Turbulent Micropolar SPH Fluids with Foam

Jan Bender, Dan Koschier, Tassilo Kugelstadt, and Marcel Weiler

Abstract—In this paper we introduce a novel micropolar material model for the simulation of turbulent inviscid fluids. The governing equations are solved by using the concept of Smoothed Particle Hydrodynamics (SPH). As already investigated in previous works, SPH fluid simulations suffer from numerical diffusion which leads to a lower vorticity, a loss in turbulent details and finally in less realistic results. To solve this problem we propose a micropolar fluid model. The micropolar fluid model is a generalization of the classical Navier-Stokes equations, which are typically used in computer graphics to simulate fluids. In contrast to the classical Navier-Stokes model, micropolar fluids have a microstructure and therefore consider the rotational motion of fluid particles. In addition to the linear velocity field these fluids also have a field of microrotation which represents existing vortices and provides a source for new ones. However, classical micropolar materials are viscous and the translational and the rotational motion are coupled in a dissipative way. Since our goal is to simulate turbulent fluids, we introduce a novel modified micropolar material for inviscid fluids with a non-dissipative coupling. Our model can generate realistic turbulences, is linear and angular momentum conserving, can be easily integrated in existing SPH simulation methods and its computational overhead is negligible. Another important visual feature of turbulent liquids is foam. Therefore, we present a post-processing method which considers microrotation in the foam particle generation. It works completely automatic and requires only one user-defined parameter to control the amount of foam.

Index Terms—Smoothed particle hydrodynamics, micropolar fluids, turbulence, incompressible fluids, foam

1 Introduction

Lagrangian fluid simulation methods based on the Smoothed Particle Hydrodynamics (SPH) approach have been investigated for several years in the field of computer graphics. An important application area is the generation of special effects in movies where a main goal is the realistic simulation of incompressible fluids in turbulent scenarios. In recent years several methods to enforce incompressibility have been introduced. However, the generation of realistic turbulent flows in Lagrangian fluid simulations is still a challenging problem.

In SPH fluid simulations turbulent details quickly get lost due to numerical diffusion [1] or due to a coarse sampling of the velocity field [2], [3] which negatively influences the visual liveliness of the flow. In the field of Eulerian smoke simulations the concept of vorticity confinement was introduced to the graphics community by Fedkiw et al. [4] to counteract numerical dissipation and to amplify existing vorticity. Later, this concept was adapted in SPH fluid simulations to obtain a more detailed motion of bubbles [5] and to counteract numerical damping [6]. However, these SPH variants of vorticity confinement are not momentum conserving and only existing vortices can be amplified. Lagrangian vortex methods, which are based on a vorticity representation of the Navier-Stokes equations, are an alternative approach to simulate turbulent fluids. They maintain a divergence-free velocity field and have no numerical dissipation. However, the handling of non-rigid obstacles and free surface flows is difficult using these methods.

In this paper we introduce an SPH simulation method for incompressible turbulent fluids to solve the above-mentioned problems. Our method is based on a novel micropolar material model for fluids. The theory of micropolar fluids is well-established in the field of mechanical engineering [7], [8]. A micropolar fluid model is a generalization of the classical Navier-Stokes model for fluids. In contrast to the classical fluid model, micropolar fluids have a microstructure, i.e. they consist of rigid, spherical micro-elements. This results in a non-symmetric stress tensor. Therefore, the governing equations of the micropolar fluid model consider the rotational motion of the fluid particles. Or more specifically each particle has an angular velocity which defines a field of microrotation. In this field, vortices can exist independently of the linear velocity field. Moreover, this field provides a source for new vortices [8].

In our work we are interested in the simulation of inviscid turbulent fluids. However, the classical micropolar material is intended for viscous flow and couples the translational and the rotational motion in a dissipative way. Therefore, we introduce a modified micropolar material model which is specially tailored for the simulation of inviscid turbulent fluids. Our micropolar fluid model is linear and angular momentum conserving. Additionally, it is easy to implement, turbulences can be controlled with a single parameter and it can be combined with any existing SPH pressure solver without additional effort. Furthermore, the computational overhead compared to a classical SPH simulation is negligible. Finally, to the best of our knowledge, we

- *J. Bender, D. Koschier, and T. Kugelstadt are with the Computer Animation Group, RWTH Aachen University, Aachen 52062, Germany. E-mail: {bender, koschier, kugelstadt}@cs.rwth-aachen.de.*
- *M. Weiler is with Graduate School CE, TU Darmstadt, Darmstadt 64289, Germany. E-mail: weiler@gsc.tu-darmstadt.de.*

Manuscript received 26 Jan. 2018; revised 24 Apr. 2018; accepted 25 Apr. 2018. Date of publication 1 May 2018; date of current version 1 May 2019. (Corresponding author: Jan Bender.)
Recommended for acceptance by S. Hu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2018.2832080

1077-2626 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: McGill University. Downloaded on March 26,2023 at 13:48:18 UTC from IEEE Xplore. Restrictions apply.

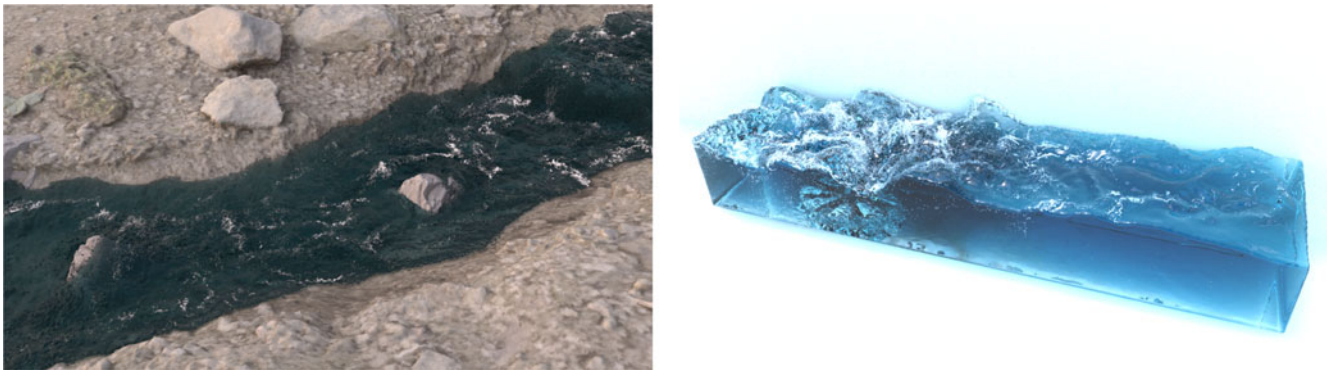

Fig. 1. Our novel micropolar fluid model allows the simulation of complex turbulent fluids with static and dynamic boundaries. Left: A turbulent river with 12 million fluid particles flows through a complex static boundary. Right: One million fluid particles interact with a fast rotating propeller.

are the first to introduce a micropolar fluid model in the field of computer graphics and the first to solve the governing equations for a micropolar model using SPH.

Our method is able to perform realistic simulations of complex turbulent fluid scenarios with several million particles as demonstrated in Fig. 1. Moreover, it provides a more realistic visual liveliness of the flow than classical SPH fluids as we show in different comparisons in Section 7.

To improve the visual quality of our results we add foam particles in a post-processing step which can be seen in Figs. 1, 5, 8 and in the accompanying video. We use an extended version of the algorithm of Ihmsen et al. [9]. Their foam generation method is based on three criteria: trapped air, wave crest and kinetic energy. We add another criterion that considers turbulences based on the microrotations of the fluid. This leads to increased visual details in turbulent regions of the fluid surface. Further, we enhance the method so that all parameters of the source terms are chosen automatically and the user can control the amount of generated foam with a single intuitive parameter.

2 Related Work

In the field of computer graphics, the simulation of turbulent fluids has been a topic of active research for nearly two decades. Moreover, SPH based methods for the simulation of free surface flow of incompressible fluids have gained popularity over the last years. In this section, we will discuss related work in the fields of SPH-based simulation, turbulent fluid simulation and micropolar material models and organized the discussion respectively. For a general survey on SPH based fluid simulations we would like to refer the reader to the state-of-the-art report of Ihmsen et al. [2].

SPH for Incompressible Fluids. In recent years, a lot of attention has been paid to the development of implicit pressure solvers in order to efficiently enforce incompressibility in SPH based fluid simulations. Existing approaches can be mainly categorized into methods that iteratively project particle positions onto an uncompressed state and pressure projection methods that use either the divergence of the velocity field or the compression on position level as source term (cf. [2]). Solenthaler and Pajarola [10] and Macklin and Müller [6] proposed predictive-corrective approaches that iteratively project particle positions onto an uncompressed state. Approaches following the strategy of pressure projections were proposed by Ihmsen et al. [11] using the compression as source term and Premože et al. [12] using the velocity divergence as source term. Recently, an approach enforcing both zero compression and a divergence-free velocity field was proposed by Bender and Koschier [13]. In this work, we adopted their approach due to its computational efficiency and stability properties. However, our micropolar model is independent of the pressure solver and can therefore be easily used in combination with other solvers, e.g., [14].

A known problem of SPH approaches is that they introduce numerical damping which is particularly evident in case of coarse discretizations (cf. [1], [2]). From a physical point of view, this leads to an undesired dissipation where high-frequency features are smoothed out as discussed by Ihmsen et al. [2]. It also reduces vorticity and prevents turbulences from forming (cf. [3]). Existing approaches that recover or amplify turbulent details can be mainly categorized into vorticity confinement, Lagrangian vortex methods and fluid up-sampling which will be discussed in the following.

Vorticity Confinement. Fedkiw et al. [4] introduced vorticity confinement to the computer graphics community to solve the issue of numerical dissipation in Eulerian simulations of smoke. The core idea is to identify existing vortices and to subsequently counteract dissipation of the rotational flow by amplification. Lentine et al. [15] extended this work to achieve energy conservation and momentum conserving vorticity confinement forces. An improved advection scheme for conservation of vortices was proposed by Zhang et al. [16]. They measure the vorticity before advection and conserve it by correcting the error acquired through measuring the violations of the vorticity equation. Although the method cannot be directly classified as vorticity confinement, it has a similar motivation and can moreover be applied in combination with vorticity confinement schemes.

Jang et al. [17] used a multilevel vorticity confinement approach to achieve turbulences in Eulerian water simulations. Vorticity confinement was also used in hybrid simulations, in which the fluids were simulated as SPH particles and where an additional grid was used for vorticity confinement. For instance Zhu et al. [18] used a hybrid method and vorticity confinement to simulate fine scale vorticial details around moving rigid objects in SPH smoke simulations. Hong et al. [5] used vorticity confinement in a simulation of bubbles in water. The water was simulated on a grid and the bubbles as SPH particles. They used vorticity confinement on the particles to achieve a more detailed motion. Also in purely particle based approaches vorticity

Authorized licensed use limited to: McGill University. Downloaded on March 26,2023 at 13:48:18 UTC from IEEE Xplore. Restrictions apply.

confinement was used. Macklin and Müller [6] determined the existing vorticity per particle using SPH and computed forces which accelerate the particles to amplify the rotational motion.

Our experiments show that using vorticity confinement directly on SPH particles is a simple way to counteract numerical dissipation, but it cannot create additional turbulent details such as additional vortices. Moreover, the previously proposed vorticity confinement in SPH is not momentum conserving and it also violates energy conservation when the parameter is chosen too high, so that it adds more energy than numerical dissipation has removed. Unfortunately, the energy and momentum conserving Eulerian approach of Lentine et al. [15] can not directly be applied to SPH simulations. In contrast, our method is able to create additional turbulent detail resulting in a feature-rich flow without the need for data structures like Eulerian grids and is linear and angular momentum conserving.

Lagrangian Vortex Methods. Another approach is to simulate turbulent fluids by using the vorticity representation of the Navier-Stokes equations. This method has several advantages. It allows a compact representation of rotational flow features such as eddies, it automatically guarantees divergence-free velocity fields and it does not suffer from numerical dissipation. The vorticity field can be discretized using particles [19], [20], curves/filaments [21], [22], or surfaces/sheets [23], [24]. Further, Zhang et al. [25] proposed a hybrid vorticity and velocity based advection scheme to restore the missing vorticity in Eulerian solvers.

However, the vortex representation also has several disadvantages. It is difficult to handle boundaries such as non-rigid obstacles and free surface flows. Therefore, a hybrid Eulerian and vorticity method was proposed by Golas et al. [26]. They used Lagrangian vortex elements in the interior of the fluid and boundaries were treated on an Eulerian grid. A further difficulty is that the velocity field has to be recovered by solving Biot-Savart integrals or a vector valued Poisson equation [27].

The vorticity based models are related to our method because of the close relationship between vorticity and microrotations. However, in our approach these quantities are coupled by the (angular) momentum equations, i.e. we do not have to recover the velocity field.

Fluid Up-Sampling Methods. One further approach, which is commonly used in Eulerian smoke simulations, is to use coarse grids for the simulation and to increase the resolution afterwards by using fine scale turbulence models [28], [29]. Similar approaches have been proposed to increase the resolution of Eulerian liquid simulations [30], [31]. Also in the Lagrangian setting methods for increasing the resolution of the velocity field [32], [33], adding high frequency turbulent surface details [34] and enriching fluid surfaces with wave details [35] have been presented.

These methods are complementary to our approach. They address the problem of too coarse samplings by generating additional small scale visual details. In contrast we address the problem of numerical dissipation and our method produces additional turbulent details on larger scales. Since the up-sampling methods act as a post processing step, they should also be applicable in combination with our approach.

Micropolar Fluids. The model of micropolar fluids was introduced by Eringen [36]. Since then it was successfully used in many practical applications. Examples are the flow of colloidal suspensions and polymeric fluids [37], liquid crystals [38], human and animal blood [39] and lubricants [40]. An introduction to the theory of micropolar fluids can be found in the textbooks of Lukaszewicz [7] and Eringen [8].

Our motivation for using the micropolar model to simulate classical fluids like water is, that microrotations are additional sources of vorticity and turbulence as pointed out by Eringen [8]. This additional vorticity results in visual effects that are not captured in standard SPH simulations due to numerical dissipation. A distinctive feature of our micropolar SPH discretization is that each particle carries an angular velocity vector additional to the usual linear velocity vector. A particle based approach for semi-Lagrangian advection in Eulerian discretizations that has some similarities to the micropolar approach was proposed by Jiang et al. [41]. While their approach also captures non-linear motion in the particle representation it is rather targeted towards minimizing numerical dissipation caused by projections between a Eulerian discretization and the particle representation. However, their method can, in the best case, only conserve angular momentum but not introduce new vortices.

Foam Generation. The realism of a simulated flow can be significantly improved by enriching the visual representation with bubbles and foam. In recent years, several approaches to generate foam for fluid simulations were presented.

In the field of grid-based solvers it is common practice to employ auxiliary particles. Motivations to incorporate auxiliary particles are to perform semi-Lagrangian advection in Eulerian simulation methods (see e.g., [41]), to add detail to under-resolved regions (see e.g., [42]) or to model secondary effects such as foam, bubbles or splashes. One of the first approaches in the field computer graphics for the generation of foam for splashing fluids was presented by Takahashi et al. [43]. They employ a grid-based fluid solver with surface tracking and generate auxiliary foam and splash particles when the curvature of the tracked surface exceeds a certain threshold. The principle of auxiliary particles was also adopted by Kim et al. [44] for modeling splashes and droplets that escape the main fluid body. Losasso et al. [45] follow a similar approach. They use an Eulerian, grid-based discretization with semi-Lagrangian advection and track the fluid surface using a particle level-set. When the advected auxiliary particles cross the free surface of the fluid body they are converted to SPH particles to model diffuse materials, i.e. spray, foam, and bubbles. A very similar approach was proposed by Wang et al. [46]. They employ a fully two-way coupled fluid simulation based on the Lattice-Boltzmann-Method and SPH with particle level-set surface tracking and convert the massless level-set particles to SPH particles in order to represent spray, foam, and bubbles depending on their location. Although this strategy is simple and yields good results, it does not generalize well to other discretization approaches due to the reliance on particle level-set surface tracking.

As opposed to generating particles based on solely geometric criteria, a physically motivated approach was proposed by Mihalef et al. [47] for the simulation of small-scale

Authorized licensed use limited to: McGill University. Downloaded on March 26,2023 at 13:48:18 UTC from IEEE Xplore. Restrictions apply.

splashes and bubbles. They seed particles following a criterion based on the Weber number (cf. [48]) which depends on the ratio of the quadratic velocity and surface tension. Particles inside the fluid are then treated as bubbles while particles located outside the fluid are treated as droplets. The Weber number was also used by Bagar et al. [49] in the context of SPH simulations. They do not employ auxiliary particles but instead directly classify SPH particles into fluid and foam. The rendering pipeline is then modified accordingly. However, disadvantages of the approach are that the level of detail is directly dependent on the resolution of the SPH discretization and that no secondary dynamic effects of the spray and foam particles evolve.

A practical method for the generation of realistic foam, spray, and bubbles as post-processing step purely building on particle-based simulations was proposed by Ihmsen et al. [9]. They generate auxiliary particles based on the potential of a fluid particle to mix with air. This potential is composed of several source terms depending on wave crests, trapped air, and kinetic energy which are controlled using several parameters.

In this work, we extend the approach of Ihmsen et al. We introduce an additional potential for vorticity and extend the kinetic energy source term by considering angular velocity. Moreover, we present a heuristic which allows an automatic parameter selection in order to reduce the number of control parameters to a single intuitive parameter.

3 Governing Equations and Micropolar Material Model

The most prominent mathematical model describing the dynamics of Newtonian fluids is the Navier-Stokes model. The model can be derived from the conservation law of linear momentum and by presuming that the mechanical stress is composed of isotropic pressure and a diffusing viscous term. An important assumption of the model is that the infinitesimally small particles which compose a fluid continuum are not subject to rotational motion. This also implies that the law of angular momentum conservation is identically fulfilled if and only if the stress tensor is symmetric.

In this section we introduce the concept of micropolar fluids and present a material model that generalizes the Navier-Stokes equations for the simulation of incompressible inviscid turbulent flow. Following the definition of Lukaszewicz [7], a micropolar fluid follows constitutive laws modeled using a generally non-symmetric stress tensor. Moreover, the definition includes that the fluid consists of rigid, spherical (and therefore rotationally invariant) particles. Based on the non-symmetric stress measures, the micropolar model additionally models rotating motions of the infinitesimal spherical particles using an angular velocity field. Due to the additional rotational degrees of freedom, the generation of vortices is facilitated and a wider range of potential dynamic effects are captured by the model.

Micropolar materials have been investigated for several decades in the field of mechanical engineering. However, the classical micropolar constitutive equations [7], [8] model viscous flow. Moreover, the viscosity parameters are also used to couple the translational and the rotational motion in a dissipative way which is not desired for inviscid or nearly inviscid flow. In the following, we derive a modified micropolar material model that is more suitable for the simulation of incompressible inviscid turbulent fluids.

3.1 Linear and Angular Momentum Conservation and Incompressibility

In the following we derive the equations for linear and angular motion. Analogously to the Navier-Stokes model, a fluid in a three-dimensional domain is described by a density field $\rho : \mathbb{R}^3 \to \mathbb{R}$ and a velocity field $\mathbf{v} : \mathbb{R}^3 \times [0, \infty) \to \mathbb{R}^3$ that maps a point $\mathbf{x}$ at time t to its velocity vector. Moreover, the model is augmented by a microrotation field $\boldsymbol{\omega} : \mathbb{R}^3 \times [0, \infty) \to \mathbb{R}^3$ that maps a point $\mathbf{x}$ at time t to its angular velocity. The laws for conservation of linear and angular momentum are then defined as

$$\rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \mathbf{T} + \mathbf{f} \tag{1}$$

$$\rho \Theta \frac{D\boldsymbol{\omega}}{Dt} = \nabla \cdot \mathbf{C} + \mathbf{T}_\times + \boldsymbol{\tau} \tag{2}$$

with $[\mathbf{T}_\times]_i = \sum_j \sum_k \epsilon_{ijk} T_{jk}$ and where $\mathbf{T}$, $\mathbf{f}$ and ϵ_{ijk} denote the stress tensor, the specific external force and the Levi-Civita symbol, respectively. The model is further augmented by the scalar, isotropic microinertia coefficient Θ, the specific external torque $\boldsymbol{\tau}$, and the couple stress tensor $\mathbf{C}$ that models stresses resulting from angular motion. A physical interpretation for the microinertia coefficient is that each infinitesimal fluid particle has a certain inertial resistance against rotational accelerations. As discussed by Chen et al. [50] the quantity can be measured experimentally using Lagrangian velocities of tracer particles. However, we decided to choose the microinertia coefficient by experimentally exploring the effects for several example values. We finally set $\Theta = 2$ in all of our results based on our experiences. We would further like to mention that Θ is independent of the size of an SPH particle as the parameter occurs in the continuous equation before discretization.

Since we intend to model incompressible materials, the fluid's density ρ must not change over time. Consequently, the incompressibility condition can be directly derived from the continuity equation resulting in

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot \mathbf{v} = 0. \tag{3}$$

This also implies that the velocity field $\mathbf{v}$ must be divergence-free, i.e., $\nabla \cdot \mathbf{v} = 0$.

3.2 Constitutive Model

In order to derive the equations of motion, a suitable constitutive model for inviscid and low-viscous fluids must be found. Moreover, we aim to develop a constitutive model that generalizes the Navier-Stokes equations. We mathematically model the stress tensor $\mathbf{T}$ and the coupling tensor $\mathbf{C}$ as

$$\mathbf{T} = -p\mathbf{I} + \mu \nabla \mathbf{v}^T - \mu_t \nabla \mathbf{v} + (\mu + \mu_t)\boldsymbol{\omega}^\times \tag{4}$$

$$\mathbf{C} = c \nabla \boldsymbol{\omega}^T, \tag{5}$$

where $[\boldsymbol{\omega}^\times]_{jk} = \sum_i \epsilon_{jik} \boldsymbol{\omega}_i$ and p and $\mathbf{I}$ denote pressure and identity matrix, respectively. The pressure term builds the basis of material models for fluids and is responsible for

Authorized licensed use limited to: McGill University. Downloaded on March 26,2023 at 13:48:18 UTC from IEEE Xplore. Restrictions apply.

counteracting compression. In the Navier-Stokes equation, viscosity is modeled using a symmetric second order tensor, i.e., $\mu(\nabla\mathbf{v} + \nabla\mathbf{v}^T)$. In contrast, we additively decouple the tensor representation and provide individual material parameters μ and μ_t. In the following, we will refer to μ and μ_t as dynamic viscosity and transfer coefficient, respectively. In order to ensure consistency with the second law of thermodynamics the material parameters must satisfy $\mu \geq 0$ and $\mu_t \geq -\mu$. Consequently, we can guarantee that the terms will only dissipate but never produce energy. The last term, i.e., $(\mu + \mu_t)\boldsymbol{\omega}^\times$, is responsible for the transfer from rotational into linear motions and vice versa. Furthermore, this term can be physically interpreted as a model for friction between the infinitesimal material particles in the fluid. Finally, we model the coupling tensor as diffusive term solely dependent on the microrotation gradient and the dynamic rotational viscosity parameter c. Note that we choose the coupling tensor to model vorticity diffusion analogous to the linear viscosity term as also proposed by Eringen [8] and Lukaszewicz [7].

3.3 Equations of Motion

As the conservation laws and constitutive equations are now established, we can finally derive the equations of motion that build the basis for the numerical fluid simulation. By plugging the constitutive Equations (4) and (5) into the conservation laws Equations (1) and (2) and by applying the incompressibility condition Equation (3), we arrive at the following representation

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho}\nabla p + \nu\Delta\mathbf{v} + (\nu + \nu_t)\nabla \times \boldsymbol{\omega} + \frac{\mathbf{f}}{\rho} \tag{6}$$

$$\Theta\frac{D\boldsymbol{\omega}}{Dt} = \zeta\Delta\boldsymbol{\omega} + (\nu + \nu_t)(\nabla \times \mathbf{v} - 2\boldsymbol{\omega}) + \frac{\boldsymbol{\tau}}{\rho}. \tag{7}$$

Here, $\nu = \frac{\mu}{\rho}$, $\nu_t = \frac{\mu_t}{\rho}$, and $\zeta = \frac{c}{\rho}$ denote the kinematic viscosity, the kinematic transfer coefficient and the kinematic rotational viscosity, respectively. Our formulation reduces to the classical Navier-Stokes equations when $\nu_t = -\nu$, $\zeta = 0$, and $\boldsymbol{\tau} = 0$ which also proves that our model is a generalization of the Navier-Stokes model. It is also worth noting, that in our model the transfer between rotational and linear motion imposed by ν_t is, in contrast to ν, non-dissipative since the coefficient is not involved in the diffusive terms.

Finally, for inviscid fluids the model can be reduced by choosing $\nu = 0$ and $\zeta = 0$ resulting in

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho}\nabla p + \nu_t\nabla \times \boldsymbol{\omega} + \frac{\mathbf{f}}{\rho} \tag{8}$$

$$\Theta\frac{D\boldsymbol{\omega}}{Dt} = \nu_t(\nabla \times \mathbf{v} - 2\boldsymbol{\omega}) + \frac{\boldsymbol{\tau}}{\rho}. \tag{9}$$

4 Smoothed Particle Hydrodynamics Discretization

In this work we use the Smoothed Particle Hydrodynamics approach to spatially discretize the governing equations. To solve these equations we have to compute the density, the pressure term, the transfer terms and the viscosity terms (only if viscosity is required). Moreover, we have to consider the interaction between fluid and boundary.

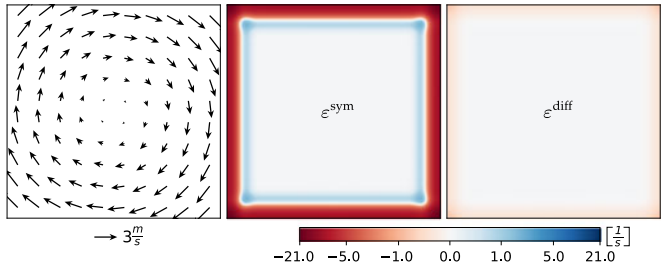

Fig. 2. Comparison of the symmetric and the difference SPH curl formulation. Left: Velocity field of 21×21 particles rotating around the origin. Middle: Error $\varepsilon^{\text{sym}} = (\nabla \times \mathbf{v})^{\text{sym}} - 2\boldsymbol{\omega}$ of the symmetric curl formulation. Right: Error $\varepsilon^{\text{diff}} = (\nabla \times \mathbf{v})^{\text{diff}} - 2\boldsymbol{\omega}$ of the difference curl formulation.

Density. Using SPH the density ρ_i at a position $\mathbf{x}_i$ is determined by

$$\rho_i = \sum_j m_j W_{ij},$$

where m_j denotes the mass of the neighboring particles j and $W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j, h)$ denotes a smoothing kernel with smoothing length h. In our simulations we use the cubic spline kernel [51]. However, also other kernel functions can be used.

Pressure. Since we want to simulate incompressible fluids, we compute the pressure forces using an implicit pressure solver. Our approach can be easily combined with any SPH pressure solver. We successfully tested our micropolar material model in combination with the current pressure solvers PBF [6], IISPH [11] and DFSPH [13]. However, for all our experiments we used DFSPH because of its good performance and its stability.

Transfer Terms. To compute the transfer terms we have to determine the curl of the linear and angular velocity field. In most of the previous works one of the following two SPH formulations of the curl operator is used: the symmetric curl formulation [52]

$$(\nabla \times \mathbf{A})_i^{\text{sym}} = -\rho_i \sum_j m_j \left(\frac{\mathbf{A}_i}{\rho_i^2} + \frac{\mathbf{A}_j}{\rho_j^2}\right) \times \nabla_i W_{ij}$$

or the difference curl formulation [51]

$$(\nabla \times \mathbf{A})_i^{\text{diff}} = \frac{1}{\rho_i}\sum_j m_j(\mathbf{A}_i - \mathbf{A}_j) \times \nabla_i W_{ij}.$$

In our earlier paper [53] we used the first one. However, recently we observed that when using this formulation visual artifacts can occur at the free surface due to particle deficiency. In an experiment we found out that the difference formulation yields better results at the free surface. In this experiment we computed the curl using both formulations for a planar velocity field of 21×21 particles rotating around the origin with an angular velocity of one (see Fig. 2, left). For such a velocity field the curl can be exactly determined as $\nabla \times \mathbf{v} = 2\boldsymbol{\omega}$. Fig. 2 (middle) shows the difference between the symmetric SPH formulation and the exact solution while Fig. 2 (right) shows the error of the difference SPH formulation. The symmetric curl formulations yields a large error of up to -21.2 at the free surface and the resulting angular velocity even points in the opposite direction of the

Authorized licensed use limited to: McGill University. Downloaded on March 26,2023 at 13:48:18 UTC from IEEE Xplore. Restrictions apply.

exact solution. In contrast the difference curl has only a low error of -0.8 at the free surface and points in the correct direction. This means that using the difference curl to obtain the angular velocity from a linear velocity field has a small damping effect at the free surface while the symmetric curl formulation can lead to visual artifacts. Therefore, we use the difference curl formulation in our work.

Viscosity. In this work we are interested in inviscid fluids. However, if viscosity is required, the Laplacian in the viscosity terms in Equations (6) and (7) can be determined using the following SPH approximation [54], [55]:

$$\Delta \mathbf{A}_i = 2(d+2)\sum_j \frac{m_j}{\rho_j}\frac{\mathbf{A}_{ij}\cdot\mathbf{x}_{ij}}{\|\mathbf{x}_{ij}\|^2+0.01h^2}\nabla W_{ij},$$

where $\mathbf{x}_{ij}=\mathbf{x}_i-\mathbf{x}_j$ and d is the number of spatial dimensions. The advantage of this formulation is that it is Galilean invariant, vanishes for rigid body rotation and conserves linear and angular momentum.

Boundary Interaction. We model boundary interactions with rigid bodies by using the method of Akinci et al. [56]. The boundaries are sampled with particles and the mass of each boundary particle is estimated as $\Psi_i=\frac{\rho_0}{\sum_k W_{ik}}$, where k are the indices of neighboring boundary particles. The SPH discretization of the transfer term in Equation (8) at the boundary is

$$\nu_t(\nabla\times\boldsymbol{\omega})_i = \nu_t\sum_k \frac{\Psi_k}{\rho_i}(\boldsymbol{\omega}_i-\boldsymbol{\omega}_k)\times\nabla_i W_{ik},$$

and the discretization of the transfer term in Equation (9) is

$$\nu_t((\nabla\times\mathbf{v})_i-2\boldsymbol{\omega}_i) = \nu_t\left(\sum_k\frac{\Psi_k}{\rho_i}(\mathbf{v}_i-\mathbf{v}_k)\times\nabla_i W_{ik}\right)-2\boldsymbol{\omega}_i.$$

Again the index k denotes a summation over all neighboring boundary particles. The angular velocity $\boldsymbol{\omega}_k$ of a boundary particle k is defined by the angular velocity $\boldsymbol{\omega}^{\mathrm{rb}}$ of the according rigid body. The velocity $\mathbf{v}_k$ is defined by the point velocity which is computed as $\mathbf{v}_k=\mathbf{v}^{\mathrm{rb}}+\boldsymbol{\omega}^{\mathrm{rb}}\times\mathbf{r}_k$, where $\mathbf{v}^{\mathrm{rb}}$ is the linear velocity of the rigid body and $\mathbf{r}_k$ is the vector that points from the rigid body's center of mass to the position of the boundary particle. This boundary handling method leads to turbulent motions when the fluid interacts with the boundary particles of a rigid body.

5 Smoothing the Velocity Field

In our work we aim to simulate turbulent inviscid fluids. Therefore, we set the viscosity parameters ν and ζ to zero and removed the corresponding terms in Equations (8) and (9). But note that our micropolar material model can also be used for viscous fluids. In this case the viscosity terms $\nu\Delta\mathbf{v}$ and $\zeta\Delta\boldsymbol{\omega}$ can be computed using the SPH approximation of the Laplacian (see Section 4).

The simulation of inviscid fluids using SPH has the problem that particles can pass through each other which leads to unphysical results since the velocity field can be locally multivalued [57], [58]. To solve this problem Monaghan [57] proposed to smooth the velocity field so that near particles move with almost identical velocities. This method, also known as XSPH [51], conserves linear and angular momentum. In previous works XSPH is sometimes used to simulate artificial viscosity (e.g., [59]) since it smoothes out turbulent details in the velocity field which lets the fluid look viscous. However, note that XSPH is non-dissipative. In this work we use XSPH due to its property that it transfers the velocity of a particle to its neighbors in absence of viscosity and use a low coefficient since we do not want to loose details.

When smoothing the velocity field using XSPH, the smoothed velocity $\hat{\mathbf{v}}_i$ for a particle is determined by

$$\hat{\mathbf{v}}_i = \mathbf{v}_i+\gamma\sum_j\frac{m_j}{\rho_j}(\mathbf{v}_j-\mathbf{v}_i)W_{ij},$$

where $0\le\gamma<1$ is a tunable parameter. In contrast to classical Navier-Stokes fluids, we have a field of angular velocities in our micropolar fluid model. Since we also want to transfer the angular velocity of a particle to its neighbors, we propose to apply XSPH to the angular velocity field as:

$$\hat{\boldsymbol{\omega}}_i = \boldsymbol{\omega}_i+\eta\sum_j\frac{m_j}{\rho_j}(\boldsymbol{\omega}_j-\boldsymbol{\omega}_i)W_{ij}.$$

Note that while the original XSPH is used to avoid that particles pass through each other, our XSPH variant for angular velocities has a different purpose. It is required to transfer the angular velocity of particles in order to get larger vortices in absence of viscosity.

6 Foam Generation

Turbulent flows typically cause the mixture of water and air which leads to the generation of air bubbles. A realistic animation of turbulent fluids must therefore consider this natural phenomenon. In this section we present a method to generate spray, foam and bubble particles in a post-processing step which improves the visual realism of our results significantly. This method is an extension of the approach of Ihmsen et al. [9]. In the following we will briefly introduce the approach of Ihmsen et al. and highlight our extensions.

Ihmsen et al. use different potentials to estimate the probability of fluid-air mixture. More specifically, they combine the potential of a fluid particle to trap air, to be at the wave crest and its kinetic energy. In our work we introduce an additional potential for vorticity and consider the angular velocity in the kinetic energy computation. Moreover, we present a heuristic which allows an automatic parameter selection. This reduces the number of values that must be defined by the user to a single intuitive parameter which controls the amount of foam particles.

Each potential I is clamped by a minimum value $\tau^{\min}$ and a maximum value $\tau^{\max}$ and mapped to the range from zero to one by the function

$$\Phi(I,\tau^{\min},\tau^{\max}) = \frac{\min(I,\tau^{\max})-\min(I,\tau^{\min})}{\tau^{\max}-\tau^{\min}}.$$

Therefore, the user has to provide the values $\tau^{\min}$ and $\tau^{\max}$ for each potential.

Trapped Air. Since air is often trapped by impacts, Ihmsen et al. propose to define the trapped air potential by the velocity difference

$$v_i^{\mathrm{diff}} = \sum_j\|\mathbf{v}_{ij}\|\left(1-\frac{\mathbf{v}_{ij}}{\|\mathbf{v}_{ij}\|}\cdot\frac{\mathbf{x}_{ij}}{\|\mathbf{x}_{ij}\|}\right)W'_{ij},$$

Authorized licensed use limited to: McGill University. Downloaded on March 26,2023 at 13:48:18 UTC from IEEE Xplore. Restrictions apply.

where $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$, and $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$. Here a special kernel function is used to get a better approximation at the free surface:

$$W'_{ij} = \frac{3}{\pi h^3} \begin{cases} 1 - \|\mathbf{x}_{ij}\|/h & \text{if} \|\mathbf{x}_{ij}\| \leq h \\ 0 & \text{otherwise.} \end{cases}$$

Note that we modified the kernel function of Ihmsen et al. to get a normalized kernel. The final potential is defined by $I_{\text{ta}} = \Phi(v_i^{\text{diff}}, \tau_{\text{ta}}^{\min}, \tau_{\text{ta}}^{\max})$.

Wave Crest. To define the wave crest potential Ihmsen et al. propose to use the surface curvature. However, only regions are of interest where the surface is convex. Therefore, the curvature is approximated as $\kappa_i = \sum_j \kappa_{ij}$ with

$$\kappa_{ij} = \begin{cases} \left(1 - \frac{\mathbf{n}_i}{\|\mathbf{n}_i\|} \cdot \frac{\mathbf{n}_j}{\|\mathbf{n}_j\|}\right) W'_{ij} & \text{if } \mathbf{x}_{ji} \cdot \mathbf{n}_i < 0 \\ 0 & \text{otherwise.} \end{cases}$$

The second criterion to identify a particle at the wave crest is to check if it moves in normal direction using

$$\delta_i = \begin{cases} 1 & \text{if} \left(\frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \cdot \frac{\mathbf{n}_i}{\|\mathbf{n}_i\|}\right) \geq 0.6 \\ 0 & \text{otherwise.} \end{cases}$$

Finally, the wave crest potential is defined by $I_{\text{wc}} = \Phi(\delta_i \kappa_i, \tau_{\text{wc}}^{\min}, \tau_{\text{wc}}^{\max})$.

Vorticity. Since our goal is to simulate turbulent flow with vortices, where air is potentially trapped, we extend the approach of Ihmsen et al. by a vorticity potential. This potential considers the differences in the angular velocity field of our micropolar material model:

$$\omega_i^{\text{diff}} = \sum_j \|\boldsymbol{\omega}_{ij}\| W'_{ij},$$

where $\boldsymbol{\omega}_{ij} = \boldsymbol{\omega}_i - \boldsymbol{\omega}_j$. The vorticity potential is defined by $I_{\text{v}} = \Phi(\omega_i^{\text{diff}}, \tau_{\text{v}}^{\min}, \tau_{\text{v}}^{\max})$.

Kinetic Energy. The last potential is based on the kinetic energy

$$E_i = \frac{1}{2} m_i \mathbf{v}_i^2 + \frac{1}{2} \Theta_i \boldsymbol{\omega}_i^2.$$

While Ihmsen et al. only consider the linear velocity field, we also take the angular velocity field of our micropolar model into account. The potential is then defined by $I_{\text{k}} = \Phi(E_i, \tau_{\text{k}}^{\min}, \tau_{\text{k}}^{\max})$.

Sampling. Finally, we combine all potentials to determine the number of foam particles that are generated in a simulation step for a fluid particle:

$$n = I_{\text{k}}(k_{\text{ta}} I_{\text{ta}} + k_{\text{wc}} I_{\text{wc}} + k_{\text{v}} I_{\text{v}}) \Delta t, \tag{10}$$

where Δt is the time step size. The parameters k_{ta}, k_{wc} and k_{v} define the maximum number of particles that should be generated by the corresponding criterion per second. The new particles are generated and randomly distributed in a cylinder which is defined by the radius of a fluid particle i and its velocity $\mathbf{v}_i$. More details can be found in [9]. Each particle is generated with a certain lifetime and it is deleted when its lifetime is over. In our work we set the lifetime in relation to the kinetic energy potential and add a random value to get more natural results.

Advection. The advection of the generated particles is performed according to [9]. Particles with less than six fluid neighbors are classified as spray particles. Bubble particles have more than 20 neighbors. Otherwise the particles are classified as foam. The velocity change of these particles is determined as

$$\Delta \mathbf{v}_i = \begin{cases} \Delta t \frac{\mathbf{f}_{\text{ext}} + \mathbf{g}}{m} & \text{spray particle} \\ \tilde{\mathbf{v}}_i & \text{foam particle} \\ \Delta t \left(-k_b \mathbf{g} + \frac{k_d}{\Delta t} (\tilde{\mathbf{v}}_i - \mathbf{v}_i)\right) & \text{bubble particle,} \end{cases}$$

where k_b and k_d are user-defined parameters to control buoyancy and drag effects, $\mathbf{g}$ is gravity, $\mathbf{f}_{\text{ext}}$ is the sum of all external forces and $\tilde{\mathbf{v}}_i = \frac{\sum_j \mathbf{v}_j W_{ij}}{\sum_j W_{ij}}$ is the averaged velocity of the neighboring fluid particles. Finally, the positions and velocities of all particles are updated as $\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \Delta t(\mathbf{v}_i(t) + \Delta \mathbf{v}_i)$ and $\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \Delta \mathbf{v}_i$ except for foam particles, where only the positions are updated to keep the particles on the fluid surface.

Automatic Parameter Selection. The generation of spray, foam and bubble particles improves the visual realism significantly. However, the approach of Ihmsen et al., which we extended in our work, has also a drawback: The user has to define many parameters. In our case 11 parameters are required to determine the number of generated foam particles: k_{ta}, k_{wc}, k_{v} and a minimum and maximum value for each potential. Finding good values for these parameters can be a tedious task for the user. Therefore, we developed a simple method to determine these parameters automatically. Since the foam is generated in a post-processing step after the simulation, we can analyze the complete simulation data before the generation starts. During this analysis we determine the maximum values per frame for the potentials I_{ta}, I_{wc}, I_{v} and I_{k}. Then we compute the average values over all frames. This gives us a suitable approximation for the values $\tau^{\max}$. We choose a lower limit of $\tau^{\min} = 0.1\tau^{\max}$ which worked well in our experiments. Since now we have suitable minimum and maximum values, all potentials range between zero and one. Therefore, we set the values k_{ta}, k_{wc} and k_{v} all to one to get well-balanced potentials. Finally, we multiply the value n in Equation (10) with a new parameter k. This value lets the user intuitively control the amount of generated foam. The advantage of our proposed automatic parameter selection is that the user has to set only one intuitive parameter instead of 11 parameters. However, an additional analysis of the simulation data is required which can take several minutes.

7 Results and Discussion

In the following we show results for our novel micropolar fluid model and comparisons with the classical Navier-Stokes model. To perform the simulations, we integrated our micropolar model in the open-source fluid library SPlisHSPlasH [60]. In all simulations we used the parallel neighborhood search algorithm introduced by Ihmsen et al. [61], the boundary handling proposed by Akinci et al. [56] and an adaptive time-stepping based on the CFL condition. Moreover, we employed the implicit pressure

Authorized licensed use limited to: McGill University. Downloaded on March 26,2023 at 13:48:18 UTC from IEEE Xplore. Restrictions apply.

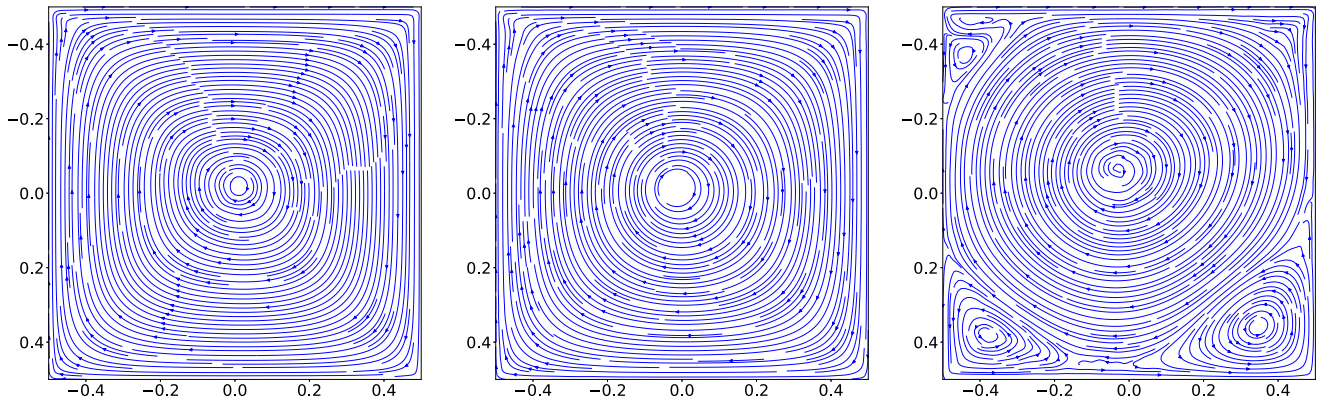

Fig. 3. Velocity fields of the lid-driven cavity benchmark. CSPH (left) and VCSPH (middle) produce only one large central vortex. In contrast MPSPH (right) shows the expected results. It produces the central vortex and three smaller vortices in the corners which are rotating in the opposite direction.

solver DFSPH [13] and enforced an average volume deviation of less than 0.01% and a divergence-error of less than 0.1 percent. In our experiments we simulate inviscid fluids ($\nu = 0$, $\zeta = 0$) and smooth the linear and angular velocity fields using XSPH as discussed in Section 5. We used a relatively small value of $\gamma = 0.002$ since our goal is to simulate almost inviscid turbulent fluids and with large values the fluid tends to look viscous (cf. Section 5). To get larger vortices in our results we set η to 0.125. As already pointed out in Section 3.1 we set $\Theta = 2$ in all of our simulations.

In this section for better readability we use the abbreviation CSPH for the classical SPH method where we solve the classical Navier-Stokes equations using DFSPH in combination with XSPH. If we compute additional vorticity confinement forces according to [6], we abbreviate the method by VCSPH. The simulation with our micropolar fluid model we denote by MPSPH.

Lid-Driven Cavity. To evaluate our method, we chose the lid-driven cavity scenario, which is a standard benchmark in computational fluid dynamics [62]. It is a 2D problem consisting of a square cavity with edge lengths $L = 1$ m, that is completely filled with fluid. The fluid is accelerated by a horizontally moving lid, such that the fluid flows with a velocity of $|\mathbf{v}| = 1$ m/s in horizontal direction at the upper boundary. At the remaining boundaries, no-slip conditions are applied. In our case these conditions are enforced by the method described in [63]. For each fluid particle i that is closer than the support radius to the boundary, a ghost particle is created by reflecting the particle at the boundary. Its velocity $\mathbf{v}_i^g$ is set to $\mathbf{v}_i^g = 2\mathbf{v}^b - \mathbf{v}_i$, such that the interpolated velocity at the boundary becomes $\mathbf{v}^b$. Further there is no gravity. The simulation was initialized with 12k fluid particles such that the cavity was completely filled. For the 2D simulations the 2D cubic spline kernel [51] was used. The kinematic viscosity ν is set, such that we get a Reynolds number $Re = \frac{|\mathbf{v}|L}{\nu}$ of 10000, which was also used in [62].

According to the results presented in [62], which were achieved with a multi-grid finite-difference solver, we expect that this system develops a steady state flow. There should be one large vortex in the center and three smaller vortices in the corners of the cavity which rotate in the opposite direction as the central vortex. Fig. 3 shows the resulting velocity fields of CSPH (left), VCSPH (middle) and our micropolar model (right). To obtain these velocity fields we simulated the scenario for a longer time period until we reached a steady flow. The classical SPH simulation shows the large central vortex, but does not produce the smaller vortices in the corners. We think the reason is the numerical diffusion of SPH which was already discussed in previous works [1], [2], [3]. With additional vorticity confinement the picture looks similar since this method only amplifies the existing central vortex, but does not create the expected additional vortices in the corners. For this simulation we used the vorticity confinement approach of Macklin and Müller [6]. We tested various values of the vorticity confinement parameter ε in their model, but only the rotation speed of the central vortex changed and no new vortices were formed. In Fig. 3 the value was set to $\varepsilon = 0.1$.

The micropolar SPH simulation shows the expected velocity profile with one additional vortex in the top left and two vortices in the bottom left and bottom right corners. In MPSPH the vorticity of the velocity field induces microrotations. The microrotations are advected along with the particle motion and are not much affected by numerical dissipation and the projection onto a divergence-free velocity field. These rotations are a source for new vortices. Further, as microrotations are transferred to nearby particles by XSPH, the formation of larger vortices is supported. An advantage of the micropolar model is that vorticity can be represented even in a single particle. VCSPH can represent vorticity only in particle neighborhoods. Moreover, in contrast to VCSPH micropolar fluids are linear and angular momentum conserving by construction. In summary it can be stated that the micropolar model is better suited for simulations of turbulent fluids than VCSPH because microrotations allow more complex interactions of particles than just accelerating existing vortices.

Comparisons. To demonstrate the effect of the transfer coefficient ν_t in our micropolar model, we first simulated a dam break scenario with 1 M fluid particles and a half sphere as static obstacle using increasing values of ν_t (see Fig. 4). The simulation was initialized with a block of water on the left, which collapsed under gravity and flowed around the sphere. In the first simulation (top) we set $\nu_t = 0$, which results in a classical SPH simulation without microrotations, and then increased ν_t from top to bottom. The comparison shows that turbulent details quickly get lost in the linear velocity field when using classical SPH. As already discussed for the lid-driven cavity example this happens due to numerical diffusion. With increasing the transfer coefficient ν_t, we observe an increasing amount of

Authorized licensed use limited to: McGill University. Downloaded on March 26,2023 at 13:48:18 UTC from IEEE Xplore. Restrictions apply.

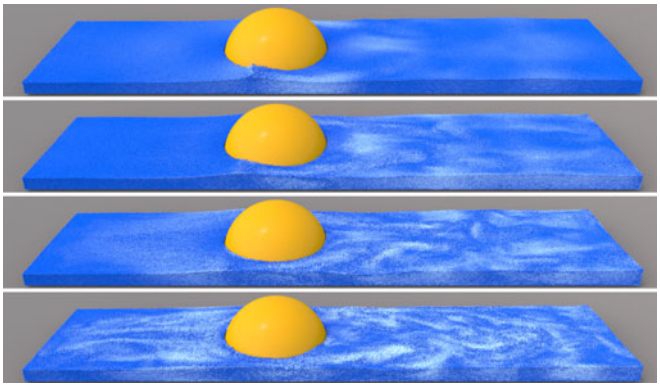

Fig. 4. Breaking dam with 1M fluid particles and a static spherical obstacle simulated with four different transfer coefficients (from top to bottom): $\nu_t = 0.0$, $\nu_t = 0.05$, $\nu_t = 0.075$ and $\nu_t = 0.1$. The linear particle velocity is color coded, white refers to high and blue to low velocity.

turbulent details in the right part of the scene after the fluid flowed around the sphere. This demonstrates that our method allows for intuitive control of additional turbulence with a single parameter.

We also compared our approach with vorticity confinement using the same scenario. In Fig. 5a comparison of CSPH (top), VCSPH (middle) and MPSPH with $\nu_t = 0.05$ (bottom) is depicted. Here we observe that artificially increasing the vorticity using the vorticity confinement method of Macklin and Müller adds energy to the system. This results in a faster fluid flow which can be seen in the middle image. In contrast to VCSPH our method treats energy in a physically consistent way and no additional energy is created. Moreover, in the end of the simulation in the accompanying video we can again observe that vorticity confinement only amplifies existing vortices while our micropolar model also supports the generations of new vortices.

Stability. Further, we simulated a more complex scene with dynamic boundary conditions and high particle velocities to show the stability and effect of our method (see Fig. 1, right). On the left the fluid gets accelerated by a rotating propeller on the ground of the basin. The propeller rotates at 150 revolutions per minute. This results in high velocities of the fluid particles and a highly turbulent motion which is realistically handled by our approach. The motion of the propeller was simulated using the implementation of

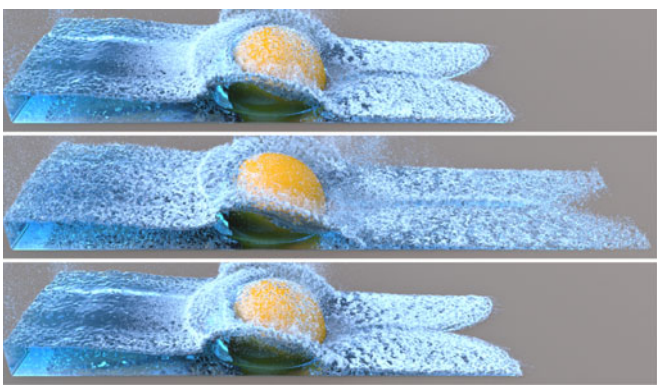

Fig. 5. Comparison of CSPH (top), VCSPH (middle) and MPSPH with $\nu_t = 0.05$ (bottom) in the dam break scenario with static spherical obstacle. Vorticity confinement adds energy to the fluid, which results in a much faster flow. In contrast MPSPH treats energy in a physically consistent way.

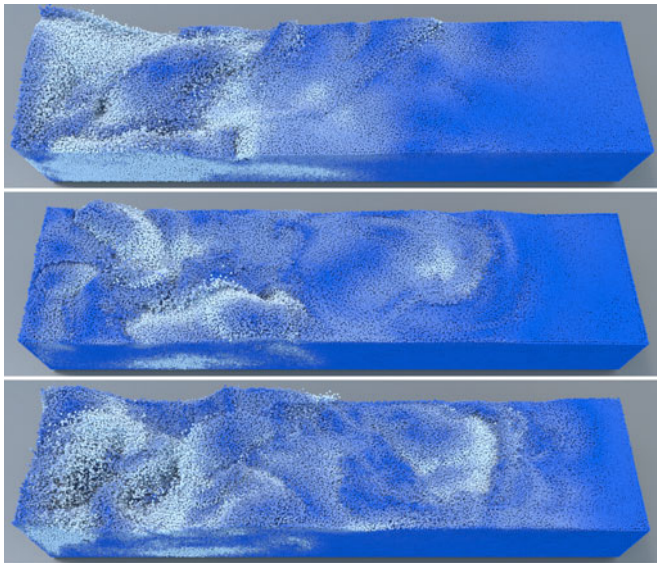

Fig. 6. Propeller scene from Fig. 1 rendered as particles with color coded linear velocities. The propeller in the left of the scene generates high fluid velocities and highly turbulent motion. Using CSPH (top) only a few turbulences reach the right part of the scene due to numerical diffusion. MPSPH with $\nu_t = 0.2$ (middle) and with $\nu_t = 0.3$ (bottom) results in much more turbulent motion and turbulences are transported farther to the right.

position-based rigid body dynamics [64] in the open-source library PositionBasedDynamics [65].

Vorticity. In Fig. 6 the same scene is shown as particle view for CSPH and MPSPH with different values of the transfer coefficient ν_t. The top image shows CSPH, where the turbulent details get lost due to numerical diffusion such that only few turbulences reach the right end of the scene. In the middle we see our MPSPH method with $\nu_t = 0.2$ and on the bottom with $\nu_t = 0.3$. The results show that increasing the transfer coefficient leads to more turbulent motion and the turbulences are transported farther to the right of the scene.

In another experiment we simulated a fast flowing fluid while increasing the transfer coefficient ν_t continuously. Moreover, we added three obstacles to the scenario to generate turbulences. The results are shown in Fig. 7 and in the accompanying video. In the top image we can see that only a few turbulences occur for a transfer coefficient of $\nu_t = 0.2$. For larger values the vorticity significantly increases (middle) and even tends to get unrealistic for values greater than 0.4 (bottom).

Performance. In our last experiment we compared the computational effort of our method and CSPH in a breaking dam scenario with 1M fluid particles and three static Stanford dragons (see Fig. 8). The times were measured on an Intel Xeon E5-2683 processor with 2.1 GHz and 16 cores. The results show that most of the computation time in a simulation step is required for the neighborhood search and the implicit pressure solver. The additional computations for the microrotations in our micropolar fluid model have only a linear time complexity. In the dragon scenario this results in a negligible computational overhead of only 5% compared to the classical SPH approach. In larger scenarios as the river scene (see Fig. 1, left) the computational overhead is even lower since the neighborhood search and the

Authorized licensed use limited to: McGill University. Downloaded on March 26,2023 at 13:48:18 UTC from IEEE Xplore. Restrictions apply.

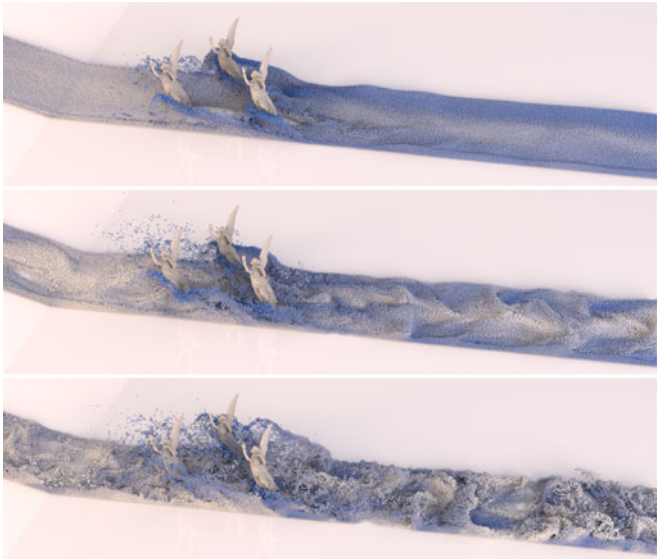

Fig. 7. Simulation of 4.7M turbulent fluid particles with three obstacles and increasing transfer coefficient ν_t. Top-down: $\nu_t = 0.2$, $\nu_t = 0.3$, $\nu_t = 0.4$.

pressure solver require proportionally more time in more complex scenarios.

8 Conclusion

We presented an SPH based approach for simulating turbulent flows of incompressible inviscid fluids. Following a micropolar model the field variables are augmented by a microrotation vector field that represents the angular velocity of infinitesimal particles in the fluid. We proposed a novel material model that generalizes the Navier-Stokes equations and facilitates the generation of vortices which results in highly turbulent flows with feature-rich surface details. The turbulence of the model can be controlled using a single parameter in the material model, i.e., the kinematic transfer coefficient ν_t. In our results we demonstrated that the model produces physically meaningful results, generates interesting turbulent flows compared to the standard approach and robustly handles highly complex large-scale scenarios while the computational overhead is negligible.

Due to the fact that our method is an SPH discretization of a generalization of the Navier-Stokes equations we could not make out any limitations beyond the already evident limitations of existing SPH based incompressible fluid solvers (see e.g., [11], [13]).

In future work we would like to further investigate microstructural models for complex materials. Besides the micropolar approach that describes a material continuum with infinitesimal spherical particles subject to rotation, microstructural models also cover particles that are arbitrarily shaped or deforming. In this regard, we would also like to investigate microstructural formulations for the simulation of elastic solids.

Improving boundary handling is another potential direction for future work. Particle sampled boundaries usually cause an undesired numerical dissipation of the kinetic energy in regions where the fluid is in contact with the boundary. Therefore, we would like to extend the approach for implicit boundary handling proposed by Koschier and Bender [66] in order to incorporate it into our solver for micropolar fluids.

Fig. 8. Breaking dam scenario with 1M fluid particles and three static obstacles.

Acknowledgments

We would like to thank Manuel Scholz for his kind support in providing the river model.
This work is supported by the *German Research Foundation (DFG)* under contract numbers BE 5132/3-1 and by the 'Excellence Initiative' of the German Federal and State Governments and the Graduate School of Computational Engineering at TU Darmstadt.

References

[1] F. de Goes, C. Wallez, J. Huang, D. Pavlov, and M. Desbrun, "Power particles: An incompressible fluid solver based on power diagrams," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 50:1–50:11, 2015.
[2] M. Ihmsen, J. Orthmann, B. Solenthaler, A. Kolb, and M. Teschner, "SPH Fluids in Computer Graphics," *Eurograph. 2014—State of the Art Reports*, S. Lefebvre and M. Spagnuolo, Eds., The Eurographics Association, 2014, doi: 10.2312/egst.20141034.
[3] J. Cornelis, M. Ihmsen, A. Peer, and M. Teschner, "Iisph-flip for incompressible fluids," *Comput. Graph. Forum*, vol. 33, no. 2, pp. 255–262, 2014.
[4] R. Fedkiw, J. Stam, and H. W. Jensen, "Visual simulation of smoke," in *Proc. 28th Annu. Conf. Comput. Graph. Interactive Tech.*, 2001, pp. 15–22.
[5] J.-M. Hong, H.-Y. Lee, J.-C. Yoon, and C.-H. Kim, "Bubbles alive," *ACM Trans. Graph.*, vol. 27, no. 3, 2008, Art. no. 48.
[6] M. Macklin and M. Müller, "Position based fluids," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 104.
[7] G. Lukaszewicz, *Micropolar Fluids: Theory and Applications*. Cambridge, MA, USA: Birkhäuser, 1999.
[8] A. C. Eringen, *Microcontinuum Field Theories: II. Fluent Media*. New York, NY, USA: Springer, 2001.
[9] M. Ihmsen, N. Akinci, G. Akinci, and M. Teschner, "Unified spray, foam and air bubbles for particle-based fluids," *Visual Comput.*, vol. 28, no. 6–8, pp. 669–677, 2012.
[10] B. Solenthaler and R. Pajarola, "Predictive-corrective incompressible SPH," *ACM Trans. Graph.*, vol. 28, no. 3, 2009, Art. no. 40.
[11] M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, and M. Teschner, "Implicit incompressible SPH," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, pp. 426–435, Mar. 2014.
[12] S. Premože, T. Tasdizen, J. Bigler, A. Lefohn, and R. T. Whitaker, "Particle-Based Simulation of Fluids," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 401–410, 2003.
[13] J. Bender and D. Koschier, "Divergence-free sph for incompressible and viscous fluids," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 3, pp. 1193–1206, Mar. 2017.
[14] M. Weiler, D. Koschier, and J. Bender, "Projective fluids," in *Proc. ACM SIGGRAPH Conf. Motion Games*, 2016, pp. 79–84.
[15] M. Lentine, M. Aanjaneya, and R. Fedkiw, "Mass and momentum conservation for fluid simulation," in *Proc. ACM SIGGRAPH / Eurographics Symp. Comput. Animation*, 2011, pp. 91–100.
[16] X. Zhang, R. Bridson, and C. Greif, "Restoring the missing vorticity in advection-projection fluid solvers," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 52:1–52:8, Jul. 2015. [Online]. Available: http://doi.acm.org/10.1145/2766982

Authorized licensed use limited to: McGill University. Downloaded on March 26,2023 at 13:48:18 UTC from IEEE Xplore. Restrictions apply.

[17] T. Jang, H. Kim, J. Bae, J. Seo, and J. Noh, "Multilevel vorticity confinement for water turbulence simulation," *Visual Comput.*, vol. 26, no. 6–8, pp. 873–881, 2010.

[18] B. Zhu, X. Yang, and Y. Fan, "Creating and preserving vortical details in SPH fluid," *Comput. Graph. Forum*, vol. 29, no. 7, pp. 2207–2214, 2010.

[19] S. I. Park and M. J. Kim, "Vortex fluid for gaseous phenomena," in *Proc. ACM SIGGRAPH / Eurographics Symp. Comput. Animation*, 2005, pp. 261–270.

[20] A. Selle, N. Rasmussen, and R. Fedkiw, "A vortex particle method for smoke, water and explosions," *ACM Trans. Graph.*, vol. 24, no. 3, 2005, pp. 910–914.

[21] A. Angelidis and F. Neyret, "Simulation of smoke based on vortex filament primitives," in *Proc. ACM SIGGRAPH / Eurographics Symp. Comput. Animation*, 2005, pp. 87–96.

[22] S. Weißmann and U. Pinkall, "Real-time Interactive Simulation of Smoke Using Discrete Integrable Vortex Filaments," in *Proc. Workshop Virtual Reality Interactions Phys. Simulation*, 2009, pp. 1–10.

[23] S. Weißmann and U. Pinkall, "Filament-based smoke with vortex shedding and variational reconnection," *ACM Trans. Graph.*, vol. 29, no. 4, 2010, Art. no. 115.

[24] T. Pfaff, N. Thürey, and M. Gross, "Lagrangian vortex sheets for animating fluids," *ACM Trans. Graph.*, vol. 31, no. 4, 2012, Art. no. 112.

[25] X. Zhang, R. Bridson, and C. Greif, "Restoring the missing vorticity in advection-projection fluid solvers," *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 52.

[26] A. Golas, R. Narain, J. Sewall, P. Krajcevski, P. Dubey, and M. Lin, "Large-scale fluid simulation using velocity-vorticity domain decomposition," *ACM Trans. Graph.*, vol. 31, no. 6, 2012, Art. no. 148.

[27] X. Zhang and R. Bridson, "A PPPM fast summation method for fluids and beyond," *ACM Trans. Graph.*, vol. 33, no. 6, 2014, Art. no. 206.

[28] T. Kim, N. Thürey, D. James, and M. Gross, "Wavelet turbulence for fluid simulation," *ACM Trans. Graph.*, vol. 27, no. 3, 2008, Art. no. 50.

[29] H. Schechter and R. Bridson, "Evolving sub-grid turbulence for smoke animation," in *Proc. ACM SIGGRAPH / Eurographics Symp. Comput. Animation*, 2008, pp. 1–7.

[30] R. Narain, J. Sewall, M. Carlson, and M. Lin, "Fast animation of turbulence using energy transport and procedural synthesis," *ACM Trans. Graph.*, vol. 27, no. 5, 2008, Art. no. 166.

[31] T. Kim, J. Tessendorf, and N. Thürey, "Closest point turbulence for liquid surfaces," *ACM Trans. Graph.*, vol. 32, no. 2, 2013, Art. no. 15.

[32] Z. Yuan, Y. Zhao, and F. Chen, "Incorporating stochastic turbulence in particle-based fluid simulation," *Vis. Comput.*, vol. 28, no. 5, pp. 435–444, 2012.

[33] X. Shao, Z. Zhou, J. Zhang, and W. Wu, "Realistic and stable simulation of turbulent details behind objects in smoothed-particle hydrodynamics fluids," *Comput. Animation Virtual Worlds*, vol. 26, no. 1, pp. 79–94, 2015.

[34] O. Mercier, C. Beauchemin, N. Thürey, T. Kim, and D. Nowrouzezahrai, "Surface turbulence for particle-based liquid simulations," *ACM Trans. Graph.*, vol. 34, no. 6, 2015, Art. no. 202.

[35] S. Yang, X. He, H. Wang, S. Li, G. Wang, E. Wu, and K. Zhou, "Enriching SPH simulation by approximate capillary waves," in *Proc. ACM SIGGRAPH / Eurographics Symp. Comput. Animation*, 2016, pp. 29–36.

[36] A. C. Eringen, "Theory of micropolar fluids," *J. Math. Mech.*, vol. 16, no. 1, pp. 1–16, 1966.

[37] H. Busuke and T. Tatsuo, "Two-dimensional shear flows of linear micropolar fluids," *Int. J. Eng. Sci.*, vol. 7, no. 5, pp. 515–522, 1969.

[38] J. D. Lee and A. C. Eringen, "Boundary effects of orientation of nematic liquid crystals," *J. Chemical Phys.*, vol. 55, no. 9, pp. 4509–4512, 1971.

[39] T. Ariman, M. A. Turk, and N. D. Sylvester, "Applications of microcontinuum fluid mechanics," *Int. J. Eng. Sci.*, vol. 12, no. 4, pp. 273–293, 1974.

[40] M. M. Khonsari and D. E. Brewe, "On the performance of finite journal bearings lubricated with micropolar fluids," *Tribology Trans.*, vol. 32, no. 2, pp. 155–160, 1989.

[41] C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin, "The Affine Particle-In-Cell Method," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 51:1–51:10, 2015.

[42] N. Foster and R. Fedkiw, "Practical Animation of Liquids," *ACM Trans. Graph.*, vol. 28, pp. 12–17, 2001. [Online]. Available: http://dl.acm.org/citation.cfm?id=383261

[43] T. Takahashi, H. Fujii, A. Kunimatsu, K. Hiwada, T. Saito, K. Tanaka, and H. Ueki, "Realistic animation of fluid with splash and foam," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 391–400, 2003.

[44] J. Kim, D. Cha, B. Chang, B. Koo, and I. Ihm, "Practical animation of turbulent splashing water," in *Proc. ACM SIGGRAPH / Eurographics Symp. Comput. Animation*, 2006, pp. 335–344. [Online]. Available: http://dl.acm.org/citation.cfm?id=1218064.1218110

[45] F. Losasso, J. O. Talton, N. Kwatra, and R. Fedkiw, "Two-way coupled SPH and particle level set fluid simulation," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, pp. 797–804, Jul. 2008.

[46] C. Wang, Q. Zhang, F. Kong, and H. Qin, "Hybrid particle–grid fluid animation with enhanced details," *Vis. Comput.*, vol. 29, no. 9, pp. 937–947, 2013.

[47] V. Mihalef, D. Metaxas, and M. Sussman, "Simulation of two-phase flow with sub-scale droplet and bubble effects," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 229–238, 2009.

[48] W. A. Sirignano and C. F. Edwards, "Fluid dynamics and transport of droplets and sprays," *J. Fluids Eng.*, vol. 122, no. 1, 2000, Art. no. 189.

[49] F. Bagar, D. Scherzer, and M. Wimmer, "A layered particle-based fluid model for real-time rendering of water," *Comput. Graph. Forum*, vol. 29, no. 4, pp. 1383–1389, 2010.

[50] J. Chen, C. Liang, and J. D. Lee, "Theory and simulation of micropolar fluid dynamics," *Proc. Institution Mech. Engin. Part N: J. Nanoengineering Nanosystems*, vol. 224, no. 1/2, pp. 31–39, 2010.

[51] J. J. Monaghan, "Smoothed particle hydrodynamics," *Annu. Review Astronomy Astrophysics*, vol. 30, pp. 543–574, 1992.

[52] D. J. Price, "Smoothed particle magnetohydrodynamics - iv. using the vector potential," *Monthly Notices Roy. Astron. Soc.*, vol. 401, no. 3, 2010, Art. no. 1475.

[53] J. Bender, D. Koschier, T. Kugelstadt, and M. Weiler, "A micropolar material model for turbulent SPH fluids," in *Proc. ACM SIGGRAPH / Eurographics Symp. Comput. Animation*, 2017, pp. 1–8.

[54] J. J. Monaghan, "Smoothed Particle Hydrodynamics," *Rep. Progress Phys.*, vol. 68, no. 8, pp. 1703–1759, 2005.

[55] M. Weiler, D. Koschier, M. Brand, and J. Bender, "A physically consistent implicit viscosity solver for SPH fluids," *Comput. Graph. Forum*, The Eurographics Association and John Wiley & Sons Ltd., vol. 37, no. 2, 2018, doi: 10.1111/cgf.13349.

[56] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, and M. Teschner, "Versatile rigid-fluid coupling for incompressible SPH," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 62:1–62:8, 2012.

[57] J. J. Monaghan, "On the problem of penetration in particle methods," *J. Comput. Phys.*, vol. 82, no. 1, pp. 1–15, 1989.

[58] J. J. Monaghan, "SPH compressible turbulence," *Monthly Notices Roy. Astronomical Soc.*, vol. 335, no. 3, pp. 843–852, 2002.

[59] H. Schechter and R. Bridson, "Ghost SPH for animating water," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 61:1–61:8, 2012.

[60] J. Bender, "SPlisHSPlasH Library," 2017. [Online]. Available: https://github.com/InteractiveComputerGraphics/SPlisHSPlasH

[61] M. Ihmsen, N. Akinci, M. Becker, and M. Teschner, "A parallel sph implementation on multi-core cpus," *Comput. Graph. Forum*, vol. 30, no. 1, pp. 99–112, 2011.

[62] U. Ghia, K. N. Ghia, and C. T. Shin, "High-resolutions for incompressible flow using the navier-stokes equations and a multigrid method," *J. Comput. Phys.*, vol. 48, no. 3, pp. 387–411, 1982.

[63] S. J. Cummins and M. Rudman, "An SPH projection method," *J. Comput. Phys.*, vol. 152, pp. 584–607, 1999.

[64] C. Deul, P. Charrier, and J. Bender, "Position-based rigid body dynamics," *Comput. Animation Virtual Worlds*, vol. 27, no. 2, pp. 103–112, 2014.

[65] J. Bender, "PositionBasedDynamics Library," 2017. [Online]. Available: https://github.com/InteractiveComputerGraphics/PositionBasedDynamics

[66] D. Koschier and J. Bender, "Density maps for improved SPH boundary handling," in *Proc. ACM SIGGRAPH / Eurographics Symp. Comput. Animation*, Jul. 2017, pp. 1–10.

Authorized licensed use limited to: McGill University. Downloaded on March 26,2023 at 13:48:18 UTC from IEEE Xplore. Restrictions apply.

Jan Bender received the diploma, PhD and habilitation degrees in computer science from the University of Karlsruhe. He is a professor of computer science and the leader of the Computer Animation Group, RWTH Aachen University. His research interests include interactive simulation methods, multibody systems, deformable solids, fluid simulation, collision handling, cutting, fracture, GPGPU and real-time visualization.

Dan Koschier received his MSc degree in computational engineering from TU Darmstadt, in 2014. He is working toward the PhD degree at RWTH Aachen University and an associate member with the Graduate School of Computational Engineering, TU Darmstadt. His research interests include physically-based simulation of deformable solids, cutting, fracture, and fluids.

Tassilo Kugelstadt received the BSc degree in physics from JGU Mainz, in 2013 and the MSc degree in computer science in natural sciences from JGU Mainz, in 2015. He is working toward the PhD degree at RWTH Aachen University. His research interests include physically-based simulation of deformable solids, elastic rods, and fluids.

Marcel Weiler received the MSc degree in computer science from the Graduate School of Computational Engineering, Technische Universität Darmstadt, where he is working toward the PhD degree. His research interest include the physically-based simulation of fluids and deformable solids, rendering and machine learning.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.

Authorized licensed use limited to: McGill University. Downloaded on March 26,2023 at 13:48:18 UTC from IEEE Xplore. Restrictions apply.