho_B(\mathbf{x}_i(t+\Delta t))$.

5.2 Two-way Coupling with Dynamic Boundaries and Friction

Over the course of the simulation, the fluid interacts with dynamic boundaries exerting forces on the fluid and vice-versa. While the influence of the rigid bodies on the fluid comes naturally due to

their movement and spatially changing position of their underlying density maps, we additionally have to account for the forces exerted by the fluid on the rigid objects. Following Newton's third law, we simply apply the negative force that the boundary exerted on the particle on the dynamic boundary object in order to ensure conservation of momentum. Finally, the force a particle i exerts on a rigid object B at the point of contact is

$$\mathbf{f}_{B\leftarrow i} = m_i \frac{p_i}{\rho_i^2} \nabla_{\mathbf{x}_i} \rho_B \big|_{\mathbf{x}_i}.$$

5.3 Friction

Usually fluid friction is modeled as viscous force for energy dissipation near the boundary (cf. [Akinci et al. 2012; Becker and Teschner 2007]). In this case, however, the friction force is not necessarily acting tangential to the boundary surface. Therefore, we introduce a friction model inspired by Coulomb friction using the density map information in order to apply forces acting exactly tangential to the boundary surface. The friction force counteracting the slip of a particle i is then defined by

$$\mathbf{f}_{i,f} = -\mu\, m_i \frac{p_i}{\rho_i^2} \|\nabla_{\mathbf{x}_i} \rho_B\|\, \mathbf{t}_i \tag{15}$$

with tangent vector $\mathbf{t}_i = \Delta\mathbf{v}_i - (\Delta\mathbf{v}_i \cdot \mathbf{n}_i)\mathbf{n}_i$ and normal vector $\mathbf{n}_i = \nabla_{\mathbf{x}_i}\rho_B / \|\nabla_{\mathbf{x}_i}\rho_B\|$, where $\mu > 0$ is the friction coefficient and $\Delta\mathbf{v}_i = \mathbf{v}_i - \mathbf{v}_B$ is the relative velocity between the fluid particle and the boundary. For a dynamic rigid body with velocity $\mathbf{v}$ and angular velocity $\boldsymbol{\omega}$ the boundary velocity is determined by $\mathbf{v}_B = \mathbf{v} + (\boldsymbol{\omega} \times (\mathbf{x}_B - \mathbf{c}))$, where $\mathbf{x}_B$ denotes the point of contact. However, in order to avoid that a particle changes its tangential direction due to $\mathbf{f}_{i,f}$ we finally use the following definition of the friction force

$$\mathbf{f}'_{i,f} = \begin{cases} \mathbf{f}_{i,f} & \text{if} \quad -\frac{\Delta t}{m_i}\mathbf{f}_{i,f} \cdot \mathbf{t}_i < \Delta\mathbf{v}_i \cdot \mathbf{t}_i \\ -\frac{m_i}{\Delta t}(\Delta\mathbf{v}_i \cdot \mathbf{t}_i)\mathbf{t}_i & \text{otherwise.} \end{cases} \tag{16}$$

Thanks to our implicit boundary representation, we can efficiently reuse the density map's gradient to compute the force. Analogously to the application of pressure forces on the dynamic boundaries, we apply the negative friction force to the dynamic object in contact.

6 RESULTS AND DISCUSSION

For the experiments in this section we integrated our boundary handling into two state-of-the-art SPH methods: DFSPH [Bender and Koschier 2017] and IISPH [Ihmsen et al. 2014a], and used the cubic spline kernel [Monaghan 1992]. Following the suggestion of the corresponding authors we enforced an average compression of less than 0.01 % and a divergence-error of less than 0.1 % (only DFSPH). Moreover, we use the parallel method of Ihmsen et al. [2011] to compute particle neighborhood information and XSPH to account for viscosity [Schechter and Bridson 2012]. For the simulation of rigid bodies we integrated the open-source library PositionBased-Dynamics [Bender 2017] in our simulator and reused the generated sparse SDFs for collision detection. All measurements in this section were performed on two Intel Xeon E5-2697 processors with 2.7 GHz, 12 cores per processor and 64GB RAM.

Comparisons with Particle Sampled Boundaries. We compared the behavior and performance of our approach with the sampling based approach of Akinci et al. [2012]. In our first experiments we demonstrate that our density maps accurately represent boundary surfaces while sampling based approaches may lead to artifacts and bias the resulting dynamics. The particle boundaries were generated using the Poisson-disk sampling algorithm with blue noise properties described in [Corsini et al. 2012]. Naturally, other sampling algorithms can be used instead. Nevertheless, the general concept of particle sampling in three dimensions is inextricably linked to the irregularities in the boundary representation.

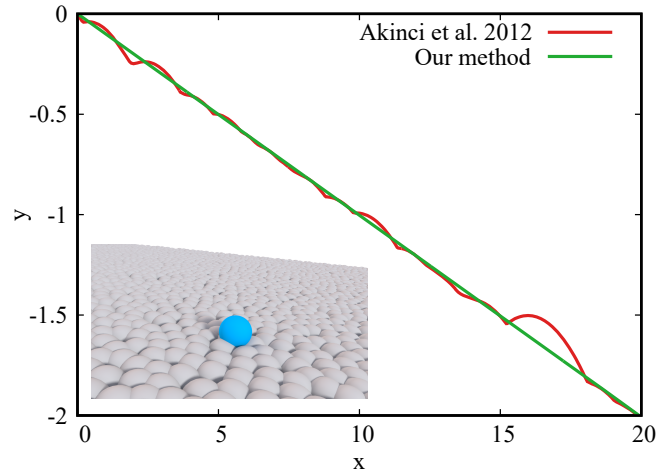
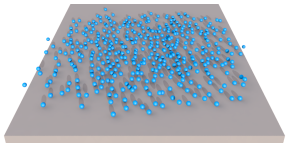


Figure 5: Trajectory of a single fluid particle flowing down an inclined plane. The used particle sampling with the sliding particle is depicted in the lower left corner.

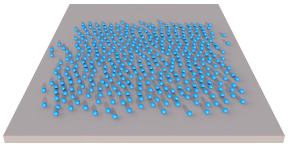
We dropped a single fluid particle on an inclined plane, comparing the particle trajectory yielded by our method to the particle based boundary handling approach of Akinci et al. [2012]. The trajectories of the fluid particle are shown in Figure 5. The graphs clearly show that the particle jumps several times for the particle-sampled boundary representation while our method lets the particle follow a smooth trajectory.

In a similar scenario depicted in Figure 6 we sampled an inclined plane with three different resolutions. While we used SPH particles with kernel support radius $r = 0.8$ the planes were sampled with sampling distances 0.2, 0.1 and 0.05, respectively. The fourth plane was discretized using our approach with cell size 0.2. Then, a grid of fluid particles is dropped on each individual plane such that the particles do not influence each other as the incompressibility condition is initially fulfilled. The trajectories of the fluid particles sliding on the sampled boundary are strongly biased resulting in a chaotic movement (cf. Figure 6a). In the case of a moderately or even strongly oversampled boundary the effect is alleviated. However, the irregularity of the sampling still considerably influences the particle dynamics (cf. Figures 6b and 6c). Finally, using our density map representation the particles linearly slide down the slope as expected (cf. Figure 6d). We would further like to mention that dense samplings lead to a larger number of particles in each particle neighborhood. This again leads to a significant computational overhead in the neighborhood search.

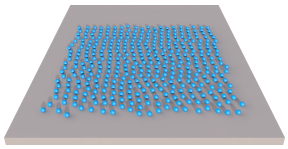
In a third and final comparison a fluid ($r = 0.2$) is trapped inside a spherical boundary with radius 3. Additionally, the boundary



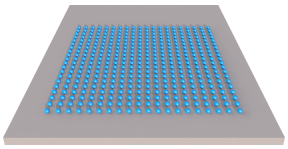
(a) Standard sampling with sampling distance 0.2 and ~69k boundary particles.



(b) Dense sampling with sampling distance 0.1 and ~286k boundary particles.



(c) Very dense sampling with sampling distance 0.05 and ~1.14M boundary particles.



(d) Density map.

Figure 6: A grid of particles slides down an inclined plane. The plane was either discretized with particles or using our approach. Our method maintains the grid structure while the boundary samplings bias the particle trajectories.

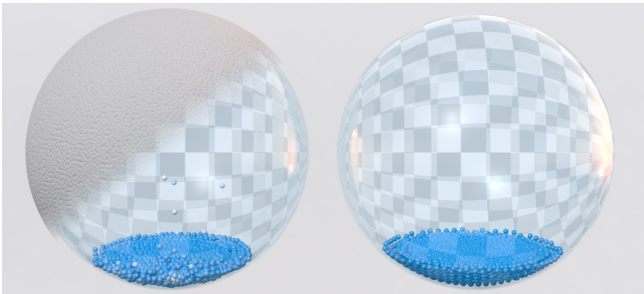


Figure 7: Comparison of frictionless fluid simulations inside rotating spheres with sampled (left) and implicit boundary (right). The sampling pattern is partially illustrated in the left image. Independent of the resolution, particles never rest due to the irregularity of the boundary while the fluid handled by the implicit boundary rests perfectly.

is rotating along the x-axis with constant angular velocity while the contact between fluid and boundary is frictionless. Again, we compared a particle sampled boundary representation with our density map approach with 100^3 grid cells. Due to the rotational invariance of the boundary geometry the rotation of the sphere should not influence the fluid. As illustrated in Figure 7 our representation perfectly fulfills this expectation while the fluid inside the particle based boundary never rests due to oscillations induced by the uneven boundary. Even worse, sometimes particles started to jump due to the irregularity of the sampling.

Complex Scenarios. We simulated a breaking dam scenario using IISPH with 2.1 million particles in a box with rounded corners and 25 static Stanford armadillos as obstacles in the first scenario (cf. Figure 8). The density map of each armadillo was discretized by $100 \times 150 \times 100$ cells. However, the simulation required only one

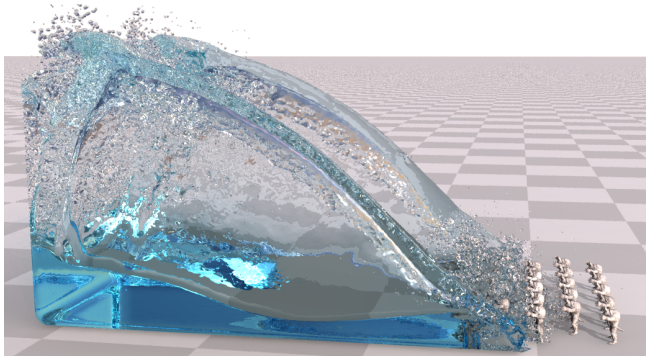


Figure 8: Breaking dam scenario consisting of 2.1 million fluid particles and 25 armadillos as static obstacles in a box.

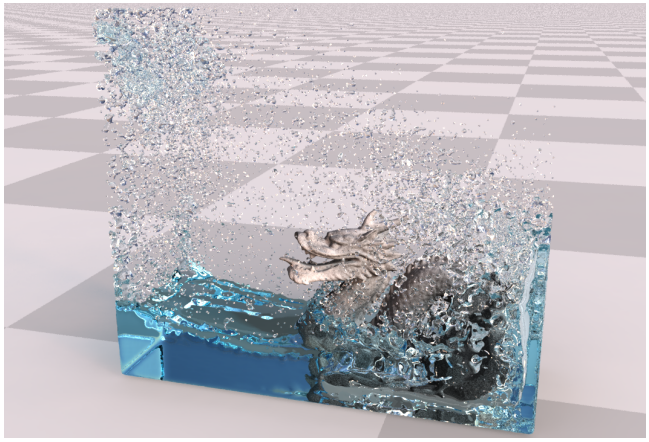


Figure 9: Benchmark scenario with a dragon obstacle.

instance of the discretization as it was reused for all instances of the object. The result shows that our method is able to robustly handle large scenarios with millions of particles.

In the second scenario we compared the performance of a fluid simulation with our method to the method of Akinci et al. [2012] using DFSPH. We used a breaking dam scenario with a static Stanford dragon as obstacle (see Figure 9). The fluid consisted of 211k particles while the obstacle was either sampled using 586k particles or discretized into $200 \times 160 \times 100$ cells. The average time step using the method of Akinci et al. took ~355ms of which the neighborhood search took ~57ms on average. A simulation using our method outperformed the previous one resulting in an average time step of ~241ms with ~33ms for the neighborhood search. On the basis of these results, we achieved a speed-up of factor 1.5. Moreover, the lack of boundary particles in our approach sped up the neighborhood search by approximately factor 2.

In a further experiment we simulated three water wheels driven by 790k fluid particles using DFSPH where the wheels were discretized using $160 \times 160 \times 64$ cells (cf. Figure 1, right). The wheels were constrained using hinge joints resulting in a complex mechanical scenario. The result shows how the two-way coupling correctly accelerates the wheels resulting in a steady and realistic movement.

Figure 10: Hollow spheres with rubber ducks and 510k fluid particles are rolling in a half-pipe with different friction coefficients: $\mu = 0$ (blue), $\mu = 250$ (green) and $\mu = 500$ (red).

In the fourth scenario illustrated in Figure 1 (left), we simulated a breaking dam scenario with 45 dynamic rigid objects of different shapes and 710k fluid particles using DFSPH without friction. The result demonstrates that our method greatly deals with several floating objects interacting and colliding with each other while being in contact with the fluid.

The sixth experiment was simulated using IISPH and shows how three hollow spheres containing rubber ducks and fluids consisting of 510k particles in total with different friction coefficients of $\mu = 0$ (blue), $\mu = 250$ (green) and $\mu = 500$ (red) roll in a half-pipe. Please note that we used the same friction coefficient for contacts between the duck and the sphere in all three simulations. In this result one can see, that the blue fluid remains mostly on the ground of the sphere and does not react to its rotation. In contrast, the green fluid gets influenced by the inner surface of the hollow sphere following its rotation. The red fluid also shows the mentioned effect and even flips the rubber duck on its back after a few seconds. Another physically meaningful effect is that the rolling movement of the spheres slows down earlier due to the energy dissipation.

In the final scene we simulated a ball floating on water ($\rho_0 = 1000$). The ball initially had a density of 100. In the course of the simulation we doubled the density of the ball every three seconds. As illustrated in Figure 11 the ball initially lightly floats on the fluid surface. The gradually increased density lets the floating height decrease. When the density finally exceeds the rest density of the fluid, the ball sinks to the ground, as expected.

Linear vs. Cubic Density Maps. To demonstrate the benefit of using cubic polynomials of Serendipity type instead of regular sampling with trilinear interpolation, we computed a high-resolution reference solution for a circular boundary with radius 1 on a two-dimensional domain for particle support radius $r = 0.1$. We generated two discretizations of the function. One using regular sampling and the other one using our cubic polynomials. We further used approximately the same number of nodes in both grids. For the regular sampling we used a grid of 101×101 cells inheriting 10404 nodes while the cubic field was constructed for a grid of 45×45 cells

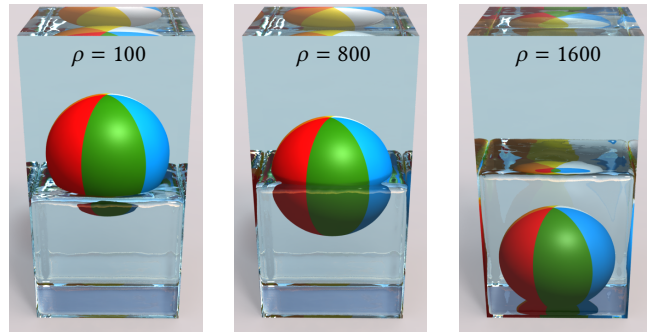


Figure 11: Solid ball with time-varying density interacts with water ($\rho_0 = 1000$).

with 10396 nodes. In order to evaluate the quality of the discretizations we computed the error to the reference solution relative to the reference density $\epsilon = |\rho_B - \rho_B^c|/\rho_0$. A plot of the error functions for both discretizations is depicted in Figure 12. Further, the average error for the regular sampling was 1.06×10^{-2}. The average error for the cubic field was 3.04×10^{-3} and therefore nearly one order of magnitude lower than the error of the linear discretization.

7 CONCLUSION

In this paper we presented the novel concept of density maps; a method to simulate static and dynamic boundaries with friction in SPH simulations. Based on signed distance information, we introduced a continuous formulation that extends the fluid's density throughout the boundary. The function was then discretized using cubic polynomials on a sparse hexahedral grid in a precomputation step. This higher order representation is able to very accurately represent curved surfaces. Even in cases of strong boundary penetrations, the involved particles can be correctly recovered. Moreover, the evaluation of the boundary density contribution using the interpolation function of the grid cells is computationally very efficient compared to approaches based on particle sampled boundaries. In our results, we demonstrated that the method is more accurate and produces no artifacts compared to sampling based techniques. It also maintains stability even in highly complex scenarios and outperforms one of the most recent sampling based approaches.

Like every other method our approach has some limitations. Models with high frequency or sharp features are smoothed out due to our cubic polynomials. Therefore, we would like to use an adaptive discretization method in the spirit of [Koschier et al. 2016]. Another limitation is that the density maps cannot be precomputed for deformables, so further research on possibilites to perform updates could be beneficial. Also, we did not consider no-slip boundary conditions. However, we are confident that these conditions can be easily imposed by extending the velocity field into the boundary as described in [Monaco et al. 2011].

REFERENCES

N. Akinci, J. Cornelis, G. Akinci, and M. Teschner. 2013. Coupling elastic solids with smoothed particle hydrodynamics fluids. *Computer Animation and Virtual Worlds* 24, 3-4 (2013), 195–203.

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

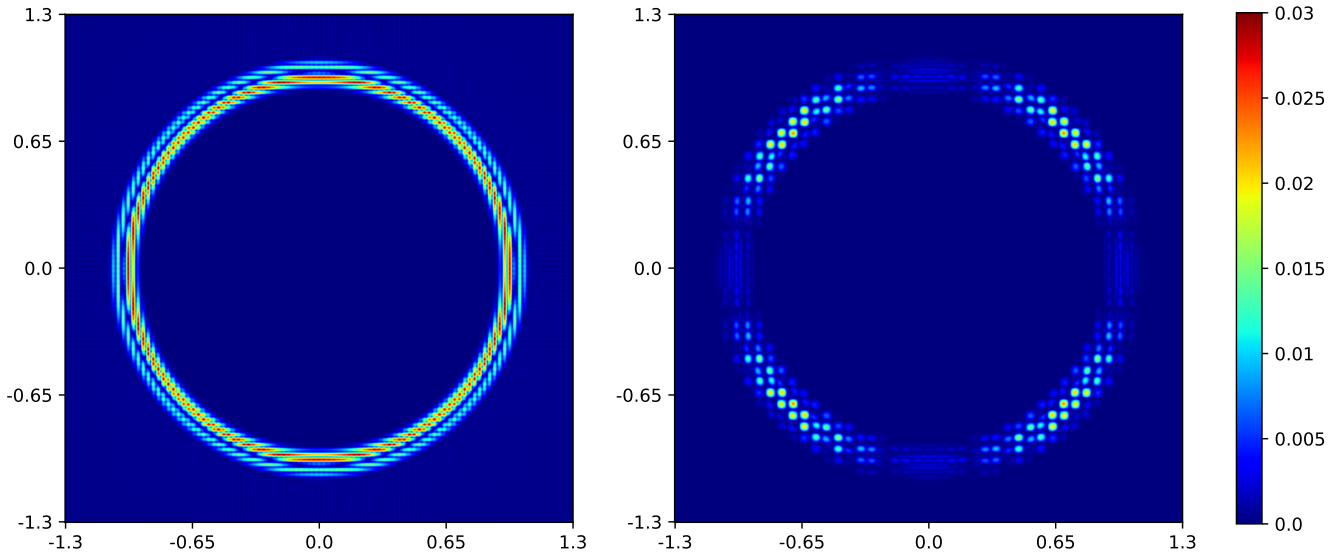


Figure 12: Error plots of piecewise linear (left) and cubic (right) density maps for a two-dimensional circular boundary with radius 1.0, smoothing length $h = 0.1$ and reference density $\rho_0 = 1000$. The colors indicate the error $\frac{|\rho_B - \rho_B^c|}{\rho_0}$.

(2012), 62:1–62:8.

D.N. Arnold and G. Awanou. 2011. The Serendipity Family of Finite Elements. *Foundations of Computational Mathematics* 11, 3 (2011), 337–344.

J.A. Bærentzen and H. Aanæs. 2005. Signed distance computation using the angle weighted pseudonormal. *IEEE Trans. on Visualization and Computer Graphics* 11, 3 (2005), 243–253.

S. Band, C. Gissler, and M. Teschner. 2017. Moving Least Squares Boundaries for SPH Fluids. In *Virtual Reality Interactions and Physical Simulations (VRIPhys).* 1–8.

M. Becker and M. Teschner. 2007. Weakly compressible SPH for free surface flows. In *ACM SIGGRAPH/Eurographics Symp. on Computer Animation.* 1–8.

M. Becker, H. Tessendorf, and M. Teschner. 2009. Direct Forcing for Lagrangian Rigid-Fluid Coupling. *IEEE Trans. on Visualization and Computer Graphics* 15, 3 (2009), 493–503.

J. Bender. 2017. PositionBasedDynamics Library. https://github.com/InteractiveComputerGraphics/PositionBasedDynamics.

J. Bender and D. Koschier. 2017. Divergence-Free SPH for Incompressible and Viscous Fluids. *IEEE Trans. on Visualization and Computer Graphics* 23, 3 (2017).

K. Bodin, C. Lacoursière, and M. Servin. 2012. Constraint fluids. *IEEE Trans. on Visualization and Computer Graphics* 18 (2012), 516–526.

M. Corsini, P. Cignoni, and R. Scopigno. 2012. Efficient and Flexible Sampling with Blue Noise Properties of Triangular Meshes. *IEEE Trans. on Visualization and Computer Graphics* 18, 6 (2012), 914–924.

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

K. Erleben. 2013. Numerical methods for linear complementarity problems in physics-based animation. In *ACM SIGGRAPH Courses.* 1–42.

M. Ferrand, D. R. Laurence, B. D. Rogers, D. Violeau, and C. Kassiotis. 2013. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. *International Journal for Numerical Methods in Fluids* 71, 4 (2013), 446–472.

M. Fujisawa and K. Miura. 2015. An Efficient Boundary Handling with a Modified Density Calculation for SPH. *Computer Graphics Forum* 34, 7 (2015), 155–162.

T. Harada, S. Koshizuka, and Y. Kawaguchi. 2007a. Smoothed particle hydrodynamics in complex shapes. In *Spring Conf. on Computer Graph.* 191–197.

T. Harada, S. Koshizuka, and Y. Kawaguchi. 2007b. Smoothed Particle Hydrodynamics on GPUs. In *Computer Graphics International.* 63–70.

X. He, N. Liu, G. Wang, F. Zhang, S. Li, S. Shao, and H. Wang. 2012. Staggered meshless solid-fluid coupling. *ACM Transactions on Graphics* 31, 6 (2012), 1.

M. Huber, B. Eberhardt, and D. Weiskopf. 2015. Boundary Handling at Cloth-Fluid Contact. *Computer Graphics Forum* 34, 1 (2015), 14–25.

M. Ihmsen, N. Akinci, M. Becker, and M. Teschner. 2011. A Parallel SPH Implementation on Multi-Core CPUs. *Computer Graphics Forum* 30, 1 (2011), 99–112.

M. Ihmsen, N. Akinci, M. Gissler, and M. Teschner. 2010. Boundary handling and adaptive time-stepping for PCISPH. In *Virtual Reality Interactions and Physical Simulations (VRIPhys).* 79–88.

M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, and M. Teschner. 2014a. Implicit incompressible SPH. *IEEE Trans. on Visualization and Computer Graphics* 20, 3 (2014), 426–435.

M. Ihmsen, J. Orthmann, B. Solenthaler, A. Kolb, and M. Teschner. 2014b. SPH Fluids in Computer Graphics. *Eurographics (STAR)* (2014), 21–42.

M.W. Jones, J.A. Bærentzen, and M. Sramek. 2006. 3D distance fields: a survey of techniques and applications. *IEEE Trans. on Visualization and Computer Graphics* 12, 4 (2006), 581–599.

D. Koschier, C. Deul, and J. Bender. 2016. Hierarchical hp-Adaptive Signed Distance Fields. In *ACM SIGGRAPH/Eurographics Symp. on Computer Animation.* 1–10.

S. Kulasegaram, J. Bonet, R. W. Lewis, and M. Profit. 2004. A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications. *Computational Mechanics* 33, 4 (2004), 316–325.

M. Macklin and M. Müller. 2013. Position Based Fluids. *ACM Transactions on Graphics* 32, 4 (2013), 1–5.

A.D. Monaco, S. Manenti, M. Gallati, S. Sibilla, G. Agate, and R. Guandalini. 2011. SPH Modeling of Solid Boundaries Through a Semi-Analytic Approach. *Engineering Applications of Computational Fluid Mechanics* 5, 1 (2011), 1–15.

J.J. Monaghan. 1992. Smoothed Particle Hydrodynamics. *Annual Review of Astronomy and Astrophysics* 30, 1 (1992), 543–574.

J.J. Monaghan. 1994. Simulating Free Surface Flows with SPH. *J. Comput. Phys.* 110, 2 (1994), 399–406.

M. Müller, D. Charypar, and M. Gross. 2003. Particle-Based Fluid Simulation for Interactive Applications. In *ACM SIGGRAPH/Eurographics Symp. on Computer Animation.* 154–159.

M. Müller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross. 2004. Interaction of fluids with deformable solids. *Computer Animation and Virtual Worlds* 15, 34 (2004), 159–171.

R. Piessens and M. Branders. 1974. A Note on the Optimal Addition of Abscissas to Quadrature Formulas of Gauss and Lobatto. *Math. Comp.* 28, 125 (1974), 135–139.

K. Raveendran, C. Wojtan, and G. Turk. 2011. Hybrid smoothed particle hydrodynamics. In *ACM SIGGRAPH/Eurographics Symp. on Computer Animation.* 33–42.

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

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

L. Yang, S. Li, A. Hao, and H. Qin. 2012. Realtime Two-Way Coupling of Meshless Fluids and Nonlinear FEM. *Computer Graphics Forum* 31, 7 (2012), 2037–2046.

O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. 2013. *The Finite Element Method: its Basis and Fundamentals (7th Edition).*