The Visual Computer
https://doi.org/10.1007/s00371-020-01914-5

ORIGINAL ARTICLE

# Robust turbulence simulation for particle-based fluids using the Rankine vortex model

Xiaokun Wang[1] · Sinuo Liu[1] · Xiaojuan Ban[1] · Yanrui Xu[1] · Jing Zhou[1] · Jiří Kosinka[2]



**Abstract**
We propose a novel turbulence refinement method based on the Rankine vortex model for smoothed particle hydrodynamics (SPH) simulations. Surface details are enhanced by recovering the energy lost due to the lack of the rotation of SPH particles. The Rankine vortex model is used to convert the diffused and stretched angular kinetic energy of particles to the linear kinetic energy of their neighbors. In previous vorticity-based refinement methods, adding more energy than required by the viscous damping effect leads to instability. In contrast, our model naturally prevents the positive feedback effect between the velocity and vorticity fields since the vortex model is designed to alter the velocity without introducing external sources. Experimental results show that our method can recover missing high-frequency details realistically and maintain convergence in both static and highly dynamic scenarios.

**Keywords** Fluid simulation · Vortex model · Turbulence · Smoothed particle hydrodynamics

## 1 Introduction

As one of the most popular approaches to simulating fluids in computer graphics and virtual reality, smoothed particle hydrodynamics (SPH) has been widely used to generate fluid animations with lively details and vivid motions. Although many novel models for animating various materials and enforcing incompressibility have been proposed, much work remains to be done to achieve and enhance realistic visual effects of complex phenomena. For example, simulating turbulent details is still elusive due to numerical dissipation [12] or coarse sampling of grids [20].

Xiaokun Wang and Sinuo Liu have Co-first author and contributed equally.

**Electronic supplementary material** The online version of this article (https://doi.org/10.1007/s00371-020-01914-5) contains supplementary material, which is available to authorized users.

✉ Xiaojuan Ban
banxj@ustb.edu.cn

Jiří Kosinka
j.kosinka@rug.nl

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China

[2] Bernoulli Institute, University of Groningen, Groningen, The Netherlands

The up-res and vortex-based methods have been used to increase the resolution of turbulent fluids. The up-res method is commonly used in Eulerian simulations, using coarse grids for the simulation and increasing resolution via fine turbulence models [21,31]. Vortex-based methods aim to create and preserve turbulence through the vorticity field, which includes the vorticity confinement and Lagrangian vortex methods. The vorticity confinement (VC) method recovers existing vortices and enhances them by adding an extra force [11,24]. However, the VC method tends to add more energy than the fluid dissipates, and only existing vortices can be amplified. The Lagrangian vortex (LV) method builds on the vorticity representation of the Navier–Stokes equations [2,33,34]. Their results show that LV method does not work well in scenarios involving liquids, even ones where there is a great possibility to generate visible vortex structures. Finally, the up-res method usually acts as a postprocessing step and thus does not address the dissipation issue directly.

To solve the problems mentioned above, we utilize the vorticity field to approximate angular velocity of particles and feed it back to the velocity field using the Rankine vortex model. In the ideal setting of the SPH approach, particles used to discretize space are small enough so the dissipation of the angular kinetic energy can be safely ignored, without affecting the overall performance. However, when high efficiency is desired and the particle size is comparatively larger,

Published online: 04 August 2020

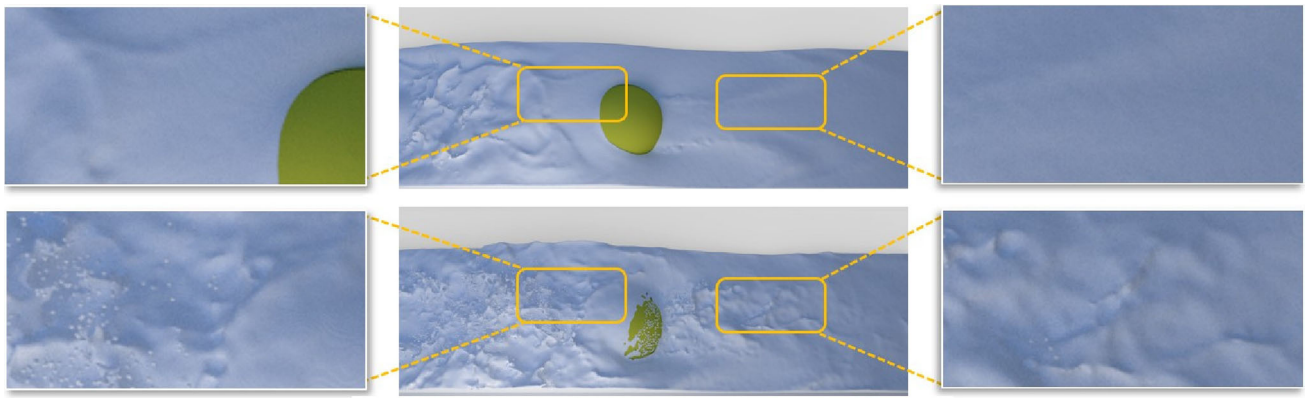


**Fig. 1** A turbulent river simulated with 94 K/s fluid particles flows around a hemisphere using IISPH [17] (top row) and our method (bottom row). Note that abundant turbulence details are generated with our approach

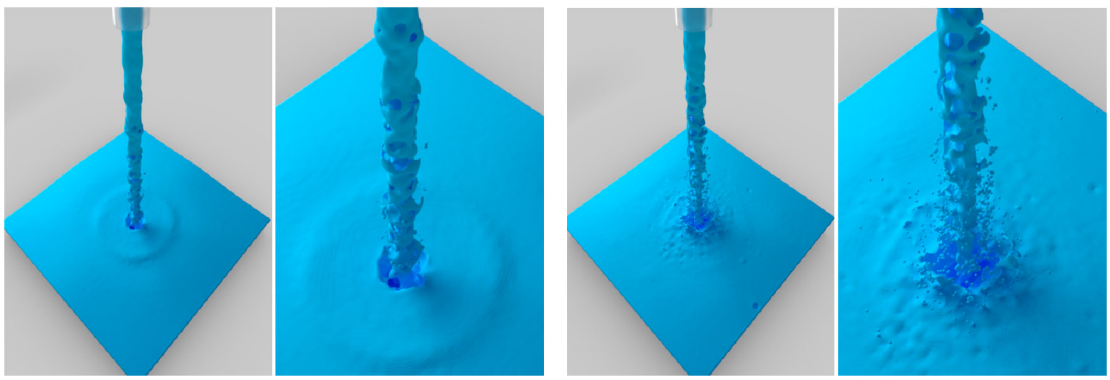



**Fig. 2** Leaking water using IISPH (left) and our method (right), both using 1.4 M fluid particles. Our method exhibits more vortex details and produces more realistic fluid shapes than IISPH

the inertia tensor absent from the equation will result in severe numerical dissipation. Therefore, restoring the missing angular velocity and converting it into linear velocity is an intuitive way to compensate the effect.

To restore high-frequency details, we investigate the relationship between vorticity and angular velocity, and estimate the rotation of the particles through the curl of linear velocity. Although previous methods have attempted to incorporate vorticity into velocity calculations, the introduced positive feedback tends to be too strong in the simulation. When excessive energy is added, it makes simulations easily reach unstable states. This is because the space-time discretizations of SPH often fail to satisfy the conditions of the continuous physical refinement theory. Existing methods thus mistakenly introduce external pressure sources into the simulation, producing additional turbulence rather than recovering it.

In order to purely recover missing details, the vorticity field should remain the same after feedback. As a result, external sources should not be involved, as they could also eliminate the positive feedback. Through a vortex algebraic profile, the induced velocity can be represented by a constant viscous core or a diffusive core which grows with time. The simplest depiction of a viscous vortex is the Rankine model [35], which exhibits key features of a viscous core. Therefore, we regard each particle as the rigid core of a Rankine vortex, and refine the linear velocity of a particle's neighbors inversely proportional to the distance between them. As the velocity of the center in a Rankine vortex is extremely large, the shear makes the viscous core rotate like a rigid body.

Our method takes the fluid dynamic interpretation of turbulence into account. Turbulence and vortex effects can thus be restored without producing unstable results; see Fig. 1. By introducing an adjustable parameter, users are able to control the roughness of the flow. Compared with the state-of-the-art turbulence methods, our method is especially good at handling violent fluids with high stability. Because our method does not introduce external sources into the simulation, its results are natural; see Figs. 2 and 3. The main contributions of our method are:

– A new particle-based turbulence simulation scheme that effectively alleviates numerical dissipation by deriving linear velocity from missing angular velocity;

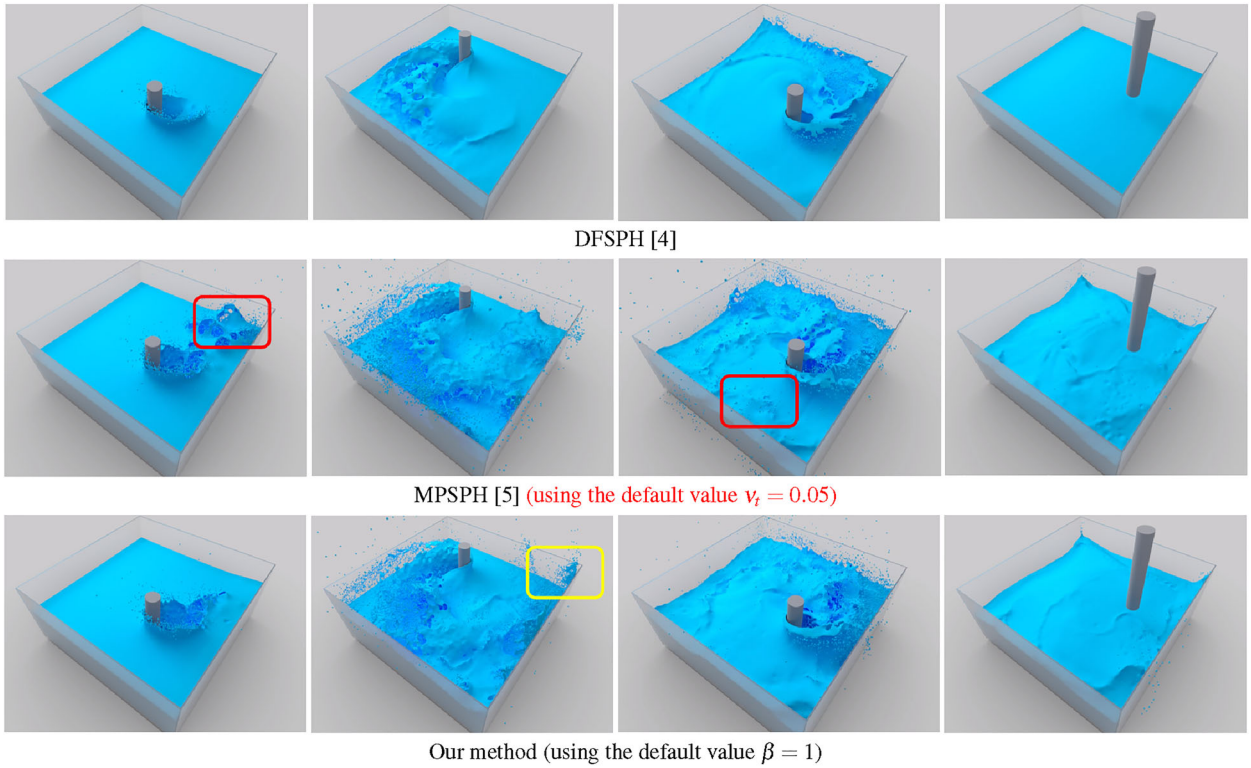



**Fig. 3** A stirring water experiment with 1.48 M fluid particles using DFSPH (top row), MPSPH (middle row) and our method (bottom row). The stick stirs the water (first to third columns), and then the stick is removed (last column). When the stick starts to stir the fluid, our method can effectively model the waves. In contrast, MPSPH does not produce stable results (red regions). Our method produces a nicely organized and vibrant effects, such as when the fluid hits the wall (yellow region)

– A viscous model for vortex flows that creates additional turbulence which can be controlled using a parameter without adding extra energy;
– An upgrade to the traditional SPH fluid framework with turbulence refinement, which can be simply integrated into other particle-based methods and fluid solvers.

The remainder of the paper is organized as follows. Section 2 reviews related work and Sect. 3 recalls the SPH method. Our vortex refinement approach is introduced in Sect. 4. Our experiments and results are presented and discussed in Sect. 5. The paper is then concluded in Sect. 6.

## 2 Related work

Fluid simulation [22] is a hot topic in computer graphics and virtual reality, largely due to the pioneering work by Jos Stam in 1999 [39], but also due to high demand by the entertainment industry. Over the last years, turbulent and detailed fluid simulation has become ever so popular. In this section, we discuss related work concerning the SPH, up-res, and vortex-based methods.

### 2.1 SPH-based fluid simulation

There are two main approaches in fluid simulation: Lagrangian methods and Eulerian methods. The main difference between them is the discretization method of space. As a grid-based approach, Eulerian methods can achieve a simpler flow field representation [13]. When the grid resolution is higher, higher resolution results can be obtained. However, these methods are limited to a fixed grid and thus find it difficult to express rich deformations of fluids. As a particle-based approach, Lagrangian methods are able to simulate fluids with large deformations accurately and efficiently [25].

The SPH method is a typical Lagrangian method, which was first proposed in the field of astrophysics [27]. In 2003, Müller et al. [30] introduced this idea to the fluid simulation community by using the ideal gas state equation with surface tension and viscosity forces. In order to enhance the simulation results, Becker and Teschner [3] proposed a weakly compressible SPH method by using the Tait equation. Solenthaler and Pajarola [38] proposed a new scheme for efficiency and incompressibility by improving pressure calculations. Ihmsen et al. [17] then proposed to apply an implicit method to obtain a more accurate velocity field. Recently, Bender

and Koschier [4] proposed to add an additional constraint to further improve the incompressibility and the accuracy of the calculation. An overview of SPH-based fluid simulations can be found in the reports of Ihmsen et al. [18] and Koschier et al. [22]. To date, the general simulation of fluids has been successfully and realistically represented by the SPH method and its variants. Nevertheless, a high demand for enhanced fluid simulations still exists, such as those able to capture detailed turbulence.

## 2.2 The up-res method

The up-res method is a simple and efficient method to enrich the apparent resolution of flow fields [42]. Its main idea is to up-sample under a coarse discretization.

### 2.2.1 The traditional up-res method

As a representative of Lagrangian methods, Shao et al. [37] ran a high-resolution simulation over a Rankine surface based on the velocity field. Mercier et al. [28] proposed a postprocess method by adding entirely new dynamics, named wave simulation, to the tracked surface [43,51]. These two methods could be used together to obtain a richer simulation result.

Among Eulerian methods, the velocity field [31] was used to derive high-resolution surface movement. Edwards and Bridson [10] combined the adaptive discontinuous Galerkin method with surface meshes, which retains much of the data structure simplicity of regular grids.

### 2.2.2 Machine learning approaches

Recently, machine learning has been introduced to synthesize turbulence of fluids [6,36]. Machine learning methods greatly speed up the simulation process, and the simulation resolution is the same as with traditional methods, but a huge amount of time is required for network training. In addition, machine learning methods are well suited for Eulerian methods as the flow field is discretized with a uniform, high-resolution grid. However, it is very difficult to combine machine learning with Lagrangian methods due to the unstructured nature of the data.

## 2.3 Vortex-based methods

Vortex-based methods aim to create and preserve turbulence by adding a vorticity field. They can be split into three main categories: vorticity confinement, Lagrangian vortex methods, and the vortex particle method.

### 2.3.1 Vorticity confinement

This method was first proposed in the field of Computational Fluid Dynamics (CFD) [40] and was introduced to computer graphics by Fedkiw et al. [11]. Existing vortices were enhanced by adding a new force, which is applied to the velocity field. In 2010, Zhu et al. [54] proposed to combine vorticity confinement with SPH by using a high-resolution overlapping grid. This method can create and preserve turbulence at solid boundaries. However, vorticity confinement can only enhance existing vortices.

### 2.3.2 Lagrangian vortex methods

These methods can be implemented using surfaces, sheets, filaments or particles. They are naturally divergence-free and inherently immune to numerical dissipation. The vortex sheet method was first proposed in the field of (CFD) [41]. In 2010, Weißmann and Pinkall [49] introduced this method to smoke simulation. In 2014, Weißmann et al. [50] proposed a filament-based computation method to represent the vortices in the flow field. Eberhardt et al. [9] improved this method by adding an algorithm to automatically extract a set of hierarchical vortex filaments. However, this method tends to be inaccurate due to the vortex field being represented by a very limited number of filaments.

### 2.3.3 The vortex particle method

This method is suitable for particle-based fluid simulations. In 2015, Zhang et al. [52] proposed a hybrid vorticity and velocity advection scheme to reduce numerical dissipation. However, this method does well only in smoke simulations and is not suitable for liquids. In 2018, Bender et al. [5] proposed a unified particle method, called the micropolar SPH (MPSPH) method, to simulate turbulent fluids with impressive results. Our method can be regarded as a vortex particle method.

### 2.3.4 Our method

To restore the energy that dissipates during SPH simulations due to discretizations, many efforts have been made to apply the curl of velocity to capture it accurately [15]. However, such methods are not suitable enough for SPH because of the excessive positive feedback between the two fields, causing numerical instability. Mathematicians and physicists have proposed many intricate models for describing high-resolution fluid motions [8]. A complete description of vortices demands the solution of the full Navier–Stokes equations, to which an analytical solution is impossible. By simplifying the governing equations, several empirical models have been proposed [44,45]. Through a desingularized

algebraic profile, the induced velocity can be represented by a constant viscous core or a diffusive core growing with time. The simplest depiction of a viscous vortex is the Rankine model [35], which is exactly what we use.

A Rankine vortex has no circulation (i.e., the line integral of the velocity field) along any closed contour that does not enclose the vortex axis. It rather has a fixed value for any contour which includes the axis [7]. The Kaufmann vortex [19] avoids the singular nature of the Rankine model. Lamb [23] and Oseen [32] then proposed a model to describe one-dimensional Navier–Stokes equations. Vatistas et al. [45] successfully found an algebraic way to represent the Lamb-Oseen model. We apply the Rankine vortex model to recover the energy lost in rotational degrees of freedom (DOFs). The time-varying viscous core in the Lamb–Oseen vortex model further proves that the Rankine vortex model is fully applicable to the SPH approach, as we detail below.

## 3 Smoothed particle hydrodynamics in fluid dynamics

SPH is a computational method for simulating continua such as fluids, although it was proposed initially to attack problems in the field of astrophysics [14,27].

In SPH, the fluid is divided into a set of discrete, moving particles whose Lagrangian properties enable the linear velocity of the particle with index $i$ to be given as

$$\boldsymbol{v}_i = \frac{\mathrm{d}\boldsymbol{x}_i}{\mathrm{d}t}, \tag{1}$$

where $\boldsymbol{x}_i$ is the position of the $i$-th particle, and $\mathrm{d}t$ is the time-step used in the simulation.

Interaction between particles happens through a user-defined kernel function $W$ with a specific smoothing length $h$. By summing up all the properties within the supporting radius of the smoothing kernel, all physical quantities of particles are obtained using $W$. An arbitrary field $A$ can thus be expressed as:

$$A(\boldsymbol{x}) = \sum_j m_j \frac{A_j}{\rho_j} W(\|\boldsymbol{x} - \boldsymbol{x}_j\|, h), \tag{2}$$

where $j$ iterates over all neighbor particles with $m_j$ the mass and $\rho_j$ the density of the $j$th particle. By setting $A$ as $\rho$, the density at location $\boldsymbol{x}_i$ is

$$\rho(\boldsymbol{x}_i) = \sum_j m_j W. \tag{3}$$

For evaluating the derivatives of a field quantity $A$, the gradient of $A$ using SPH is simply

$$\nabla A(\boldsymbol{x}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(\|\boldsymbol{x} - \boldsymbol{x}_j\|, h). \tag{4}$$

The Navier–Stokes equations for the Lagrangian case can be written as:

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} = -\frac{1}{\rho}\nabla p + \mathbf{g} + \nu\nabla^2\boldsymbol{v}, \tag{5}$$

where $\mathbf{g}$ represents gravity and $\nu$ is the kinematic viscosity coefficient (we use $\nu = 0.05$ and artificial viscosity [29] in our implementation). To obtain the linear velocity from the acceleration and the time-step accurately, we employ an SPH symplectic integrator, namely the leapfrog scheme [22].

## 4 Vorticity refinement

In traditional SPH simulations, the forces affecting particles are the pressure force, gravity, and viscosity, and used to ensure stability and incompressibility. To further capture the dynamics of the fluid, viscosity is often taken into account only when the relative velocities between particles are less than 90 degrees apart in direction. However, spacial discretiszation for fluid simulation introduces macro-angular velocity for particles, and shear is not considered to affect the linear velocity field, which leads to a loss in dynamic performance, especially for turbulence effects.

### 4.1 Numerical dissipation in SPH

Striking a balance between efficiency and accuracy has always been an important issue in fluid simulations. To prevent the loss of rotational kinetic energy when pursuing efficiency, we add angular velocity as a three-dimensional attribute to each fluid particle, through which the missing rotational kinetic energy can be restored. Moreover, high frequency details can be recovered by the refinement of linear velocity from angular velocity. This is shown in Fig. 6, where each particle in the flow is regarded as a single vortex.

Many vortex models have been recently produced to analyze swirl, axial, and radial components of the velocity induced by a viscous trailing tip vortex. In this paper, we incorporate tangential viscosity from angular velocity, and thus vortex models are ideal to capture the required rotational influence. To recover the kinetic energy lost in rotational DOFs due to spacial discretization, we apply the Rankine vortex model and an adjustable viscous core to utilize the change of angular velocity of every particle at each time-

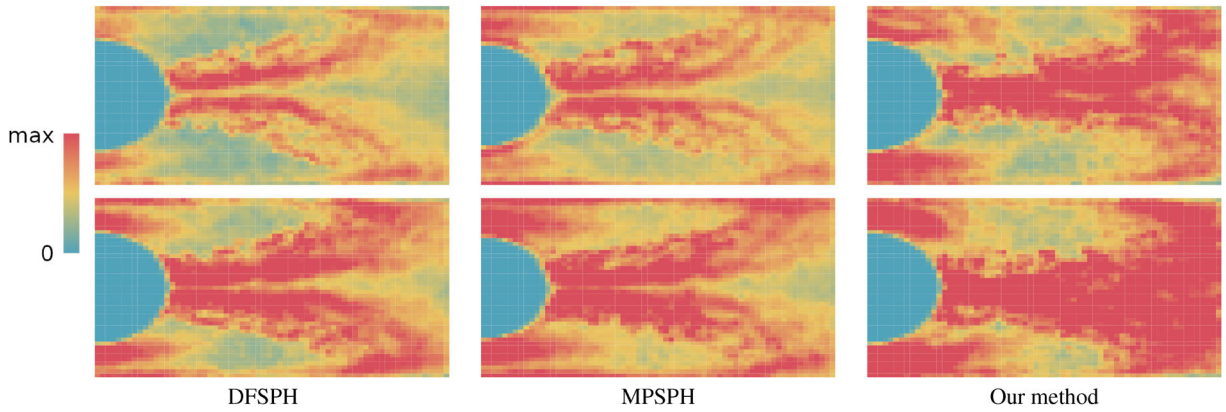



**Fig. 4** A comparison of vorticity in 2D on two frames (see the supplementary material for the full video). We use the magnitude of the vorticity of the particles in the visualization. Blue corresponds to no vorticity and red to large vorticity. The fluid flows around a pillar (semi-disk; blue area to the left) from left to right. This contrasts the vorticity among DFSPH, the MP solver, and our VR method

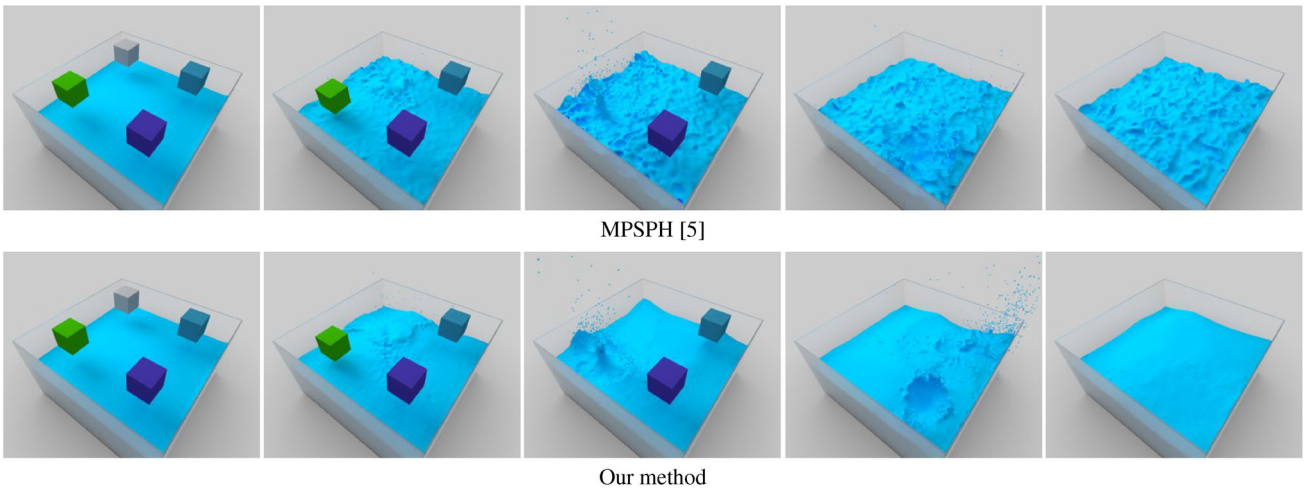



**Fig. 5** Four blocks falling into water, one by one, with 1.08M fluid particles using MPSPH and our method. This simulation comparison shows the superiority of our method over MPSPH. Our method maintains energy convergence (see the last column) while recovering turbulence (second to fourth columns)

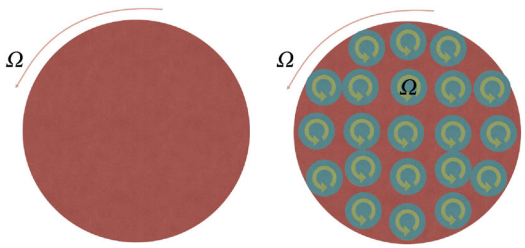



**Fig. 6** A two-dimensional disk spinning around its centre of mass with an angular velocity of $\Omega$. On the right, the disk is discretized into small particles for an SPH approach. However, traditional SPH approaches neglect the self-spinning of these particles, which leads to numerical dissipation

step to create turbulence from the shear generated by angular velocity.

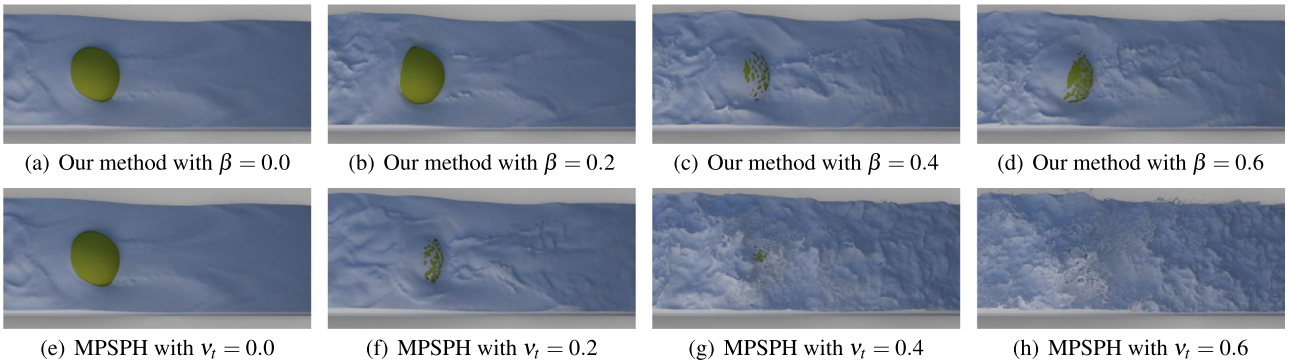
We derive the angular velocity $\Omega$ through the vorticity field $\boldsymbol{\omega}$ as $\Omega = \boldsymbol{\omega}/2$. We compute the vorticity field using the differential form as:

$$\boldsymbol{\omega}_i = \nabla \times \boldsymbol{v}_i = \frac{1}{\rho_i} \sum_j m_j (\boldsymbol{v}_i - \boldsymbol{v}_j) \times \nabla_i W_{ij}. \tag{6}$$

In the view of efficiency, we obtain the dissipated vorticity using the difference between two time-steps $n$ and $n-1$ as:

$$\delta \boldsymbol{\omega}_i^n = \boldsymbol{\omega}_i^n - \boldsymbol{\omega}_i^{n-1}. \tag{7}$$

The angular velocity used to refine the linear velocity should be negative wrt. the dissipated angular velocity because the vorticity dissipation is caused by the neighbor particles; this implies a negative factor to the vortex centre.



(a) Our method with $\beta = 0.0$ (b) Our method with $\beta = 0.2$ (c) Our method with $\beta = 0.4$ (d) Our method with $\beta = 0.6$

(e) MPSPH with $\nu_t = 0.0$ (f) MPSPH with $\nu_t = 0.2$ (g) MPSPH with $\nu_t = 0.4$ (h) MPSPH with $\nu_t = 0.6$

**Fig. 7** A simulation using 94 K/s turbulent fluid particles with an inserted hemisphere modeled with our method and MPSPH with different parameter values. Top row: Our method with the control parameter $\beta = 0.0$, 0.2, 0.4 and 0.6. Bottom row: MPSPH with the transfer coefficient $\nu_t = 0.0$, 0.2, 0.4 and 0.6. The turbulence effects increase with the respective parameter value. Observe that our method can not only achieve more noticeable effects than MPSPH, but it also ensures stability

Improving on Wang et al. [48], a more accurate form of vorticity dissipation for each particle can be derived from the curl of Navier–Stokes equation as:

$$-\frac{\mathrm{d}\boldsymbol{\omega}_i}{\mathrm{d}t} = (\boldsymbol{\omega}_i \cdot \nabla)\boldsymbol{v}_i + \nu\nabla^2\boldsymbol{\omega}_i. \tag{8}$$

However, $\boldsymbol{v}_i$ is derived after the projection step, which means the velocity refinement should take place only after that. Therefore, fluctuations in the density field could potentially jeopardize the stability of the simulation, and at an excessive computation overhead. We therefore proceed differently.

### 4.2 The Rankine vortex model

A two-dimensional Rankine vortex model can be expressed in polar coordinates as:

$$V_\theta(r) = \begin{cases} \left(\frac{\Gamma}{2\pi r_\mathrm{c}}\right)\bar{r} & 0 \le \bar{r} \le 1, \\ \left(\frac{\Gamma}{2\pi r_\mathrm{c}}\right)\frac{1}{\bar{r}} & \bar{r} > 1, \end{cases} \tag{9}$$

where $\Gamma$ is a fixed circulation, and $\bar{r} = r/r_\mathrm{c}$ with $r$ the radial distance and $r_\mathrm{c}$ the radius of the viscous core. Although an idealized free vortex has a linear velocity inversely proportional to the distance from the center, a more accurate approximation in the real world should take also shear into account, which prevents the linear velocity to be infinitely large near the vortex center. This profile simplifies a natural vortex to a rigid-like core and a vorticity-free external area. This reduces the computational cost due to its linear nature.

A more precise and suitable description of the time-variant radius of the viscous core is given in the vortex model by Lamb and Oseen, which is a solution to the one-dimensional Navier–Stokes equations [23,32]. It grows with time as:

$$r_\mathrm{c} = \sqrt{4\alpha\nu t}, \tag{10}$$

where $\alpha$ is the Oseen parameter with a value of 1.25643. The scale of the viscous core should be carefully chosen when using the Rankine vortex to match the scales of the time and space discretization.

Considering the time-step $\Delta t$ and the kinematic viscosity $\nu$ of the fluid, the largest possible viscous core used in each simulation step is $r_{\mathrm{c}_{\max}} = \sqrt{4\alpha\nu\Delta t}$, and the smallest is $r_{\mathrm{c}_{\min}} = 0$. According to the Rankine vortex model, the influence of the vortex reaches its peak with $r_{\mathrm{c}_{\max}}$, and has no effect at $r_{\mathrm{c}_{\min}}$. In order to enable our users to alter the performance directly, we introduce an adjustable viscous coefficient $\beta$ to make our method controllable [23]

$$r_\mathrm{c} = \beta r_{\mathrm{c}_{\max}} = \beta\sqrt{4\alpha\nu\Delta t} \tag{11}$$

with $0 \le \beta \le 1$. The default value of $\beta = 1$ is used, unless stated otherwise. The influence of this parameter is discussed in Fig. 7.

### 4.3 Turbulence feedback using the Rankine vortex

In order to recover the missing details in a simulation, we transform the angular velocity to let it reasonably affect the linear velocity. Linear velocity refinement using vorticity has been widely used in lots of previous Lagrangian vortex methods. However, many of them tend to be unstable, especially when the movements are intense, even though they are theoretically stable in the continuous model. This vorticity regeneration problem that occurs in the SPH approach has not been considered in detail previously; see Fig. 8.

Vorticity regeneration in the refining process occurs when the velocity field is enhanced after a time-step in which vorticity is refined. However, the energy would increase exponentially if not subject to a certain viscosity constraint. To avoid instability, the refined velocity field must not feed extra energy back to the vorticity field. But when using dis-

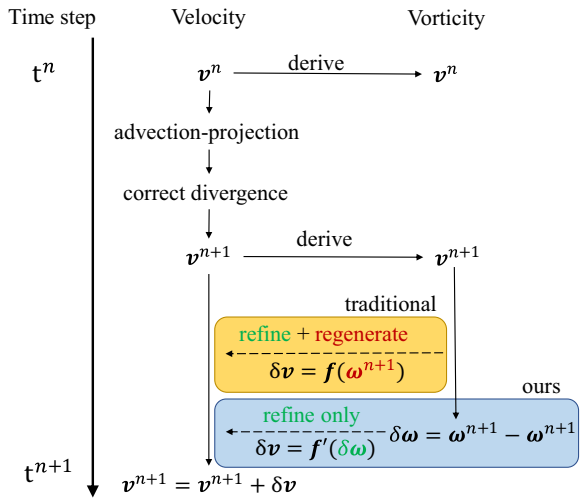



**Fig. 8** Traditional Lagrangian vortex methods (yellow) utilize the vorticity field directly to refine the missing linear velocity. This refinement regenerates excessive energy in the vorticity at the next time step (red). Instead, in our scheme (blue) we refine the dissipated vorticity field (green) to avoid this issue

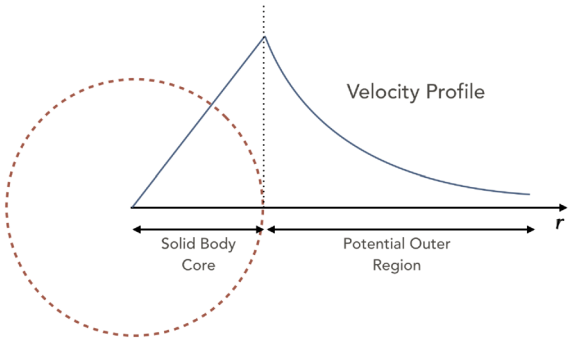



**Fig. 9** A solid core in the two-dimensional Rankine vortex. We consider a fluid particle as the core, and outside the core the velocity is decreased inverse-proportionally to the distance $r$ to the vortex center

cretizations, even a naturally convergent method could result in severe instabilities.

In theory, a flow can be divided into forced flow and natural flow. Forced flow is the flow in which pressure is affected by an external source. When the velocity field is refined as forced flow, the vorticity field is affected in return and extra energy is generated due to additional pressure. Thus the Rankine vortex model must be applied to enforce convergence, i.e., so that it does not generate excess energy.

Circulation can be used to describe the strength of a vortex. Circulation-free means that the Rankine vortex does not generate additional velocity that changes the vorticity. However, in this Rankine vortex model, the velocity of the vortex center is infinite, and thus unrealistic. However, the shear near the origin is large enough to slow the velocity down, creating a rigid-like core that strikes a balance between shear and velocity, as shown in Fig. 9.

The linear velocity on the surface of the viscous core can be obtained from Eq. (9) as

$$\boldsymbol{v}_i^{\mathrm{sur}}(\mathbf{e}_\theta) = V_\theta(r_\mathrm{c})\mathbf{e}' = \left(\frac{\Gamma}{2\pi r_\mathrm{c}}\right)\mathbf{e}', \tag{12}$$

where $\mathbf{e}_\theta$ is the unit vector pointing at a specific point on the viscous core surface, and $\mathbf{e}'$ is the unit vector representing the direction of velocity of that point, subject to $\mathbf{e}' \cdot \mathbf{e}_\theta = 0$ and $\mathbf{e}' \cdot \boldsymbol{\omega}_i = 0$. Consequently, the refinement from particle with index $i$ to that with index $j$ can be expressed as:

$$\delta\boldsymbol{v}_{i\to j}(\mathbf{e}_\theta) = V_\theta(r)\mathbf{e}' = \left(\frac{\Gamma}{2\pi r_\mathrm{c}}\right)\frac{1}{\bar{r}}\mathbf{e}' = \frac{r_\mathrm{c}}{r}\boldsymbol{v}_i^{\mathrm{sur}}(\mathbf{e}_\theta), \tag{13}$$

where $r$ with $r > r_\mathrm{c}$ is the distance between the involved particles, and $\mathbf{e}_\theta$ points from $\mathbf{x}_i$ to $\mathbf{x}_j$.

At the same time, due to the relationship between angular velocity and linear velocity, the linear velocity on the surface of the viscous core of the $i$th particle can be expressed as:

$$\boldsymbol{v}_i^{\mathrm{sur}}(\mathbf{e}_\theta) = \frac{\delta\boldsymbol{\omega}_i}{2} \times (r_\mathrm{c}\mathbf{e}_\theta), \tag{14}$$

where $r_\mathrm{c}\mathbf{e}_\theta$ is the displacement from the viscous core center to its surface. Thus, Eq. (13) is converted to

$$\delta\boldsymbol{v}_{i\to j}(\mathbf{e}_\theta) = \frac{r_\mathrm{c}^2}{2r}(\delta\boldsymbol{\omega}_i) \times \mathbf{e}_\theta. \tag{15}$$

When applying our method with the fluid-rigid coupling method introduced by Akinci et al. [1], the boundary particles contribute no vorticity to the fluid since boundary particles carry no angular velocity and the overall vorticity field would become unstable near the rigid body, especially for moving ones. So the vorticity is computed within the fluid only.

This method can be regarded as a velocity factor in the advection step of the SPH simulation, thus causing no density fluctuations in normal SPH approaches. We integrated our method with IISPH [17] and DFSPH [4], showing good results; see Sect. 5.

In the SPH approach, the radius of a viscous core growing with time needs to be less than the radius of the fluid particles in the simulation. Otherwise, the refinement is not captured by the system. Fortunately, this is easy to achieve for fluids with a high Reynolds number, while low Reynolds number fluids do not cause much turbulence.

The whole process of refinement is summarized in Algorithm 1, including the steps adapted from the IISPH and DFSFH methods.

**Algorithm 1:** Robust turbulence simulation using the Rankine vortex model

1: // Vortex-based turbulence ($\alpha = 1.25643,\ 0 \le \beta \le 1$)
2: $\boldsymbol{\omega}^n = \nabla \times \boldsymbol{v}^n$
3: $\delta\boldsymbol{\omega} = \boldsymbol{\omega}^n - \boldsymbol{\omega}^{n-1}$
4: $r_c = \beta\sqrt{4\alpha\nu\Delta t}$
5: $\delta\boldsymbol{v}_i = \sum_j \frac{r_c^2}{2r}(\delta\boldsymbol{\omega}_i) \times \mathbf{e}_\theta$
6: $\boldsymbol{v}^n = \boldsymbol{v}^n + \delta\boldsymbol{v}$
7: // Traditional SPH method with the scheme of IISPH
8: $\tilde{\boldsymbol{v}} = \text{Advect}(\boldsymbol{v}^n)$
9: $\hat{\boldsymbol{v}} = \text{Project}(\boldsymbol{v}^n)$
10: $\boldsymbol{x}^{n+1} = \Delta t\hat{\boldsymbol{v}} + \boldsymbol{x}^n$
11: // Correction based on DFSPH
12: $\boldsymbol{v}^{n+1} = \text{CorrectDivergence}(\hat{\boldsymbol{v}})$

## 5 Experiments and discussion

In this section, we compare the simulation results of our method to several fluid and turbulence simulation methods. We integrated MPSPH [5] and our method with the DFSPH approach [4] (Figs. 3, 4) and the IISPH approach [17] (all other figures). In addition, we employed the neighborhood search algorithm presented by Ihmsen et al. [16], the boundary handling method proposed by Akinci et al. [1] and the surface tension model addressed by Wang et al. [46]. The physics simulation framework was implemented in C++, the fluid surfaces were constructed using anisotropic method [47] and visual results were rendered with Blender. We conducted our experiments on a workstation with Intel Xeon E5-2687 v4 (15M Cache, 3.5 GHz, 4 cores) CPU, 80GB RAM, and NVIDIA Quadro P4000 GPU.

In order to compare with the MP method, we set the parameters for comparison based on the second law of thermodynamics. The micropolar model can be described using the following equations:

$$\begin{aligned} \frac{D\boldsymbol{v}}{Dt} &= -\frac{1}{\rho}\nabla p + \nu\nabla^2\boldsymbol{v} + (\nu+\nu_t)\nabla\times\boldsymbol{\omega} + \mathbf{g} \\ \epsilon\frac{D\omega}{Dt} &= \nu_t(\nabla\times\boldsymbol{v} - 2\omega) + \frac{\vartheta}{\rho} \end{aligned} \tag{16}$$

where $\nu_t$ is the kinematic transfer coefficient (set as $\nu_t = 0.05$ unless stated otherwise), $\boldsymbol{\omega}$ is the angular velocity, $\epsilon$ is an isotropic micro-inertia coefficient, and $\vartheta$ denotes the Levi–Civita tensor and external torque. According to Eq. 16, the micropolar method increases the kinetic energy by introducing $(\nu+\nu_t)\nabla\times\boldsymbol{\omega}$ to the Navier–Stokes equations. In cases where $2\boldsymbol{\omega} = \nabla\times\boldsymbol{v}$, the term $\nu\nabla^2\boldsymbol{v} + (\nu+\nu_t)\nabla\times\boldsymbol{\omega}$ in Eq. 16 can be rewritten as $\frac{1}{2}(\nu-\nu_t)\nabla^2\boldsymbol{v}$. This means it potentially violates the second law of thermodynamics and causes instability if $\nu_t$ is greater than $\nu$. Since the coefficient is generally set to $\nu = 0.05$, we naturally choose $\nu_t = 0.05$ for the MP solver, which corresponds to $\beta = 1$ in our method.

### 5.1 Comparison and convergence

In a classical scenario in Fig. 7, 94 K fluid particles run over a hemisphere in a tunnel, and rich turbulence details are expected to arise. We compared our method with MPSPH [5] with various coefficient settings in this scenario. When $\beta = 0$ or $\nu_t = 0$, both are equivalent to the IISPH method. In this case, some vague and feeble turbulence is produced when the fluid is injected into the tunnel. Although the fluid is then partially blocked by the hemisphere, no significant interaction details are visible around the hemisphere. Behind the obstacle, two regular shallow traces gradually disappear along the flow. At the right end, the current restores calm.

In this experiment, both our method and MPSPH are capable of increasing turbulence details. As $\beta$ increases, the flow gets more violent at the injection port and the traces become more irregular and apparent. The fluid gets stacked in front of the obstacle using both methods, and the hemisphere is nearly fully submerged. Compared with our method, MPSPH increases the turbulence in a fiercer way. The fluid behavior with $\beta = 0.4$ looks as furious as that of $\nu_t = 0.2$. We choose $\beta = 0.6$ throughout this paper as it better shows the effect in our tests. Note that MPSPH increases turbulence by producing lots of small fragments, but our method tends to preserve the original form at a macroscopic level, and to intensify dissipated curves. Moreover, the flow at the right end looks too intense when applying MPSPH.

In Fig. 4, 1.5 K/s fluid particles flush a pillar using DFSPH, MPSPH and our method. We visualize vorticity magnitude of the particles. Observing the wake behind the pillar, our method recovers energy from numerical dissipation. Compared to the MP solver, our method is able to recover more vorticity and thus also detailed turbulence.

Figure 10 shows the disturbance caused by four blocks falling into a tank of water at the same time. We compare our method with MPSPH to assess convergence under extreme conditions. Compared to the fluid, the density of these blocks is considered large enough so that the acceleration of the blocks should be maintained as gravity acceleration throughout the experiment. Significant turbulence can be observed in both methods when the blocks hit the surface. But when they reach the bottom, the generated effects start to vary. One significant difference between MPSPH and ours is that turbulence in our method smooths out over time, while unstable movements appear and do not vanish in MPSPH.

Figure 11 illustrates the total kinetic energy of the blocks-falling-into-water scenario. Both MPSPH and our method accumulate more kinetic energy than IISPH. Our method carries a lot of kinetic energy at 1.6 s into the simulation. In MPSPH, the positive feedback effect eventually overcomes the damping effect, and starts to grow until a dynamic equilibrium is reached instead of a static state. In contrast, our method converges to the stable state, just as IISPH does.

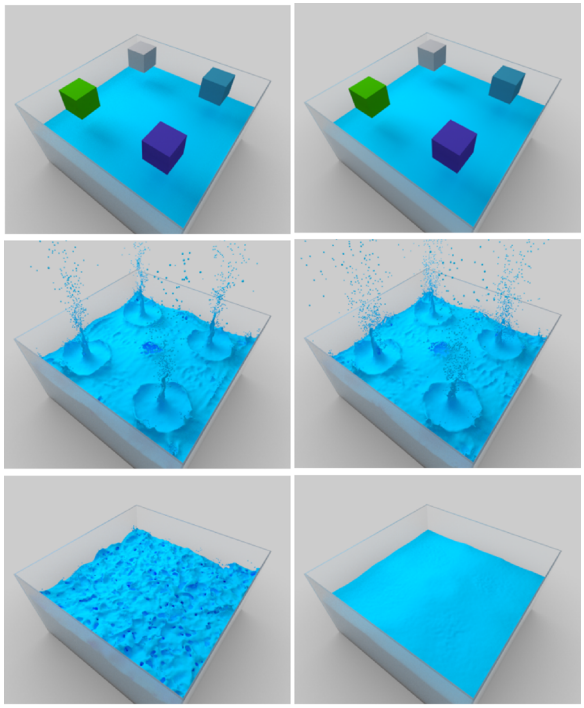



**Fig. 10** Four blocks fall into the fluid simultaneously. The gravitational potential energy of the blocks is transformed into the kinetic energy of the fluid. Our method produces a smooth surface over time, whereas MPSPH keeps producing unrealistic turbulence. Top row: $t = 0$; blocks are dropped. Middle row: $t = 1.94$ s; blocks just under the surface. Bottom row: $t = 11.1$ s; blocks reach the bottom of the container

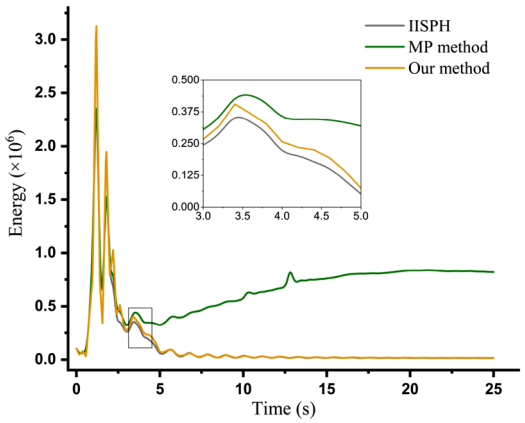



**Fig. 11** Total energy comparison of three methods: IISPH, MPSPH (MP method), and ours

Splitting the linear and the rotational kinetic energy apart as shown in Fig. 12b, we find that the rotational kinetic energy in our method keeps the same pace as the linear kinetic energy. In MPSPH, as the rotational kinetic energy and the linear kinetic energy increase, they stimulate each other, as revealed in Fig. 12a, which results in the instability of the simulation. This also explains that MPSPH cannot maintain the original shape on the macro level. It generates extra turbulence and destroys areas that do not need refinement.

In another experiment, 1.4M fluid particles flow into a tank through a funnel; see Fig. 2. As the fluid is falling, gravity plays a major role in the conventional SPH model, which means the shape of the fluid remains generally the same throughout the descent. In real life, however, the liquid that flows out of a funnel starts out tight and orderly, then widens and disperses via stretching and acceleration of the fluid under gravity. The Weber number (WE), which represents the relationship between fluid velocity and surface tension, decreases as the fluid falls. When the WE reaches a specific threshold, fluids disperse. Compared to IISPH, our solver is more realistic, showing correct behavior of surface tension and shear stress by making the fluid split when falling.

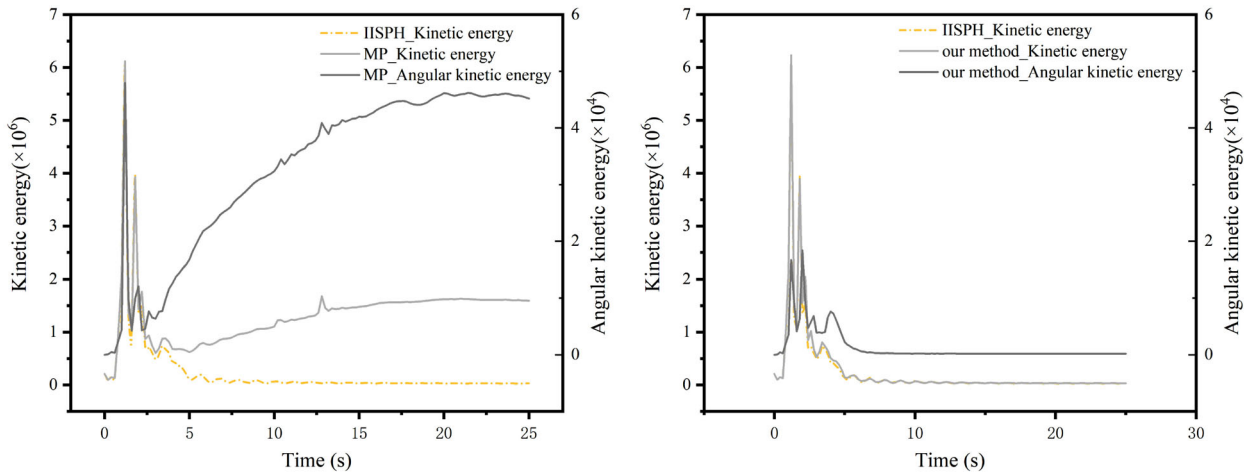
The vorticity field in this scene benefits from the velocity difference under gravity in relation to the WE. Because of energy dissipation and the loss of angular velocity, neither waves nor splashes were preserved in IISPH. In contrast, our solver is capable of generating water splashes and producing reasonable turbulence over a Rankine surface. These high frequency details are very crucial for this type of scenes.

## 5.2 Turbulence performance

In order to demonstrate the turbulence performance of our method, we compare the visual effects of our method, MPSPH, IISPH and DFSPH in this section.

In Fig. 3, a cylindrical stick was inserted into a tank of water, and stirred at a uniform speed for eight seconds. The water splashed around due to the quick movement of the stick. Observe that the simulation result under our method and MPSPH is more delicate than DFSPH. This is due to the recovery of the dissipated velocity by vorticity, while maintaining a certain amount of energy. When the stick starts to stir the fluid, our method can effectively enhance the waves, while MPSPH introduces velocity instabilities (red regions). In the second column, both of the refinement methods are able to generate vivid dynamic effects. Our method tends to behave in a more organized way and has a more vibrant effect when the fluid hits the wall (yellow region). After the stirring process, the stick is pulled out of the fluid, and the water starts to calm down. The traditional SPH approach calms the fluid down quickly due to numerical dissipation. In contrast, the refinement methods are capable of retaining the detailed waves for a longer period of time. Observe that overall, our method produces a more natural result than MPSPH.

In a similar scenario to that shown in Fig. 10, four blocks fall into the water one by one, intensely impacting the water due to their potential energy; see Fig. 5. The drop of each piece further increases the system's energy and creates new turbulence details at the corresponding location. It can be seen that both our method and MPSPH produce rich turbu-




**(a)** Energy comparison of IISPH and MPSPH **(b)** Energy comparison of IISPH and our method

**Fig. 12** Energy comparison among three methods. **a**, **b** Relationship between the linear and rotational kinetic energy of these methods. MPSPH shows a strong positive feedback effect, while our method manages to eliminate it

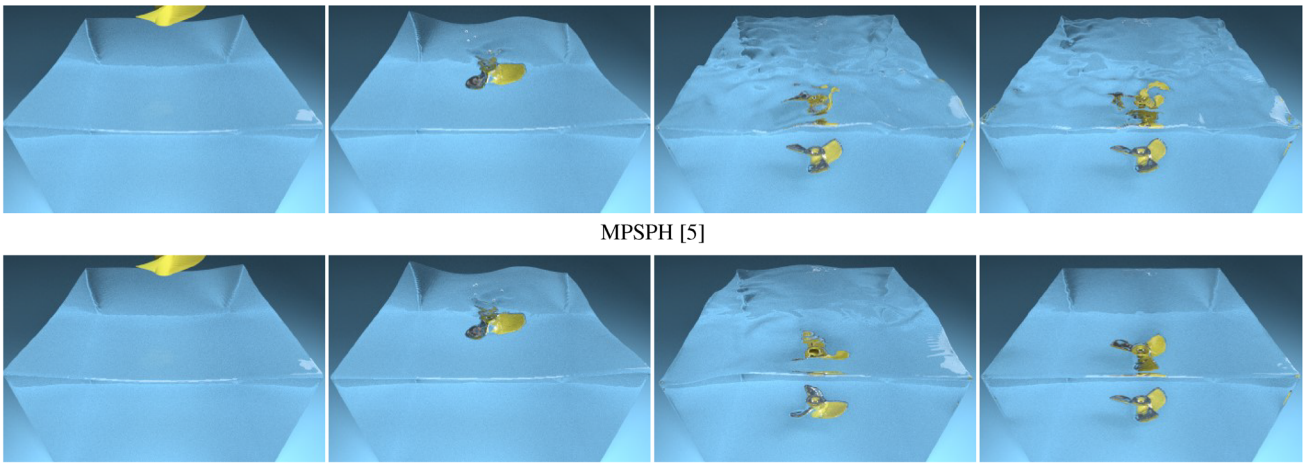



**Fig. 13** A propeller interacts with water using 1.13 M fluid particles. It first descends slowly (first and second columns) and then rotates clockwise at 5 radians per second (third column). Finally, the propeller stops rotating and the water is expected to gradually calm down (fourth column). From left to right column, the simulation times are $t = 0$, $t = 16$ s, $t = 37$ s and $t = 54$ s

lence effects, but in different styles and characteristics. The turbulence effect of MPSPH is very fragmented and small, while our method produces an orderly and stable result. This further demonstrates that our method is suitable for simulating a fierce collision: It can maintain convergence while restoring turbulence, without adding too much energy.

In Fig. 13, a propeller was slowly submerged into water and rotated clockwise, at a speed of 5 radians per second. In the aspect of restoring the turbulence effect, both MPSPH and our method can clearly capture the turbulence of the fluid surface. The turbulence recovered by our method shows a trend of clockwise movement on a macroscopic level, which is more in line with reality. By contrast, the turbulence effect produced by MPSPH is very fragmented, lacking macroscopic motion. Although the details in MPSPH are more abundant, we have demonstrated in previous comparisons that this is at a cost of energy surge. Therefore, our method can not only restore a certain degree of turbulent effects, but also maintain macroscopic stability. This is further confirmed by the last column in Fig. 13.

All combined, the experiments presented in the paper and the supplementary videos show that our method can simulate turbulence realistically with good convergence, producing vivid, and turbulent details and capable of full stabilization.

## 6 Conclusion

We have presented an SPH-based method for recovering turbulence details for low-viscosity incompressible fluids. Built on the Lagrangian modelling approach, our method can be easily integrated into any SPH method with negligible computational overhead. It draws on the fluid dynamic interpretation for turbulence as multiple unstable vortices interact with each other. According to the unconditional stability of potential flow in the SPH simulation process, we showed that the Rankine vortex model is preferable to adjust the linear velocity and to generate small vortices in the vicinity of each particle.

Numerical dissipation is an inherent disadvantage of SPH methods, especially when it comes to simulating turbulence. It is difficult to find an intuitive way to eliminate dissipation completely. Our method successfully resolves the instability effects when the positive feedback exceeds the convergence limit of viscous damping. Moreover, it is fully capable of creating abundant turbulence effects, but also of maintaining stability in extreme scenarios.

Nevertheless, turbulence is an extremely complex phenomenon, which is often accompanied by bubbles and white foam [26]. In this paper, we have only enhanced the turbulence effect from the perspective of energy loss. We do not cover gas–liquid coupling. Focusing on multi-phase coupling, especially gas–liquid coupling, to achieve richer turbulence details remains a challenging avenue for future research. Besides, extend this method to Machine Learning methods [53] are also of great importance and very challenging.

**Acknowledgements** This research was supported by the National Natural Science Foundation of China (61702036, 61873299), the National Key Research and Development Program of China (2016YFB1001404), the Finance Science and Technology Project of Hainan Province (ZDYF 2019009) and the Fundamental Research Funds for the Central Universities of China (FRF-TP-19-043A2).

### Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

1. Akinci, N., Ihmsen, M., Akinci, G., Solenthaler, B., Teschner, M.: Versatile rigid-fluid coupling for incompressible SPH. ACM Trans. Graph. (TOG) **31**(4), 62 (2012)
2. Angelidis, A., Neyret, F.: Simulation of smoke based on vortex filament primitives. In: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 87–96. ACM, New York (2005)
3. Becker, M., Teschner, M.: Weakly compressible SPH for free surface flows. In: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 209–217 (2007)
4. Bender, J., Koschier, D.: Divergence-free smoothed particle hydrodynamics. In: Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 147–155. ACM, New York (2015)
5. Bender, J., Koschier, D., Kugelstadt, T., Weiler, M.: Turbulent micropolar SPH fluids with foam. IEEE Trans. Vis. Comput. Graph. **25**(6), 2284–2295 (2018)
6. Chu, M., Thuerey, N.: Data-driven synthesis of smoke flows with CNN-based feature descriptors. ACM Trans. Graph. (TOG) **36**(4), 69 (2017)
7. Clancy, L.J.: Aerodynamics. Halsted Press, London (1975)
8. Colagrossi, A., Graziani, G., Pulvirenti, M.: Particles for fluids: SPH versus vortex methods. Math. Mech. Complex Syst. **2**(1), 45–70 (2013)
9. Eberhardt, S., Weißmann, S., Pinkall, U., Thuerey, N.: Hierarchical vorticity skeletons. In: Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation—SCA, pp. 1–11 (2017)
10. Edwards, E., Bridson, R.: Detailed water with coarse grids: combining surface meshes and adaptive discontinuous Galerkin. ACM Trans. Graph. (TOG) **33**(4), 136 (2014)
11. Fedkiw, R., Stam, J., Jensen, H.W.: Visual simulation of smoke. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 15–22. ACM, New York (2001)
12. Fernando, D.G., Corentin, W., Jin, H.: Power particles: an incompressible fluid solver based on power diagrams. ACM Trans. Graph. **34**(4), 1–11 (2015)
13. Foster, N., Metaxas, D.: Realistic animation of liquids. Graph. Models Image Processing **58**, 471–483 (1995)
14. Gingold, R.A., Monaghan, J.J.: Smoothed particle hydrodynamics: theory and application to non-spherical stars. Mon. Not. R. Astron. Soc. **181**(3), 375–389 (1977)
15. Holm, D.D.: Fluctuation effects on 3D Lagrangian mean and Eulerian mean fluid motion. Physica D **133**(1–4), 215–269 (1999)
16. Ihmsen, M., Akinci, N., Becker, M., Teschner, M.: A parallel SPH implementation on multi-core CPUs. Comput. Graph. Forum **30**(1), 99–112 (2011)
17. Ihmsen, M., Cornelis, J., Solenthaler, B., Horvath, C., Teschner, M.: Implicit incompressible SPH. IEEE Trans. Vis. Comput. Graphics **20**(3), 426–435 (2014)
18. Ihmsen, M., Orthmann, J., Solenthaler, B., Kolb, A., Teschner, M.: SPH fluids in computer graphics. In: Eurographics 2014: STARs (2014)
19. Kaufmann, W.: Über die ausbreitung kreiszylindrischer wirbel in zähen (viskosen) flüssigkeiten. Arch. Appl. Mech. **31**(1), 1–9 (1962)
20. Kim, T., Tessendorf, J., Thuerey, N.: Closest point turbulence for liquid surfaces. ACM Trans. Graph. (TOG) **32**(2), 15 (2013)
21. Kim, T., Thürey, N., James, D., Gross, M.: Wavelet turbulence for fluid simulation. ACM Trans. Graph. (TOG) **27**(3), 50 (2008)
22. Koschier, D., Bender, J., Solenthaler, B., Teschner, M.: Smoothed particle hydrodynamics for physically-based simulation of fluids and solids. In: EUROGRAPHICS 2019 Tutorials. Eurographics Association (2019). https://doi.org/10.2312/egt.20191035
23. Lamb, H.: Hydrodynamics. Cambridge University Press, Cambridge (1993)
24. Lentine, M., Aanjaneya, M., Fedkiw, R.: Mass and momentum conservation for fluid simulation. In: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 91–100. ACM, New York (2011)
25. Liu, M., Liu, G.: Smoothed particle hydrodynamics (SPH): an overview and recent developments. Arch. Comput. Methods Eng. **17**(1), 25–76 (2010)

26. Liu, S., Wang, B., Ban, X.: Multiple-scale simulation method for liquid with trapped air under particle-based framework. In: 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 842–850. IEEE, New York (2020)
27. Lucy, L.B.: A numerical approach to the testing of the fission hypothesis. Astron. J. **82**, 1013–1024 (1977)
28. Mercier, O., Beauchemin, C., Thuerey, N., Kim, T., Nowrouzezahrai, D.: Surface turbulence for particle-based liquid simulations. ACM Trans. Graph. (TOG) **34**(6), 202 (2015)
29. Monaghan, J.J.: Smoothed particle hydrodynamics. Rep. Prog. Phys. **68**(8), 1703 (2005)
30. Müller, M., Charypar, D., Gross, M.: Particle-based fluid simulation for interactive applications. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 154–159 (2003)
31. Narain, R., Sewall, J., Carlson, M., Lin, M.C.: Fast animation of turbulence using energy transport and procedural synthesis. ACM Trans. Graph. (TOG) **27**(5), 166 (2008)
32. Oseen, C.: Über wirbelbewegung in einer reibenden flüssigkeit. Ark. F. Mat. Astron. Fys **7**, 14 (1912)
33. Park, S.I., Kim, M.J.: Vortex fluid for gaseous phenomena. In: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 261–270. ACM, New York (2005)
34. Pfaff, T., Thuerey, N., Gross, M.: Lagrangian vortex sheets for animating fluids. ACM Trans. Graph. (TOG) **31**(4), 112 (2012)
35. Rankine, W.J.M.: Manual of Applied Mechanics. Griffin, New York (1876)
36. Sato, S., Dobashi, Y., Kim, T., Nishita, T.: Example-based turbulence style transfer. ACM Trans. Graph. (TOG) **37**(4), 84 (2018)
37. Shao, X., Zhou, Z., Zhang, J., Wu, W.: Realistic and stable simulation of turbulent details behind objects in smoothed-particle hydrodynamics fluids. Comput. Anim. Virtual Worlds **26**(1), 79–94 (2015)
38. Solenthaler, B., Pajarola, R.: Predictive–corrective incompressible SPH. ACM Trans. Graph. (TOG) **28**(3), 40 (2009)
39. Stam, J.: Stable fluids. ACM Trans. Graph. **1999**, 121–128 (1999)
40. Steinhoff, J., Underhill, D.: Modification of the Euler equations for "vorticity confinement": application to the computation of interacting vortex rings. Phys. Fluids **6**, 2738–2744 (1994)
41. Stock, M.J., Dahm, W.J., Tryggvason, G.: Impact of a vortex ring on a density interface using a regularized inviscid vortex sheet method. J. Comput. Phys. **227**(21), 9021–9043 (2008)
42. Thuerey, N., Kim, T., Pfaff, T.: Turbulent fluids. In: ACM SIGGRAPH 2013 Courses, p. 6. ACM, New York (2013)
43. Thürey, N., Wojtan, C., Gross, M., Turk, G.: A multiscale approach to mesh-based surface tension flows. ACM Trans. Graph. (TOG) **29**(4), 48 (2010)
44. Vatistas, G.H.: New model for intense self-similar vortices. J. Propul. Power **14**(4), 462–469 (1998)
45. Vatistas, G.H., Kozel, V., Mih, W.: A simpler model for concentrated vortices. Exp. Fluids **11**(1), 73–76 (1991)
46. Wang, X., Ban, X., Liu, S., He, R., Xu, Y.: Small-scale surface details simulation using divergence-free sph. J. Vis. Lang. Comput. **48**, 91–100 (2018)
47. Wang, X., Ban, X., Liu, X., Zhang, Y., Wang, L.: Efficient extracting surfaces approach employing anisotropic kernels for SPH fluids. J. Vis. **19**(2), 301–317 (2016)
48. Wang, X., Liu, S., Ban, X., Xu, Y., Zhou, J., Wang, C.: Recovering turbulence details using velocity correction for SPH fluids. In: SIGGRAPH Asia 2019 Technical Briefs, SA'19, pp. 95–98 (2019)
49. Weißmann, S., Pinkall, U.: Filament-based smoke with vortex shedding and variational reconnection. ACM Trans. Graph. **29**(4), 1–12 (2010)
50. Weißmann, S., Pinkall, U., Schröder, P.: Smoke rings from smoke. ACM Trans. Graph. (TOG) **33**(4), 140 (2014)
51. Wojtan, C., Müller-Fischer, M., Brochu, T.: Liquid simulation with mesh-based surface tracking. In: ACM SIGGRAPH 2011 Courses, p. 8. ACM, New York (2011)
52. Zhang, X., Bridson, R., Greif, C.: Restoring the missing vorticity in advection–projection fluid solvers. ACM Trans. Graph. (TOG) **34**(4), 52 (2015)
53. Zhang, Y., Ban, X., Du, F., Di, W.: Fluidsnet: end-to-end learning for Lagrangian fluid simulation. Expert Syst. Appl. **2020**, 113410 (2020)
54. Zhu, B., Yang, X., Fan, Y.: Creating and preserving vortical details in SPH fluid. Comput. Graph. Forum **29**(7), 2207–2214 (2010)



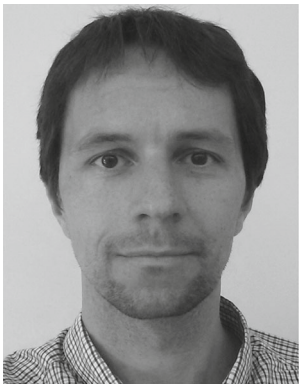

**Xiaokun Wang** received his Ph.D. degree in Computer Science and Technology from University of Science and Technology Beijing, China, in 2017. Currently, he is a lecturer at University of Science and Technology Beijing. His research interests include computer graphics, virtual reality and human–computer interaction.

**Sinuo Liu** received her undergraduate Diploma in Computer Science and Technology from University of Science and Technology Beijing, China, in 2016. Currently, she is a Ph.D. student at University of Science and Technology Beijing. Her research field is computer graphics, especially physically-based fluid simulation.

**Xiaojuan Ban** is a professor of Computer Science and Technology and leader of the Computer Animation Group at University of Science and Technology Beijing, China. She received her master in computer application and Ph.D. degree in control theory and control engineering from the University of Science and Technology Beijing. Her research interests include computer graphics, artificial intelligence, human–computer interaction, big data analysis and 3D visualization.

**Yanrui Xu** received his undergraduate Diploma in Computer Science and Technology from University of Science and Technology Beijing, China, in 2017. Currently, he is a postgraduate student at University of Science and Technology Beijing. His research field is computer graphics, especially physically-based fluid simulation.

**Jing Zhou** received his undergraduate Diploma in Computer Science and Technology from University of Science and Technology Beijing, China, in 2019. Currently, he is a postgraduate student at University of Science and Technology Beijing. His research field is computer graphics, especially physically-based fluid simulation.



**Jiří Kosinka** is an Assistant Professor in the Scientific Visualization and Computer Graphics research group at the Bernoulli Institute of the University of Groningen, the Netherlands. He received his Ph.D. degree from Charles University in Prague in 2006. His research interests include computer graphics, geometric modelling, and computer aided design.