


Article

# An Improved High-Realism Turbulence Simulation of Ocean Scenes in a Maritime Simulator

Tianhui Zhu <sup>1</sup> , Hongxiang Ren <sup>1,\*</sup>, Haijiang Li <sup>2</sup> and Xiao Yang <sup>1</sup>

<sup>1</sup> Navigation College, Dalian Maritime University, Dalian 116026, China; dmuzth@gmail.com (T.Z.); yangxiao@dmlu.edu.cn (X.Y.)

<sup>2</sup> Collaborative Innovation Center for Transport Studies, Dalian Maritime University, Dalian 116026, China; haijiang@dmlu.edu.cn

\* Correspondence: dmu\_rhx@163.com

**Abstract:** The realism of ocean visual systems is a key challenge in developing maritime simulators within ocean engineering. Achieving high realism in turbulence simulation is crucial for enhancing the effectiveness of these simulators. Traditional spectrum-based methods lack realism and fail to generate turbulent interaction effects. To address this, an improved Hybrid Smoothed Particle Hydrodynamics method is proposed for simulating ocean scenes, incorporating advanced micropolar fluid model techniques to enhance detail realism. The proposed algorithm introduces a density constraint solver that directly adjusts particle distribution and couples it with a divergence-free velocity solver, aiming to construct a physical-based fluid simulation framework that enhances detail realism in ocean scene simulations. The results demonstrate that the proposed method effectively accelerates the convergence of constraint conditions, reduces simulation time, and improves overall incompressibility. Additionally, the introduced turbulence model addresses high-frequency detail loss caused by numerical dissipation in the Smoothed Particle Hydrodynamics method, enabling more complex navigation scenarios. This study provides theoretical and technical references for achieving realistic ocean scene simulations in maritime simulators.

**Keywords:** maritime simulator; hybrid SPH; physical-based simulation; micropolar fluids; ocean scene



**Citation:** Zhu, T.; Ren, H.; Li, H.; Yang, X. An Improved High-Realism Turbulence Simulation of Ocean Scenes in a Maritime Simulator. *J. Mar. Sci. Eng.* **2024**, *12*, 1498. <https://doi.org/10.3390/jmse12091498>

Academic Editor: Salman Nazir

Received: 6 August 2024

Revised: 20 August 2024

Accepted: 28 August 2024

Published: 30 August 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

### 1.1. Motivation

The maritime simulator is an essential platform for seafarer training, theoretical engineering research, scientific practice, and other fields. Compared with the high cost of real ship training and research, the maritime simulator provides a virtual maritime environment for skill development. Maritime simulators are widely regarded as an efficient, safe, and low-cost alternative, and are extensively utilized in the field of maritime education and research. The International Maritime Organization (IMO) mandates that cadets use simulators for training under the International Convention on Standards of Training, Certification, and Watchkeeping for Seafarers (STCW Convention) and its amendments. Furthermore, it imposes additional mandatory requirements on navigation. Various performance requirements for the simulator have been established. As one of the critical components of the maritime simulator, the realism of the visual system directly determines the overall realism, making the study of its realism crucial. The ocean scene occupies more than half of the interface of the entire visual system. The highly realistic ocean scene should accurately simulate the movement of seawater and the interactions between seawater, ships, and navigation aids. Currently, most modeling methods for ocean scenes in visual systems employ spectrum-based methods. The advantage of this method is that it enables real-time simulation of large-scale seawater movements at a faster rate, with increased computational speed. However, when it is necessary to capture detailed scenarios, such as fluid–solid coupling effects such as green water and seawater impacting and breaking against the hull, the spectral

method falls short. Additionally, the scenes simulated by the spectral method lack physical authenticity. The use of computational fluid dynamics (CFD) methods to simulate seawater movement significantly enhances the physical realism of the simulation results and allows for a better simulation of complex ocean interaction scenes under various conditions. This approach represents the future development trend of maritime simulator visual systems.

## 1.2. Related Work

### 1.2.1. Physical-Based Fluid Simulation

The physical-based simulation of fluid is primarily based on CFD methods, derived from the Navier–Stokes equation, an equation governing a fundamental set of fluid mechanics. There are two principal methods for describing flow field motion: the Eulerian method and the Lagrangian method. Eulerian methods describe spatial aspects, with the finite difference method being a prominent example [1]. The Lagrangian methods focus on describing material points, with prominent examples including the finite element method [2] and the smoothed particle hydrodynamics (SPH) method [3]. Because wave motion in ocean scenes can cause significant deformation and fragmentation upon contact with interactive objects, grid-based methods often result in issues such as grid distortion [4] during simulation. Grid-less particle methods, such as the particle-based SPH method, eliminate the need for differential operations on the grid. Particles are coupled through kernel functions, thus avoiding this issue and making it highly suitable for simulating ocean scenes. The SPH method, originally proposed by Gingold [5], utilized Newton's particle equations for simulating multivariable stellar models in astrophysics. Later, Monaghan [6] first introduced it into the field of fluid simulation to address the issue of requiring boundary conditions for any moving surface, and the SPH method was applied to simulate free surface flow. Stam [7] employs the diffusion process to simulate gas phenomena, such as smoke. Müller et al. [8] introduced the SPH method into computer graphics fluid simulation first, deriving the viscous term and pressure field from NSE and designing a new kernel function to achieve surface tracking and rendering at an interactive rate. However, the performance level was modest and the overall experimental results were limited, as this was constrained by the graphics processing unit (GPU) available at that time. Furthermore, the derivation of the model in this study is based on the ideal gas equation of state, leading to the fluid system being incompletely incompressible. Following Müller's work, research into the incompressibility of SPH fluids emerged as a prominent topic in the academic community.

In the field of ocean engineering, as the most prevalent fluid type, seawater is incompressible. Therefore, studying the incompressibility of seawater enables precise modeling of the waves on its surface, thereby enhancing the realism of the overall simulation. The study of fluid incompressibility can be categorized into two types based on the methods used to solve pressure, the local pressure solver and the global pressure solver. The local pressure solver employs the equation of state to determine the pressure of the fluid system with the representative methods WCSPH and PCISPH [9,10]. The abbreviations of relevant proper nouns can be looked up in Appendix A. Becker applied the high-sonic Tait equation to minimally oscillate the density of the flow field [9]. This approach alleviated concerns regarding the physical authenticity issues caused by SPH compressibility. However, this method inherently exhibits density oscillations. Compared to incompressible fluids, physical realism is lower. Additionally, the instability of the flow field requires simulations to be conducted with smaller time steps, thereby increasing the simulation duration. Solenthaler [10] implements a predictive correction scheme by estimating the fluid system's velocity and position at the next time step to achieve an incompressible state through density corrections. Consequently, current mainstream research focuses on the global pressure solver. The global pressure solver primarily derives the pressure of the fluid system by solving the Pressure Poisson Equation (PPE), notable examples of which are IISPH, DFSPH, and PBF [11–13]. Ihmsen et al. [12] utilized the symmetric pressure formula and the SPH discretization of the continuity equation to derive a discrete-form solution

for PPE. Bender considered both the incompressibility of the volume and the condition of the divergence-free velocity field in incompressible fluids. Simultaneously solving two PPEs has become a popular and stable simulation method, allowing for larger timesteps in large-scale fluid simulations [11]. The PBF method proposed by Macklin is an approximate SPH method to some extent [13]. It enforces constant density constraints on the fluid system and directly adjusts the positions of fluid particles to enforce incompressibility. This method yields excellent results. As the direct resolution of the PPE is not required, this method enables large time steps for real-time level simulations. However, the artificial pressure term in this method is devoid of physical significance.

In recent years, several coupling methods combining particle and grid techniques have emerged, including MultiFLIP and NBFLIP [14,15]. The core concept derives from the FLIP method, which employs particles to represent fluids and grids to calculate interactions among particles. This approach reduces numerical dissipation in fluid simulations through coupling. Based on the SPH method, numerous coupling techniques involving grid-particle and particle-particle interactions have been developed, including Hybrid Eulerian-DFSPH, DFPBF, FDSPH, and SPH-MPS [16–19]. These coupling methods are primarily applied in specific field scenarios with positive outcomes; however, they are not ideal for ocean engineering applications. Additionally, the SPH method utilizes kernel functions to estimate fluid properties and facilitates the discretized particle approximation process, resulting in varying degrees of numerical dissipation [20]. Therefore, when employing the SPH method for fluid simulations, additional measures are necessary to recreate the highly realistic turbulence effects. Applying a turbulence reduction method enhances the detail in simulations.

### 1.2.2. Turbulence Restoring Method

The high-frequency details of the SPH method are often smoothed out due to its coarse discretization process, resulting in the loss of vorticity in fluid particles [21]. The reduction of turbulence can affect the realism of ocean scenes, making it essential to restore turbulence details to enhance the realism of the whole scenario. Currently, methods for restoring turbulence details are classified into four main types: up-res sampling, vorticity confinement, the Lagrangian vortex method, and vortex particle method [22–25].

Up-res sampling involves adding a high-resolution mesh or high-frequency turbulence details to the coarsely discretized upper surface of the fluid system, thus improving the realism of its turbulence details. In grid-based fluid simulation, Edwards introduced an adaptive volume grid method to further subdivide the grid and augment fluid details [26]. With advancements in machine learning, high-resolution turbulence can be generated by training convolutional neural networks, although this training is time-consuming and primarily enhances surface turbulence effects [27,28].

The vorticity confinement method works by calculating the optimal vortex in the fluid system and applying vorticity confinement forces to increase the velocity of fluid particles, thereby intensifying the vortex effect. Jang et al. achieved improved outcomes by stratifying vorticity limitations and applying varied levels of restriction forces depending on system conditions [29]. Macklin and Müller developed a fluid simulation method based on PBD and employed the vorticity confinement method to counteract the reduced rotational velocity from numerical damping. The vorticity confinement method is straightforward to implement and can maintain and amplify existing vortices. However, it does not create additional fluid details, and the amplification level is regulated by the adjustment coefficient. When the adjustment coefficient is excessively high, this method introduces additional energy to the fluid system; thus, the system energy is non-conservative and hard to control.

The Lagrangian vortex method restores turbulence details using various elements such as filaments, particles, patches, curves, and surfaces [30–34]. This method fundamentally solves the vorticity form of the Navier–Stokes equations, thereby ensuring that the fluid system’s overall divergence remains 0 without suffering from numerical dissipation. A notable disadvantage of this method is its difficulty in handling boundary issues, particularly with

non-rigid bodies and free surfaces. To address this, some studies have simplified the boundary problem by using Eulerian mesh; however, this approach still incurs high computational times as it relies on the Biot–Savart integral to compute the velocity field [35,36].

The vortex particle method, a relatively new approach, exhibits strong adaptability for particle-based fluid simulations. Zhang et al. introduced an advection scheme that combines vorticity and velocity, aiming to minimize numerical dissipation [25]. While this method excels in simulating smoke, it is less effective for liquid simulations. Wang et al. utilized the Rankine vortex model to develop a turbulence simulation method using vortex particles [37]. They modeled viscous vortices by simplifying control equations and employing empirical models instead of analytical solutions. For simulating inviscid or nearly inviscid fluids, the micropolar fluid model offers superior performance, preserving system momentum and energy while capturing detailed vortex currents [38]. Bender et al. utilized this approach to simulate foam rotation on water surfaces [39]. This study introduces and adapts this technique to the hybrid divergence-free SPH (HYSPH) method outlined herein, enhancing turbulence detail and realism in ocean scene simulations within the maritime simulator.

The main contributions of this article are outlined as follows:

(1) We propose a novel fluid simulation method, HYSPH, employing a density constraint solver to replace the constant density solver used in the traditional DFSPH method. This modification significantly enhances the convergence speed of the model and facilitates simulations with larger time steps.

(2) Utilizing the new HYSPH method, we incorporate the micropolar fluid model to counteract the loss of turbulence details due to the inherent numerical damping caused by the method itself, thereby enhancing the turbulent surface details.

(3) We have developed a physically based ocean scene simulation framework, tailored for maritime simulators. This framework offers enhanced physical authenticity and interactivity compared to spectrum-based methods.

The remainder of this paper is structured as follows. Section 2 discusses the SPH discretization and the physical solver model. Section 3 details the turbulence simulation method. Section 4 presents and analyzes our experimental results. Finally, Section 5 summarizes this paper and outlines future research directions.

## 2. Simulation Model

### 2.1. Governing Equation and SPH Discretization

In the practice of maritime simulators, the effects of salinity and temperature are often ignored when dealing with the motion of oceans and waves. Waves are considered uniform, isothermal, and incompressible fluids. According to physical oceanography, the key physical quantities describing fluid motion include the velocity field, the density field, the pressure field, and others. Ocean waves are a type of fluid and when described through the mass conservation formula, the motion control equations of ocean waves conform to the requirements of continuum mechanics theory. The specific form is as follows.

$$\frac{D\rho_i}{Dt} + \rho_i \nabla \mathbf{v}_i = 0 \quad (1)$$

where  $\rho_i$  represents the density and  $\mathbf{v}_i$  represents the velocity vector of the fluid particle. The explanations of key variables can be found in Appendix B. The movement of seawater is described using principles of CFD, adhering to the three fundamental conservation laws of physics: conservation of energy, conservation of momentum, and conservation of angular momentum. The momentum equation quantifies the rate of momentum change for fluid particles.

$$\rho_i \frac{D^2 \mathbf{x}_i}{Dt^2} = \nabla \sigma_{ij} + \mathbf{f}_{\text{ext}} \quad (2)$$

The right side of the momentum equation represents the sum of all internal and external forces acting on the fluid, where  $\sigma_{ij}$  is the stress tensor of the fluid itself and  $\mathbf{f}_{\text{ext}}$

represents the sum of external body forces. The unique properties of the fluid are captured through the construction of a constitutive relationship for  $\sigma_{ij}$  and the movement of the fluid is ultimately governed by the forces suffered per unit volume. The constitutive equation for incompressible fluids is typically expressed as follows.

$$\sigma_{ij} = -p\delta_{ij} + \mu(\nabla\mathbf{v} + \nabla\mathbf{v}^T) \tag{3}$$

where  $p$  represents the pressure and  $\mu$  represents the dynamic viscosity. By plugging Equation (3) into Equation (2), the NSE for incompressible fluids can be derived as follows.

$$\rho_i \frac{D\mathbf{v}_i}{Dt} = -\nabla p_i + \mu \nabla^2 \mathbf{v}_i + \mathbf{f}_{\text{ext}} \tag{4}$$

SPH is a discretization method for continuous media based on the Lagrangian perspective. The continuous fluid field is discretized into interacting fluid particles, obtained through integral approximation. This method is widely used in particle system simulations. Physical quantities relevant to controlling fluid motion are carried by these particles. For each fluid particle  $i$ , the field value  $A(x_i)$  at  $x_i$  can be obtained by summing the values of all adjacent particles  $j$  and interpolating the results.

$$A(x_i) = \sum_j \frac{m_j}{\rho_j} A_j W(x_i - x_j, r) \tag{5}$$

where  $m_j$ ,  $\rho_j$ , and  $W(x_i - x_j, r)$  represent the mass, density, and smoothing kernel function of neighboring fluid particles, respectively, and  $r$  is the smoothing radius of the kernel function. Similar to fluid particle  $i$ , neighboring particles also carry physical quantities such as velocity, position, and pressure. To enhance the intuitiveness of the equations in this article, a simplified notation  $W_{ij}(x_i - x_j, r)$  will be employed to reduce complexity. It is not necessary to continuously track the density value of fluid particles at  $x_i$ . This value can be interpolated from the weighted mass field of neighboring particles.

$$\rho_i = \sum_j m_j W_{ij} \tag{6}$$

Each term in Equation (4) can be independently split and solved. To address the pressure gradient term  $\nabla p_i$ , the pressure  $p_i$  must first be computed. Pressure computation is dependent on density. It is typically derived by solving the PPE, yielding more accurate results. However, this approach significantly extends simulation times. Conversely, using the ideal gas equation leads to density oscillations that result in fluid compression. The Tait equation is employed to compute the pressure [11].

$$p_i = B \left( \left( \frac{\rho_i}{\rho_0} \right)^\gamma - 1 \right) \tag{7}$$

where  $B = \kappa_i \rho_0 / \gamma$  is known as the pressure constant,  $\kappa_i$  and  $\gamma$  are stiffness parameters and the rest density. Typically, based on empirical evidence,  $\gamma$  is set to 1 and  $\kappa_i$  will be derived subsequently based on the solver's force analysis.

### 2.2. Hybrid Smoothed Particle Hydrodynamics Solver

DFSPH is one of the commonly used SPH methods. Its core concept involves solving pressure using two specific solvers: the divergence-free velocity solver and the constant density solver. In a uniform incompressible fluid, the density change over time is 0. Based on Equation (1), the following relationship can be derived.

$$\frac{D\rho_i}{Dt} = \rho_i \nabla \cdot \mathbf{v}_i = 0 \rightarrow \nabla \cdot \mathbf{v}_i = 0 \tag{8}$$

From the equation, it can be derived that the material derivative of fluid density is 0.  $\nabla \cdot \mathbf{v}_i$  represents the divergence of the fluid particle velocity field. For incompressible

fluids, the divergence of the velocity field consistently remains 0. Therefore, imposing a divergence-free condition on the fluid system via a pressure solution is essential for achieving realistic fluid simulations. The pressure computed for a fluid particle  $i$  with a divergent velocity field, using this Equation, is approximated using SPH as follows.

$$\mathbf{F}_i^p = -\frac{m_i}{\rho_i} \nabla p_i = -\kappa_i^v \frac{m_i}{\rho_i} \sum_j m_j \nabla W_{ij} \tag{9}$$

To ensure the conservation of overall momentum in the fluid system, the net force from the pressure on fluid particle  $i$  and the total pressure exerted by particle  $i$  on neighboring particles sum to 0. The pressure exerted by fluid particle  $i$  on neighboring particles is expressed as follows.

$$\mathbf{F}_{j \leftarrow i}^p = -\frac{m_i}{\rho_i} \frac{\partial p_i}{\partial \mathbf{X}_j} = \frac{\kappa_i^v m_i m_j}{\rho_i} \nabla W_{ij} \tag{10}$$

The objective of the velocity divergence-free solver is to fulfill the conditions set in Equation (8) by modifying the velocities in response to pressure exerted on the fluid particles.

$$\frac{D\rho_i}{Dt} = \sum_j m_j (\mathbf{v}_i - \mathbf{v}_j) \nabla W_{ij} \tag{11}$$

According to Newton’s second law, where  $\Delta \mathbf{v}_i = \mathbf{F}_i^p \Delta t / m_i$ ,  $\Delta \mathbf{v}_j = \mathbf{F}_{j \leftarrow i}^p \Delta t / m_j$ , by substituting Equations (9) and (10) into Equation (11), the value of the stiffness parameter  $\kappa_i^v$  can be expressed as follows.

$$\kappa_i^v = -\frac{\rho_i}{\Delta t} \frac{D\rho_i}{Dt} \frac{1}{|\sum_j m_j \nabla W_{ij}|^2 + \sum_j |m_j \nabla W_{ij}|^2} \tag{12}$$

Next, we apply the Formula  $\alpha_i = -1 / (|\sum_j m_j \nabla W_{ij}|^2 + \sum_j |m_j \nabla W_{ij}|^2)$ . It can be obtained that  $\alpha_i$  is a coefficient only related to the position. When fluid particle  $i$  is at the boundary of the fluid system, the deficiency of neighboring particles can lead to unstable computations; therefore, appropriate threshold conditions must be established. Finally, the force acting on fluid particle  $i$  is derived as follows.

$$\mathbf{F}_{total}^p = \mathbf{F}_i^p + \sum_j \mathbf{F}_{j \leftarrow i}^p = -m_i \sum_j m_j \left( \frac{\kappa_i^v}{\rho_i} + \frac{\kappa_j^v}{\rho_j} \right) \nabla W_{ij} \tag{13}$$

The velocity change caused by  $\mathbf{F}_{total}^p$  is:

$$\Delta \mathbf{v}_i = \frac{\mathbf{F}_{total}^p}{m_i} \Delta t \tag{14}$$

To further ensure the incompressibility of the fluid system, the traditional DFSPH method employs a constant density solver as the secondary solver. By solving the PPE using  $\rho_i - \rho_0$  as the source term and computing the positions of fluid particles, the fluid system can reach the incompressible state. Therefore, the difference between the current density and the rest density is ideally maintained at 0. In this paper, we adopt the concept of constraints from Müller et al. in position-based dynamics and introduce a novel nonlinear density constraint solver to replace the constant density solver in the traditional DFSPH method [40]. The core concept involves the following constraints.

$$C_i(\mathbf{x}) = \frac{\rho_i}{\rho_0} - 1 \tag{15}$$

The constraint is  $C_i(\mathbf{x}) \leq 0$ , where  $\mathbf{x} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$ ,  $n$  is the number of neighbor particles of the fluid particle  $i$ . By introducing a particle position correction  $\Delta \mathbf{x}_i$  and applying the first-order Taylor expansion to the constraints, we derive:

$$C_i(\mathbf{x} + \Delta\mathbf{x}_i) = C_i(\mathbf{x}) + \nabla C_i(\mathbf{x}) \cdot \Delta\mathbf{x}_i + O(|\Delta\mathbf{x}_i|^2) \tag{16}$$

To ensure the conservation of both momentum and angular momentum, the density constraint solver restricts the direction of  $\Delta\mathbf{x}_i$  to align with the constraint gradient direction, denoted as  $\nabla C_i(\mathbf{x})$ , i.e.,  $\Delta\mathbf{x}_i = \lambda_i \nabla C_i(\mathbf{x})$ . Additionally,  $\lambda_i$ , the Lagrangian multiplier, maximizes the growth rate of  $\Delta\mathbf{x}_i$  and accelerates the convergence of the algorithm. Under the specified constraint conditions, the gradient of the constraint for fluid particle  $i$  is  $\nabla C_i(\mathbf{x}) = 1/\rho_0 \nabla \rho_i$ . Following SPH expansion, its discrete form is derived as follows.

$$\nabla C_i(\mathbf{x}) = \frac{1}{\rho_0} \sum_j m_j \nabla W_{ij} \tag{17}$$

Similar to the concept of symmetric forces in the divergence-free solver, in the density constraint solver, the stability of constraint conditions can be ensured by deriving the gradient at the neighboring particle as follows.

$$\nabla C_{j \leftarrow i}(\mathbf{x}) = -\frac{1}{\rho_0} m_j \nabla W_{ij} \tag{18}$$

Plugging Equations (17) and (18) into constraint Equation (16), the Lagrangian multiplier  $\lambda_i$  can be obtained as:

$$\lambda_i = -\frac{C_i(\mathbf{x})}{|\nabla C_i(\mathbf{x})|^2 + \sum_j |\nabla C_{i \leftarrow j}(\mathbf{x})|^2} = \rho_0^2 \alpha_i C_i(\mathbf{x}) \tag{19}$$

In the Equation, the parameter  $\alpha_i$  corresponds with the  $\alpha_i$  used in the velocity divergence-free solver, enabling the density constraint solver proposed in this paper to more effectively match the velocity divergence-free solver of the original method. Upon obtaining the Lagrangian multiplier, the final displacement offset is computed according to Equation (20):

$$\Delta\mathbf{x}_i = \lambda_i \nabla C_i(\mathbf{x}) + \sum_j \lambda_j \nabla C_j(\mathbf{x}) = \frac{1}{\rho_0} \sum_j m_j (\lambda_i + \lambda_j) \nabla W_{ij} \tag{20}$$

So far, the overall algorithm of the HYPH model is shown in the Algorithm 1.

---

**Algorithm 1.** HYPH Model

---

```

for all particle  $i$  do
  search neighborhood  $N_i$  of particle  $i$ 
  calculate densities  $\rho_i$  and factors  $\alpha_i$ 
  calculate non-pressure forces  $F_i^{(non-pre)}$ 
   $\mathbf{v}_i^* = \mathbf{v}_i + F_i^{(non-pre)} \Delta t / m_i$ 
  while ( $\rho_{error} > \eta$ )  $\vee$  ( $iter < solver\_iter$ )
    for all particle  $i$  do
      calculate densities  $\rho_i$  and constraints  $C_i$ 
    if  $C_i > 0$  do
      calculate  $\lambda_i = \rho_0^2 \alpha_i C_i(\mathbf{x})$ 
    for all particle  $i$  do
      calculate  $\Delta\mathbf{x}_i = \sum_j m_j (\lambda_i + \lambda_j) \nabla W_{ij} / \rho_0$ 
      update position  $\mathbf{x}_i = \mathbf{x}_i + \Delta\mathbf{x}_i$ 
    update  $\rho_{error}$  and  $iter$ 
    calculate  $D\rho/Dt$ 
    while ( $\rho_{error} > \eta$ )  $\vee$  ( $iter < solver\_iter$ )
      for all particle  $i$  do
        calculate  $\kappa_i^v = -\alpha_i \rho_i D\rho_i / \Delta t Dt$ 
        calculate  $\Delta\mathbf{v}_i = -\Delta t \sum_j m_j (\kappa_i^v / \rho_i + \kappa_j^v / \rho_j) \nabla W_{ij}$ 
        update  $D\rho/Dt$ 
      update  $\rho_{error}$  and  $iter$ 

```

---

### 3. High-Realism Turbulence Simulation with Micropolar Fluid Model

#### 3.1. Conservation of Linear and Angular Momentum and Incompressibility

Fluid in three-dimensional space can be described by the density field  $\rho : \mathbb{R}^3 \rightarrow \mathbb{R}$ . The velocity field  $\mathbf{v} : \mathbb{R}^3 \times [0, \infty) \rightarrow \mathbb{R}^3$  indicates the linear velocity vector of a point  $\mathbf{x}$  at time  $t$ . In the micropolar fluid model, the fluid is also influenced by the micro-rotation field.  $\boldsymbol{\omega} : \mathbb{R}^3 \times [0, \infty) \rightarrow \mathbb{R}^3$  represents the angular velocity vector of a point  $\mathbf{x}$  at time  $t$ . The conservation of linear and angular momentum can be defined as follows.

$$\rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \mathbf{T} + \mathbf{f} \tag{21}$$

$$\rho \Theta \frac{D\boldsymbol{\omega}}{Dt} = \nabla \cdot \mathbf{C} + \mathbf{T}_\times + \boldsymbol{\tau} \tag{22}$$

where  $[\mathbf{T}_\times]_i = \sum_j \sum_k \varepsilon_{ijk} \mathbf{T}_{jk}$ ,  $\mathbf{T}$  denotes the stress tensor while  $\mathbf{f}$  and  $\varepsilon_{ijk}$  represent a specific external force and a Levi-Civita symbol, respectively. The isotropic microinertia coefficient is represented by  $\Theta$ , it reflects the inertial resistance of fluid particles to rotational acceleration. Based on multiple experimental deductions and comparisons, Bender et al. decided to use empirical values and set the isotropic microinertia coefficient to 2 [39].  $\boldsymbol{\tau}$  represents a specific external torque and the coupled stress tensor  $\mathbf{C}$  is used to simulate the stress generated by angular motion.

#### 3.2. Constitutive Equations

The constitutive equation, also known as the rheological equation, describes the interrelationships between specific kinematic quantities of a continuous medium. To develop a micropolar fluid model suitable for simulating inviscid and micro-viscous fluids, it is essential to establish constitutive relationships for the stress tensor  $\mathbf{T}$  and the coupling tensor  $\mathbf{C}$ .

$$\mathbf{T} = -p\mathbf{I} + \mu \nabla \mathbf{v}^T - \mu_t \nabla \mathbf{v} + (\mu + \mu_t) \boldsymbol{\omega}^\times \tag{23}$$

$$\mathbf{C} = c \nabla \boldsymbol{\omega}^T \tag{24}$$

where  $[\boldsymbol{\omega}^\times]_{jk} = \sum_i \sum_k \varepsilon_{jik} \omega_i$ ,  $p$  represents pressure,  $\mathbf{I}$  represents the identity matrix, and the pressure term establishes the foundation of the fluid model, offsetting compression. In the traditional Navier–Stokes equation, the viscosity term is represented by a symmetric second-order tensor,  $\mu (\nabla \mathbf{v} + \nabla \mathbf{v}^T)$ . In contrast, in the micropolar fluid model, two independent material parameters  $\mu$  and  $\mu_t$  are used to represent the separation tensor.  $\mu$  and  $\mu_t$  denote dynamic viscosity and the transfer coefficient, respectively. To ensure consistency with the second law of thermodynamics, the material parameters must satisfy  $\mu \geq 0$  and  $\mu_t \geq -\mu$ . This ensures that the terms in the model will only dissipate energy and will not generate new energy. The last term,  $(\mu + \mu_t) \boldsymbol{\omega}^\times$ , is responsible for transferring the rotational motion of fluid particles to linear motion, and vice versa. Finally, we model the coupling tensor as a diffusion term that depends on the gradient of micro-rotation and dynamic rotational viscosity parameters.

#### 3.3. Equation of Motion and Discretization

By incorporating the constitutive equation into the conservation equations of linear and angular momentum and applying the incompressibility condition, the motion equation of the micropolar fluid model can be derived.

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho} \nabla p + \nu \Delta \mathbf{v} + (v + v_t) \nabla \times \boldsymbol{\omega} + \frac{\mathbf{f}}{\rho} \tag{25}$$

$$\Theta \frac{D\boldsymbol{\omega}}{Dt} = \zeta \Delta \boldsymbol{\omega} + (v + v_t) (\nabla \times \mathbf{v} - 2\boldsymbol{\omega}) + \frac{\boldsymbol{\tau}}{\rho} \tag{26}$$

$\nu = \mu/\rho$ ,  $v_t = \mu_t/\rho$ , and  $\zeta = c/\rho$  represent kinematic viscosity, the kinematic transfer coefficient, and kinematic rotational viscosity, respectively. When  $v_t = -v$ ,  $\zeta = 0$  and  $\boldsymbol{\tau} = 0$ ,



the motion equation simplifies to the classical Navier–Stokes equation, demonstrating that the micropolar fluid model is an extension and improvement of the Navier–Stokes model. It is important to note that the conversion between linear motion and rotational motion is controlled by the kinematic transfer coefficient, which is non-dissipative because it does not appear in the diffusion term. For the simulation of waves in maritime scenarios, due to the inviscid nature of waves, the kinematic viscosity and rotational viscosity are set to 0, simplifying the motion equation as follows.

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho}\nabla p + v_t\nabla \times \boldsymbol{\omega} + \frac{\mathbf{f}}{\rho} \tag{27}$$

$$\Theta\frac{D\boldsymbol{\omega}}{Dt} = v_t(\nabla \times \mathbf{v} - 2\boldsymbol{\omega}) + \frac{\boldsymbol{\tau}}{\rho} \tag{28}$$

The linear acceleration of fluid particles is influenced by the term  $v_t\nabla \times \boldsymbol{\omega}$ , and the angular acceleration is also affected by the same term  $v_t(\nabla \times \mathbf{v} - 2\boldsymbol{\omega})$ . This demonstrates that the update of the linear velocity of fluid particles is influenced by their angular velocity.

After deriving the motion equation, the SPH method is used to solve the particle control equation. The calculation of density and pressure terms employs the position-based fluid pressure solver introduced earlier. Here, we focus on calculating the curl operator for the linear velocity and angular velocity fields. There are two methods to obtain the curl operator: the symmetric curl formula and the differential curl formula. Using the symmetric curl formula may cause visual artifacts on the liquid’s free surface due to local particle defects. To address this issue, we choose the differential curl formula for the solution.

$$(\nabla \times \mathbf{A})_i^{\text{diff}} = \frac{1}{\rho_i} \sum_j (\mathbf{A}_i - \mathbf{A}_j) \times \nabla_i W_{ij} \tag{29}$$

At this point, the algorithm for the micropolar fluid model is summarized in Algorithm 2. It should be noticed that each step of the algorithm needs to be calculated in sequence according to the process to obtain the final result.

---

**Algorithm 2.** Micropolar Fluid Model

---

```

for all particle i do
  find neighbors
for all particle i do
  compute density  $\rho_i$ 
for all particle i do
  compute non-pressure forces  $\mathbf{F}_i^{\text{nonpre}}$ 
  compute transfer forces  $\mathbf{F}_i^{\text{trans}}$ 
  compute transfer torque  $\boldsymbol{\tau}_i^{\text{trans}}$ 
  compute time step size  $\Delta t$  according to CFL
for all particle i do
   $\mathbf{v}_i^* = \mathbf{v}_i + \Delta t(\mathbf{F}_i^{\text{nonpre}} + \mathbf{F}_i^{\text{trans}} + \mathbf{F}_i^{\text{ext}})/m_i$ 
for all particle i do
  enforce incompressibility using pressure solver
  update  $\mathbf{v}_i^*$ 
for all particle i do
   $\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i^*$ 
   $x_i(t + \Delta t) = x_i^* + \Delta t\mathbf{v}_i(t + \Delta t)$ 
   $\boldsymbol{\omega}_i(t + \Delta t) = \boldsymbol{\omega}_i(t) + \Delta t(\boldsymbol{\tau}_i^{\text{trans}} + \boldsymbol{\tau}_i^{\text{ext}})/m_i\Theta$ 

```

---

**4. Results and Discussion**

We compare the novel proposed HYPSPH method with the DFSPH method to verify the effectiveness and stability of the model. Subsequently, through various scenario tests, we benchmark it against the popular turbulence method and the classic pure SPH method

to assess the performance of the model. The experimental results from the same scenarios under different methods are obtained using a unified simulation framework implemented in C++. The simulation platform configuration for the experiment includes an Intel(R) Core (TM) i5-13400F CPU (2.5 GHz), 16 GB RAM, and an NVIDIA GeForce RTX 3060 GPU. Particle rendering is achieved using a custom-built OpenGL-based renderer, while realistic scene rendering utilizes Blender 3.5, the integrated development environment employed is Microsoft Visual Studio 2019, the utilized numerical computing library is eigen, and we adopted OpenGL3.6 as the Graphics API.

4.1. Performance of HYPSPH Model

In this section, we validate our proposed HYPSPH method by using the classic dam break scenario in inland navigation practices and the CFD field, comparing and analyzing it alongside the DFSPH method under identical experimental conditions. We initially compared the total solution time costs of the two methods across three different scales of particle numbers. As depicted in Figure 1, the particle numbers are 12K, 59K, and 205K, corresponding, respectively, to small, medium, and large scales. Different particle numbers correspond to varying complexities in neighborhood particles; therefore, conducting experiments across models with varied particle numbers is essential to ascertain the simulation’s stability and effectiveness.

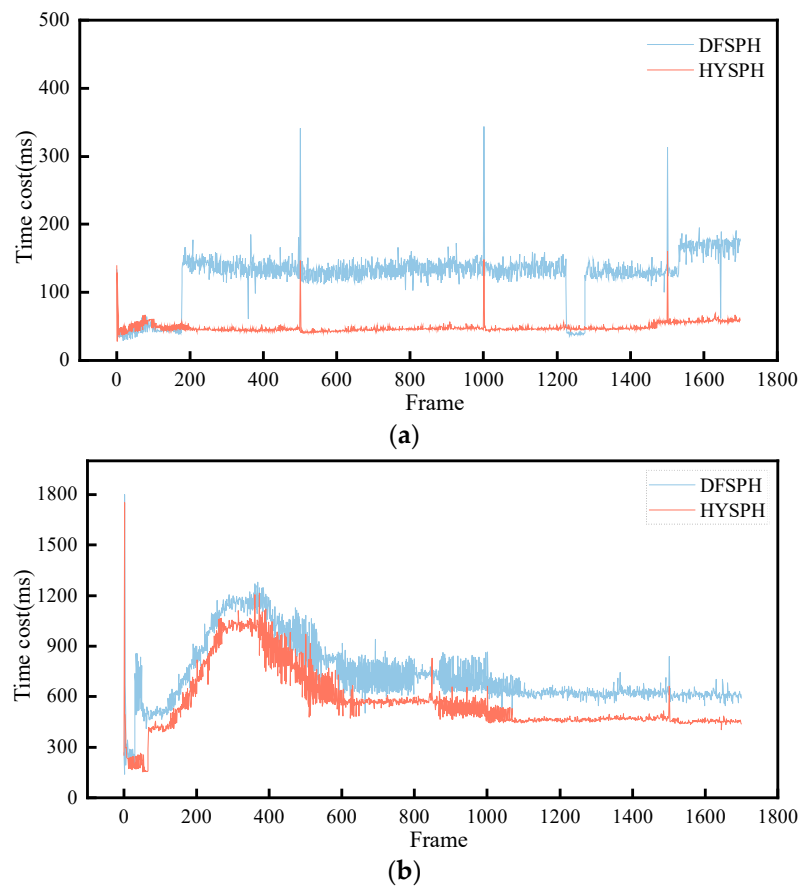
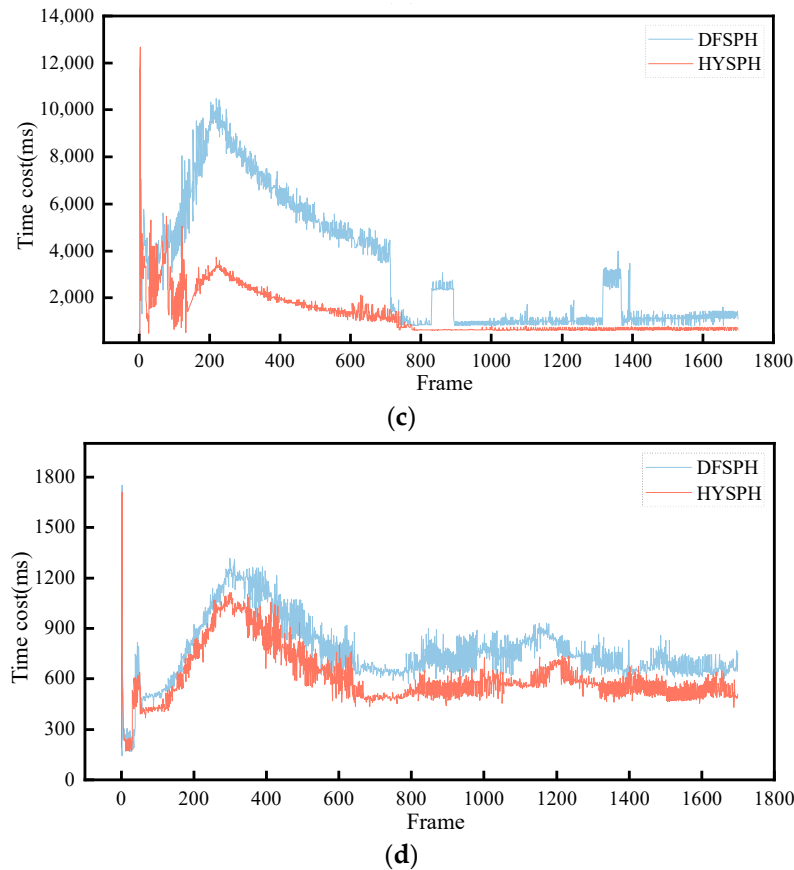


Figure 1. Cont.



**Figure 1.** Comparison of Single Frame Solution Time for Fluid Model Across Various Particle Numbers. (a) 12 K particles with dam break; (b) 59 K particles with dam break; (c) 205 K particles with dam break; (d) 59 K particles with dam break lighthouse coupling.

As indicated in the legend, the data in light blue represent the solution costs for different frames using the DFSPH method, while the data in orange correspond to the HYPSPH method. We extracted the data from the first 1700 frames for visual representation. As the fluid’s motion becomes static and stable after 1700 frames, referencing further solution costs becomes less relevant. Overall, with an increase in particle numbers, the corresponding time cost per time step also increases. Simulations conducted under varying particle numbers are illustrated in Figure 1b,c of our proposed HYPSPH method. The solution time for the first frame is significantly higher than that for subsequent frames. This occurs because the divergence-free solver among the two solvers requires extensive initialization, necessitating additional calculations to enhance fluid system convergence. In the experiment shown in Figure 1a, due to the small particle numbers, convergence calculations require less time, resulting in a negligible improvement in the solution time for the first frame. Meanwhile, fluid particles encounter the problem of neighboring particle deficiency and need to perform a neighborhood search again, resulting in increased time in 500, 1000, and 1500 frames compared to other frames. From the Figure, it is evident that the single-frame time cost of the proposed HYPSPH method is lower than that of the DFSPH method, and this cost advantage becomes more pronounced as the number of particles increases. For instance, in the 59 K particle scenario, the single-frame time cost for the DFSPH method is approximately 707 ms, compared to about 559 ms for our proposed HYPSPH method.

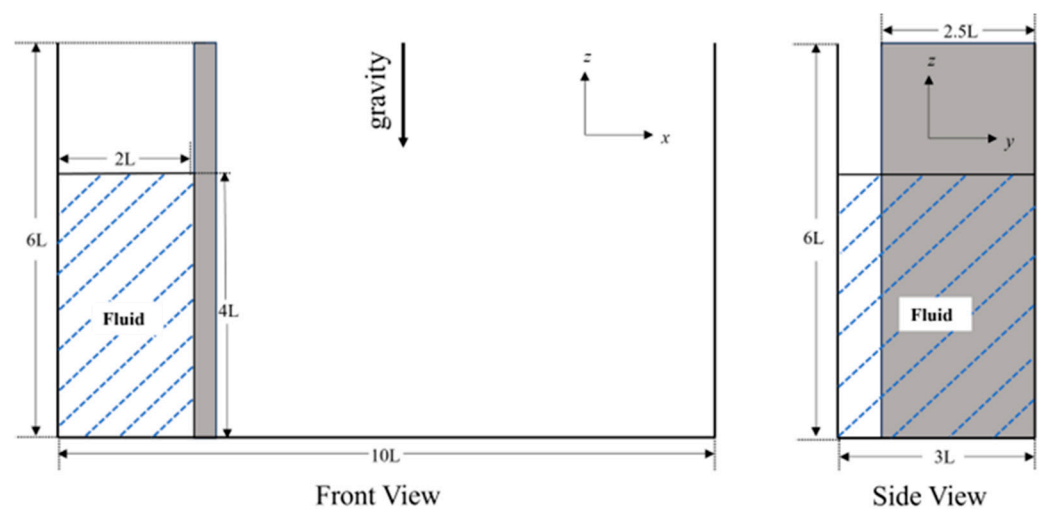
To validate the proposed method, experiments were carried out in different scenarios with the same number of particles, specifically 59 K particles. Two lighthouses were positioned on the right side of the fluid blocks of the standard dam break scenario to

validate the method in a fluid-rigid body coupling setting. The results, as depicted in Figure 1d, align with the findings of the simple dam break scenario.

For stability analysis of the fluid model, the time step is a crucial consideration in SPH-based methods. A larger time step results in a more stable and faster simulation. In the simulation experiments, the Courant–Friedrichs–Lewy (CFL) condition, commonly utilized in fluid dynamics, was used to determine the time step. The CFL condition adaptively adjusts the time step size to optimize benefits. As depicted in Figure 1c, the single-frame time cost of the DFSPH method shows significant fluctuations between the 800th and 1000th frames and between the 1200th and 1400th frames, indicating a reduction in the simulation time step. The proposed HYSPPH method demonstrates relative stability.

#### 4.2. Analysis of Turbulence Model Effects

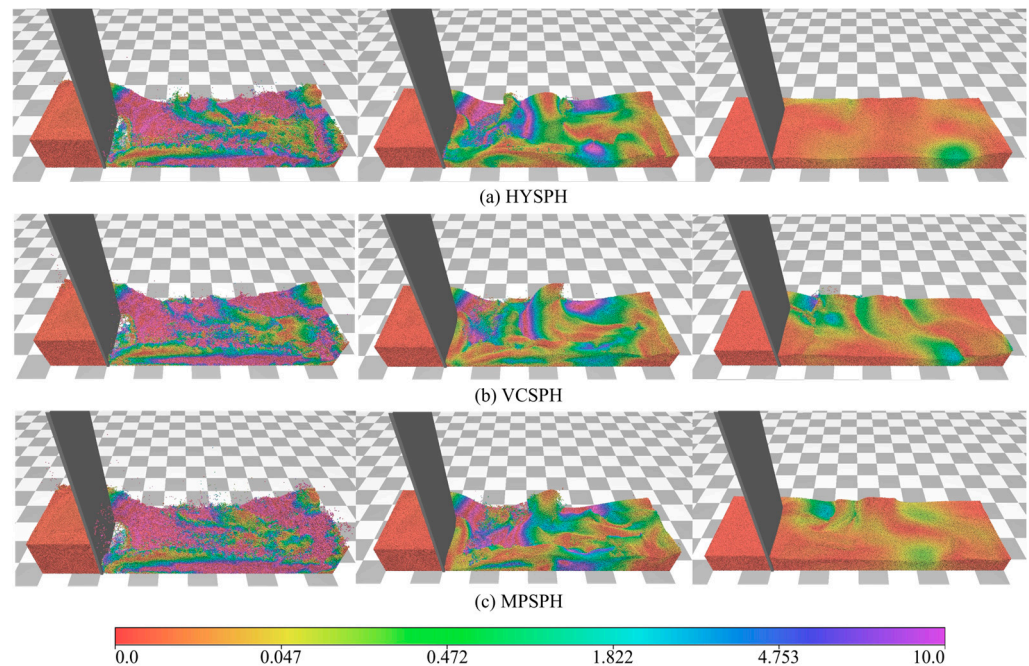
In this section, we design a baffle dam break scenario with specific constraints to capture and analyze turbulence quality. As depicted in Figure 2, the fluid block is positioned on the left side of the bounding box, which measures  $10 L \times 3 L \times 6 L$ . The fluid block itself has a volume of  $2 L \times 3 L \times 4 L$ , containing 862 K particles. A baffle, measuring  $2.5 L$  in width and  $6 L$  in height, is located on the right side of the fluid block. A gap is present at the back of the scene. The fluid model moves within the gap solely under the influence of gravity, with no other external forces affecting its motion. Each surface of the bounding box and the baffle is sampled with static boundary particles using a three-dimensional Poisson disk sampling algorithm. These particles have the same density as the fluid’s rest density, and their velocity is consistently 0.



**Figure 2.** Plan View of the Baffle Dam Break Scenario Diagram.

Fluid particles are rendered in various colors to represent their velocity field. Color variations clearly distinguish turbulence dispersion in both the surface and deeper layers of the fluid. The relationship between fluid particle velocity and color is detailed in the color bar beneath the image. The effectiveness of the turbulence model is analyzed by comparing it to the Vorticity Confinement method proposed by Macklin et al. [13] and the traditional pure SPH method. For ease of reference, the Vorticity Confinement method is henceforth referred to as VCSPH, and the micropolar fluid model method as MPSPH. Figure 3a displays the experimental results using solely the proposed pure HYSPPH method, without incorporating additional turbulence models. It illustrates that while the pure SPH method generates some turbulence during fluid particle release, the vortices rapidly weaken upon passing through the baffle and eventually disappear in the later stages of fluid movement. Simultaneously, the surface particles of the fluid system appear viscous with inactive movement. Figure 3b employs VCSPH, while Figure 3c utilizes MPSPH. Compared to the pure SPH method, both VCSPH and MPSPH generate more pronounced surface

turbulence; however, surface particle movement in MPSPH is more dynamic, and internal particles consistently exhibit rotational velocities. This occurs because the MPSPH method refines particle velocities based on the vorticity field, allowing for the natural preservation of vortices. As the fluid system stabilizes, the MPSPH method eliminates micro-rotations on the surface, leading to overall system stability; however, local surface particles in VCSPH retain high kinetic energy. Comparative analysis demonstrates that the turbulence model methods generate more kinetic energy than the traditional pure SPH method. The MPSPH method facilitates the expression of vorticity in individual particles through micro-rotation of the velocity field, thereby enhancing the restoration of high-frequency turbulence with vivid detail and realism.

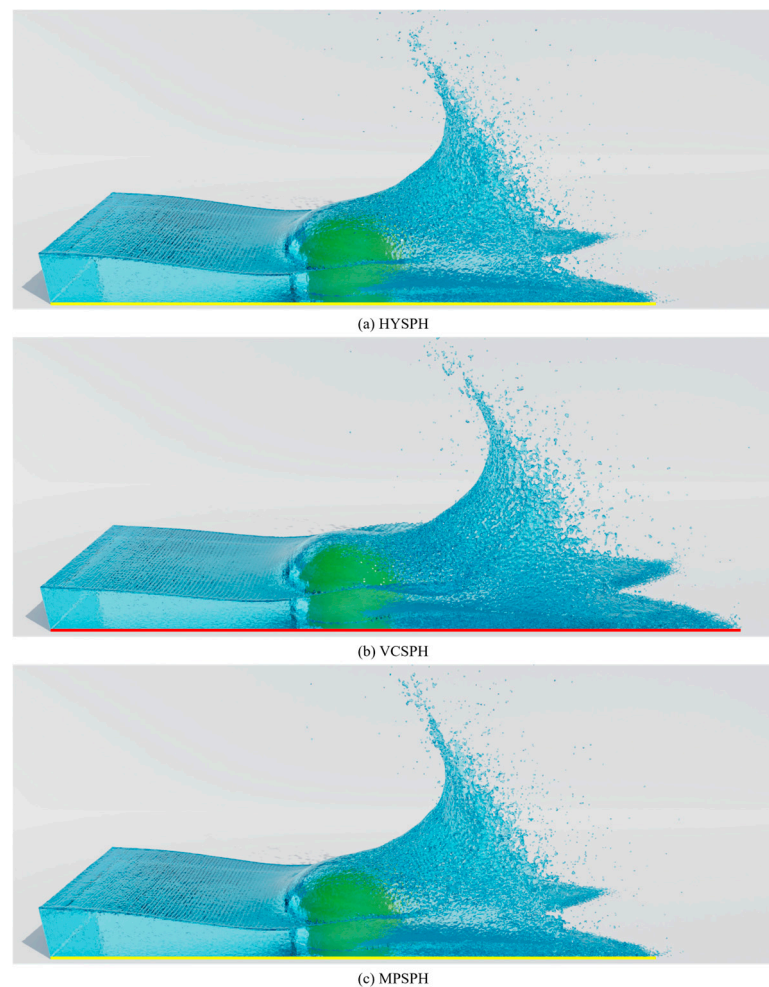


**Figure 3.** Comparison of Experimental Results on Turbulence Details in the Baffle Dam Break Scenario. The different color in the images are the velocity distributions of fluid particles.

As rotational velocity introduces kinetic energy, we designed an experimental scenario to verify the conservation of the system’s overall energy and to study the impact of various adjustment coefficients on experimental outcomes. A fixed-volume fluid block, containing 617 K particles, is located on the left side of the bounding box. A hemispherical obstacle, like the baffle collapse scenario, is positioned on the right side of the scene. No external forces affect the entire fluid system. In this scenario, the entire energy of the fluid system derives solely from gravitational potential energy, which is converted into kinetic energy to drive particle motion. As illustrated in the Figure, the experimental results for the same frame from three different simulation methods are displayed. Figure 4a shows the results from the HYPSPH method, while Figure 4b,c displays outcomes from the VCSPH and MPSPH methods, respectively. The adjustment coefficients for both methods are set at 0.5.

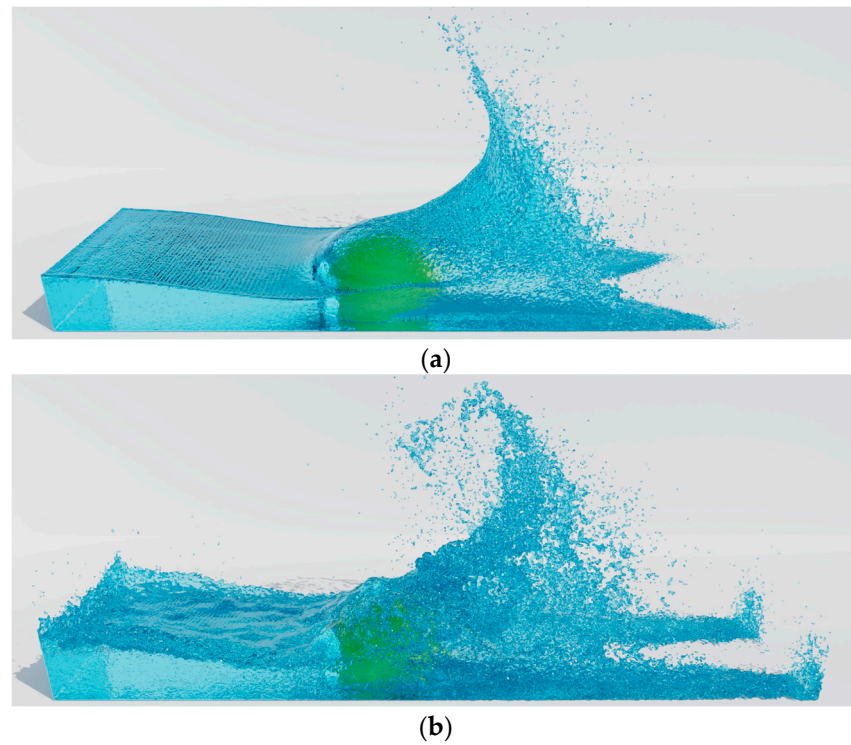
Observation of the image reveals that the front edge of the fluid system in Figure 4b has reached the boundary of the bounding box. As shown in Figure 4, the length of the red line represents the distance of fluid system motion under the VCSPH method, while the length of the yellow line represents HYPSPH and MPSPH. Compared to the HYPSPH and MPSPH methods, the VCSPH method results in faster overall fluid system movement and higher speeds of fluid particles. This method exhibits higher kinetic energy, whereas the MPSPH and HYPSPH methods maintain consistent fluid particle motion. The VCSPH method artificially amplifies the overall energy of the system by amplifying the vorticity, leading to unconservative kinetic energy in the fluid. In contrast to VCSPH, all physical quantity

computations in the MPSPH and our HYPSPH methods are based on the conservation of momentum and energy, ensuring that these methods do not generate additional energy.



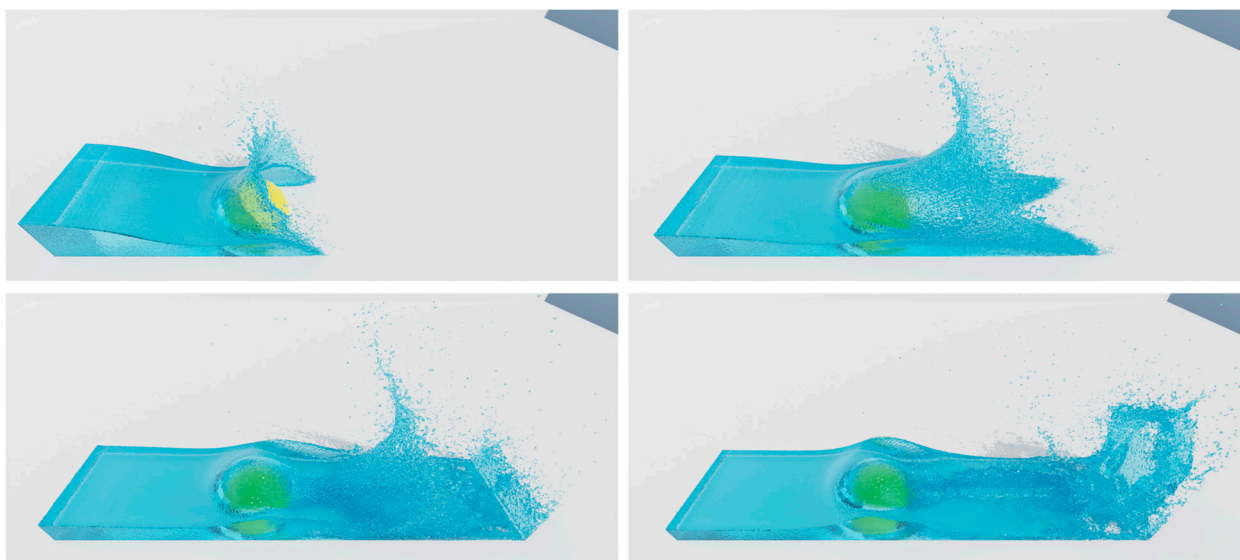
**Figure 4.** Energy Comparison of Fluid Systems in 617 K Particle Hemisphere Dam Break Scenarios.

The magnitude of the adjustment coefficient in the MPSPH method significantly influences the motion of the fluid system. Choosing a smaller adjustment coefficient results in turbulence effects that cannot fully overcome SPH's numerical damping, rendering the restoration of turbulence details less pronounced. Conversely, a larger adjustment coefficient leads to distorted overall motion in the fluid system and non-conservation of energy. The Figure displays experimental results from the same frame using various adjustment coefficients. On the left, results are shown for an adjustment coefficient of  $v_t = 0.5$ , while on the right, the coefficient is set to  $v_t = 1.0$ . Figure 5 illustrates that near the fluid surface on the left of the obstacle ball and the corner of the bounding box, the fluid undergoes spontaneous micro-rotational motions, resulting in distorted movements. Concurrently, the overall velocity on the right side of the obstacle ball is higher, generating additional kinetic energy and leading to energy non-conservation in the system. After experimenting with multiple adjustment coefficients,  $v_t = 0.5$  is selected as the optimal coefficient for the simulation. This value also aligns with the empirical values reported in the relevant literature employing this method.

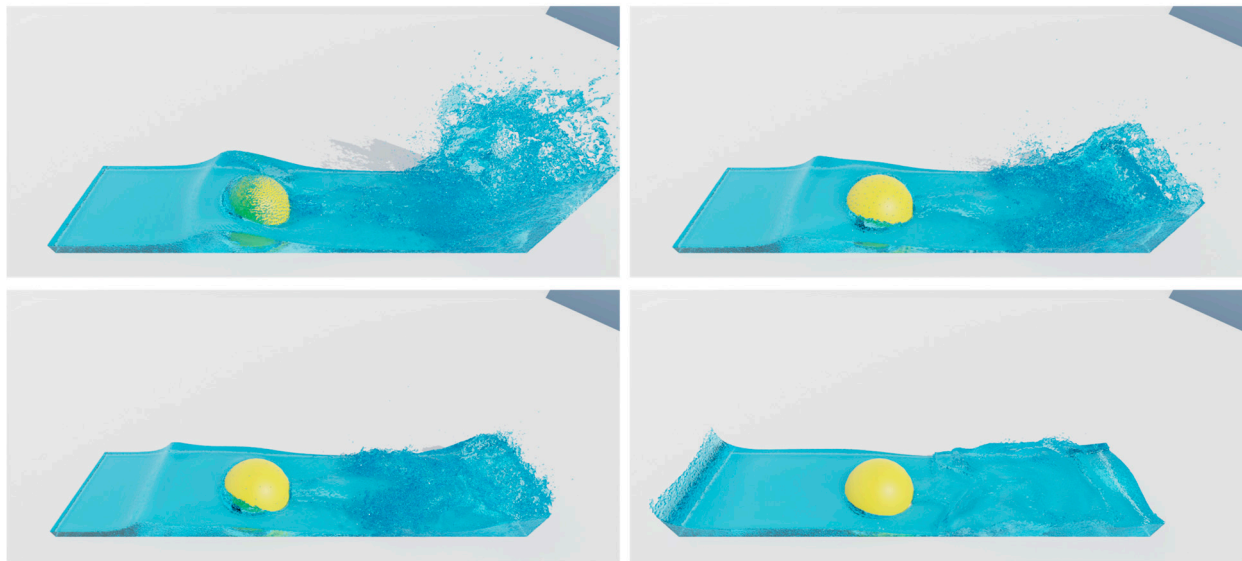


**Figure 5.** Turbulent Performance of the MPSPH Method with Different Adjustment Coefficients. (a)  $v_t = 0.5$ . (b)  $v_t = 1.0$ .

The experimental results depicted in Figure 6 represent MPSPH under optimal conditions, specifically at  $v_t = 0.5$ . It clearly demonstrates that the high-frequency details on the surface of the fluid system are well-preserved, and the flow around the hemispherical obstacle exhibits effective turbulence. Furthermore, the corresponding vortex effect is not compromised by the numerical dissipation inherent in the SPH method, maintaining extensive vortex details.



**Figure 6.** Cont.



**Figure 6.** Overall Experimental Effect Diagram of the MPSPH Method with Adjustment Coefficient  $v_t = 0.5$ .

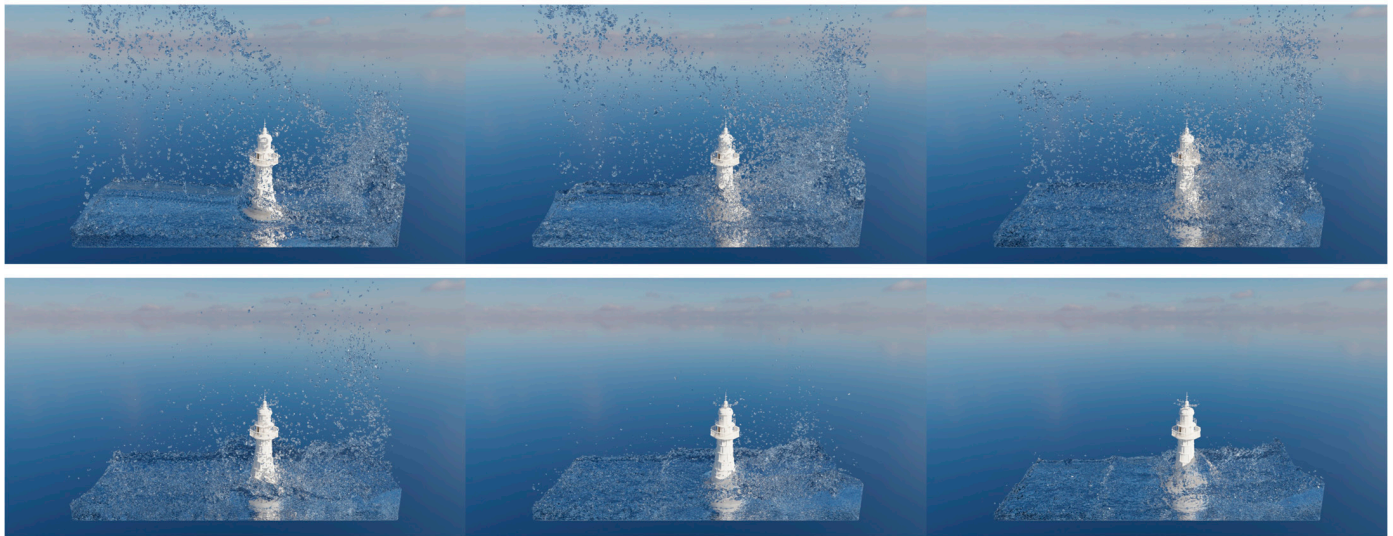
#### 4.3. Analysis of Coupling Effect of Marine Scene

To verify the effectiveness of our proposed novel HYPSPH method and the micropolar fluid model in handling classic navigation scenarios, we designed the following experimental setup. The lighthouse is positioned on the right side of the freely released fluid. Both the bounding box and the lighthouse are sampled as static boundaries using the PDS algorithm. The velocity of these static boundary particles is 0, and their density matches the rest density of the fluid system. Compared to the baffles and hemispheric obstacles in previous experiments, the boundary conditions of the lighthouse model are significantly more complex, meeting the demands of various complex scenes encountered in navigation practice. Therefore, using a lighthouse as an experimental model is representative. Traditional fluid-rigid coupling methods such as collision detection struggle with complex rigid meshes; thus, the lighthouse scenario was selected to demonstrate the flexibility, stability, and applicability of unified boundaries comprehensively. The experiment result in Figure 7 shows that the seawater changes its direction upon coupling with the lighthouse, resulting in water flow around the structure. Simultaneously, a turbulence method applied to the seawater surface generates high-frequency turbulence details. The overall movement of the seawater is more realistic and aligns closely with physical reality.



**Figure 7.** Cont.





**Figure 7.** Classic Navigation Scene: Fluid-Structure Coupling Effect Diagram of Lighthouse Scene.

## 5. Conclusions and Future Work

This paper introduces a novel HYPH method for simulating seawater scenes in maritime simulators. Compared to existing DFSPH methods, our method, which combines a density-constrained solver with a velocity divergence-free solver, achieves faster convergence, enhanced incompressibility, and improved stability. Additionally, we employ the Micropolar Fluid model to conduct high-realism turbulence simulations in the simulator. Compared to the pure SPH method, adding the turbulence model significantly enhances the simulation of surface flow turbulence details and vortex effects. This approach provides a framework for implementing special effects in ocean scenes, such as propeller rotation, splint waves, and ship wake flow. However, the high computational cost of this physically based method limits the range of implementation scenarios, currently allowing small-scale real-time simulations. To maintain simulation quality while meeting real-time requirements, future development of ocean simulators should occur within a parallel framework to simultaneously process the fluid model. Additionally, special attention must be given to the mathematical motion models of various ship types to ensure more realistic ship movement.

**Author Contributions:** Conceptualization, T.Z.; methodology, T.Z. and H.L.; validation, T.Z. and H.R.; experiments, T.Z. and H.L.; visualization, T.Z.; writing—original draft preparation, T.Z.; supervision, H.R. and X.Y.; Formal analysis, H.L., writing—review and editing, H.R., X.Y. and H.L.; funding acquisition, H.R. and X.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Science Foundation of China (Grant No. 52071312), the Key Science and Technology Projects in Transportation Industry (Grant No. 2022-ZD3-035), the Applied Basic Research Program Project of Liaoning Province (Grant No. 2023JH2/101300144), the Guangxi Key Research and Development Plan (Grant No. GUIKE AB22080106), and the Dalian Science and Technology Innovation Fund Project (Grant No. 2022JJ12GX035).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

**Appendix A. The Abbreviations of Proper Nouns (Sort by the Order of Appearance in the Text)**

Abbreviation	Proper Noun
CFD	Computer Fluid Dynamics
SPH	Smoothed Particle Dynamics
GPU	Graphics Processing Unit
WCSPH	Weakly Compressed SPH
PCISPH	Predictive-Corrective Incompressible SPH
PPE	Pressure Poisson Equation
IISPH	Implicit Incompressibility SPH
DFSPH	Divergence-Free SPH
PBF	Position-Based Fluids
NBFLIP	Narrow Band FLIP
MPS	Moving Particle Semi-implicit
HYSPPH	Hybrid SPH
VCSPH	Vorticity Confinement SPH
MPSPH	Micropolar Fluids SPH

**Appendix B. The Explanation of Key Variables**

Variables	Explanation	Variables	Explanation
$\rho_i$	Density of fluid particle	$\mu$	Dynamic viscosity
$\mathbf{v}_i$	Velocity of fluid particle	$W$	Smoothing kernel function
$\sigma_{ij}$	Stress tensor of fluid	$\lambda_i$	Lagrangian Multiplier
$\mathbf{f}_{ext}$	Stress of external body force	$\Theta$	Isotropic microinertia coefficient
$p$	Pressure of fluid particle	$v_t$	Kinematic transfer coefficient
$\omega_i$	Angular velocity		

**References**

- Shabro, V.; Torres-Verdín, C.; Javadpour, F.; Sepehrnoori, K. Finite-difference approximation for fluid-flow simulation and calculation of permeability in porous media. *Transp. Porous Media* **2012**, *94*, 775–793. [\[CrossRef\]](#)
- Li, H. The finite element method. In *Graded Finite Element Methods for Elliptic Problems in Nonsmooth Domains*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–12.
- Koschier, D.; Bender, J.; Solenthaler, B.; Teschner, M. A survey on SPH methods in computer graphics. In *Computer Graphics Forum*; Wiley: Hoboken, NJ, USA, 2022; pp. 737–760.
- Pfefferkorn, R.; Betsch, P. Hourglassing-and locking-free mesh distortion insensitive Petrov–Galerkin EAS element for large deformation solid mechanics. *Int. J. Numer. Methods Eng.* **2023**, *124*, 1307–1343. [\[CrossRef\]](#)
- Gingold, R.A.; Monaghan, J.J. Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Mon. Not. R. Astron. Soc.* **1977**, *181*, 375–389. [\[CrossRef\]](#)
- Monaghan, J.J. Simulating free surface flows with SPH. *J. Comput. Phys.* **1994**, *110*, 399–406. [\[CrossRef\]](#)
- Stam, J.; Fiume, E. Depicting fire and other gaseous phenomena using diffusion processes. In Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 15 September 1995; pp. 129–136.
- 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, San Diego, CA, USA, 26–27 July 2003; pp. 154–159.
- Becker, M.; Teschner, M. Weakly compressible SPH for free surface flows. In Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, CA, USA, 2–4 August 2007; pp. 209–217.
- Solenthaler, B.; Pajarola, R. Predictive-corrective incompressible SPH. *ACM Trans. Graph.* **2009**, *28*, 1–6. [\[CrossRef\]](#)
- Bender, J.; Koschier, D. Divergence-free smoothed particle hydrodynamics. In Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Los Angeles, CA, USA, 7–9 August 2015; pp. 147–155.
- Ihmsen, M.; Cornelis, J.; Solenthaler, B.; Horvath, C.; Teschner, M. Implicit incompressible SPH. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 426–435. [\[CrossRef\]](#)
- Macklin, M.; Müller, M. Position based fluids. *ACM Trans. Graph.* **2013**, *32*, 1–12. [\[CrossRef\]](#)
- Boyd, L.; Bridson, R. MultiFLIP for energetic two-phase fluid simulation. *ACM Trans. Graph.* **2012**, *31*, 16. [\[CrossRef\]](#)
- Ferstl, F.; Ando, R.; Wojtan, C.; Westermann, R.; Thuerey, N. Narrow band FLIP for liquid simulations. In *Computer Graphics Forum*; Wiley: Hoboken, NJ, USA, 2016; pp. 225–232.

16. Li, H.; Ren, H.; Duan, X.; Wang, C. An Improved Meshless Divergence-Free PBF Framework for Ocean Wave Modeling in Marine Simulator. *Water* **2020**, *12*, 1873. [[CrossRef](#)]
17. Liu, K.; Liu, Y.; Li, S.; Chen, H.; Chen, S.; Arikawa, T.; Shi, Y. Coupling SPH with a mesh-based Eulerian approach for simulation of incompressible free-surface flows. *Appl. Ocean Res.* **2023**, *138*, 103673. [[CrossRef](#)]
18. Roy, B.; Poulin, P. A hybrid Eulerian-DFSPH scheme for efficient surface band liquid simulation. *Comput. Graph.* **2018**, *77*, 194–204. [[CrossRef](#)]
19. Shobeyri, G. Simulating Free Surface Flows Using a New Incompressible SPH Model Improved by MPS Method. *Arab. J. Sci. Eng.* **2023**, *49*, 4513–4526. [[CrossRef](#)]
20. Ihmsen, M.; Orthmann, J.; Solenthaler, B.; Kolb, A.; Teschner, M. SPH fluids in computer graphics. In *Eurographics 2014—State of the Art Reports*; The Eurographics Association: Aire-la-Ville, Switzerland, 2014. [[CrossRef](#)]
21. Cornelis, J.; Ihmsen, M.; Peer, A.; Teschner, M. IISPH-FLIP for incompressible fluids. *Comput. Graph. Forum* **2014**, *33*, 255–262. [[CrossRef](#)]
22. 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*, Vancouver, BC, Canada, 5–7 August 2011; pp. 91–100.
23. Mercier, O.; Beauchemin, C.; Thuerey, N.; Kim, T.; Nowrouzezahrai, D. Surface turbulence for particle-based liquid simulations. *ACM Trans. Graph. (TOG)* **2015**, *34*, 1–10. [[CrossRef](#)]
24. Park, S.I.; Kim, M.J. Vortex fluid for gaseous phenomena. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Los Angeles, CA, USA, 29–31 July 2005; pp. 261–270.
25. Zhang, X.; Bridson, R.; Greif, C. Restoring the missing vorticity in advection-projection fluid solvers. *ACM Trans. Graph.* **2015**, *34*, 1–8. [[CrossRef](#)]
26. Edwards, E.; Bridson, R. Detailed water with coarse grids: Combining surface meshes and adaptive discontinuous galerkin. *ACM Trans. Graph. (TOG)* **2014**, *33*, 1–9. [[CrossRef](#)]
27. Chu, M.; Thuerey, N. Data-driven synthesis of smoke flows with CNN-based feature descriptors. *ACM Trans. Graph. (TOG)* **2017**, *36*, 1–14. [[CrossRef](#)]
28. Sato, S.; Dobashi, Y.; Kim, T.; Nishita, T. Example-based turbulence style transfer. *ACM Trans. Graph.* **2018**, *37*, 1–9. [[CrossRef](#)]
29. Jang, T.; Kim, H.; Bae, J.; Seo, J.; Noh, J. Multilevel vorticity confinement for water turbulence simulation. *Vis. Comput.* **2010**, *26*, 873–881. [[CrossRef](#)]
30. Angelidis, A.; Neyret, F. Simulation of smoke based on vortex filament primitives. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Los Angeles, CA, USA, 29–31 July 2005; pp. 87–96.
31. Eberhardt, S.; Weissmann, S.; Pinkall, U.; Thuerey, N. Hierarchical vorticity skeletons. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Los Angeles, CA, USA, 28–30 July 2017; pp. 1–11.
32. Pfaff, T.; Thuerey, N.; Gross, M. Lagrangian vortex sheets for animating fluids. *ACM Trans. Graph. (TOG)* **2012**, *31*, 1–8. [[CrossRef](#)]
33. Selle, A.; Rasmussen, N.; Fedkiw, R. A vortex particle method for smoke, water and explosions. In *ACM SIGGRAPH 2005 Papers*; Association for Computing Machinery: New York, NY, USA, 2005; pp. 910–914.
34. Weißmann, S.; Pinkall, U. Filament-based smoke with vortex shedding and variational reconnection. In *ACM SIGGRAPH 2010 Papers*; Association for Computing Machinery: New York, NY, USA, 2010; pp. 1–12.
35. Golas, A.; Narain, R.; Sewall, J.; Krajcevski, P.; Dubey, P.; Lin, M. Large-scale fluid simulation using velocity-vorticity domain decomposition. *ACM Trans. Graph. (TOG)* **2012**, *31*, 1–9. [[CrossRef](#)]
36. Zhu, B.; Yang, X.; Fan, Y. Creating and Preserving Vortical Details in SPH Fluid. *Comput. Graph. Forum* **2010**, *29*, 2207–2214. [[CrossRef](#)]
37. Wang, X.; Liu, S.; Ban, X.; Xu, Y.; Zhou, J.; Kosinka, J. Robust turbulence simulation for particle-based fluids using the Rankine vortex model. *Vis. Comput.* **2020**, *36*, 2285–2298. [[CrossRef](#)]
38. Lukaszewicz, G. *Micropolar Fluids. Modeling and Simulation in Science, Engineering and Technology*; Birkhäuser Boston Inc.: Boston, MA, USA, 1999.
39. Bender, J.; Koschier, D.; Kugelstadt, T.; Weiler, M. Turbulent Micropolar SPH Fluids with Foam. *IEEE Trans. Vis. Comput. Graph.* **2019**, *25*, 2284–2295. [[CrossRef](#)]
40. Müller, M.; Heidelberger, B.; Hennix, M.; Ratcliff, J. Position based dynamics. *J. Vis. Commun. Image Represent.* **2007**, *18*, 109–118. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.