symmetry

Article

A Symmetric Particle-Based Simulation Scheme towards Large Scale Diffuse Fluids

Sinuo Liu [1,2], **Xiaojuan Ban** [1,2,*], **Ben Wang** [1,2] **and Xiaokun Wang** [1,2]

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Xueyuan Road 30, Haidian District, Beijing 100083, China; liusinuo@xs.ustb.edu.cn (S.L.); s20170708@xs.ustb.edu.cn (B.W.); wangxiaokun@ustb.edu.cn (X.W.)

[2] Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China

* Correspondence: banxj@ustb.edu.cn; Tel.: +86-010-62334980

Received: 28 February 2018; Accepted: 27 March 2018; Published: 29 March 2018

Abstract: We present a symmetric particle simulation scheme for diffuse fluids based on the Lagrangian Smoothed Particle Hydrodynamics (SPH) model. In our method, the generation of diffuse particles is determined by the entropy of fluid particles, and it is calculated by the velocity difference and kinetic energy. Diffuse particles are generated near the qualified diffuse particle emitters whose diffuse material generation rate is greater than zero. Our method fits the laws of physics better, as it abandons the common practice of adding diffuse materials at the crest empirically. The coupling between diffuse materials and fluid is a post-processing step achieved by the velocity field, which enables the avoiding of the time-consuming process of cross finding neighbors. The influence weights of the fluid particles are assigned based on the degree of coupling. Therefore, it improved the accuracy of the diffuse particle position and made the simulation results more realistic. The approach is appropriate for large scale diffuse fluid, as it can be easily integrated in existing SPH simulation methods and the computational overhead is negligible.

Keywords: symmetry; Smoothed Particle Hydrodynamics (SPH); diffuse fluids; physically-based simulation; cooperation visualization

1. Introduction

Diffuse fluids include spray, foam and bubble [1,2]. They are widely presented in metallurgical engineering, industry analysis, virtual reality game, and many other fields. Diffuse materials usually attach to the fluid's boundary or splash around. These phenomena indicate the shape of the boundary and the tendency of the flow. The addition of these details can make the fluid simulation more realistic (Figure 1). However, diffuse materials have many complicated physical natures. It is extremely hard to simulate diffuse fluids exactly. On one hand, diffuse materials are usually generated by the rushing fluid flow which is hard-to-describe. In addition, they need to be deleted dynamically because of the uncertain survival time. On the other hand, the coupling between diffuse materials and fluid is difficult to be modeled.

In fluid simulation field, the Euler method based on the point of flow field space is difficult to model fluids with great deformation due to the limitation of fixed grid [3]. However, the Lagrange method based on fluid particle is suitable for simulating splash scenes with diffuse materials [4]. In this paper, a symmetric particle-based method named Smoothed Particle Hydrodynamics (SPH) [5] is used to handle this procedure. SPH method is one of the Lagrange methods. There are two main difficulties to simulate diffuse fluids. One is how to model the generation and termination of diffuse particles [6]. Due to the unstable state of diffuse materials, they need to be defined as dynamically surviving particles. The complex generating conditions and deleting conditions should be established for diffuse fluids.

Another is how to handle the coupling between diffuse materials and fluid efficiently [7]. Small-scale scenes have higher requirements on the surface characteristics [8,9], while large-scale scenes pay more attention to the accuracy of the description of the movement tendency [10]. At the same time, the number of particles in a large scene increases exponentially, and the process of simulation will consume much time [11].

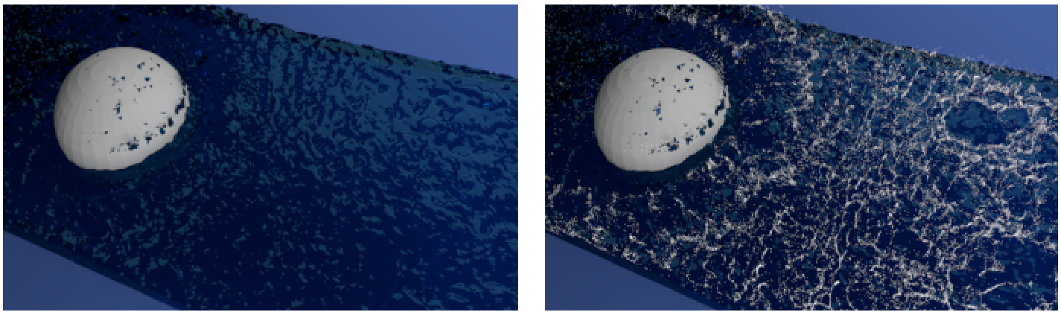


Figure 1. A dam burst hit a ball from the right side. The 60th frame of the experiment. Render result without diffuse materials (**left**). Our method (**right**). When the dam impacts the ball, curved water will be formed around the ball. Compared with the non-diffuse-material results, the rendering result with diffuse materials can capture the direction, shape and other details of fluid more clearly, and the simulation result is more realistic.

A new efficient diffuse fluid simulation scheme based on SPH is proposed in this paper. The generation of diffuse particles is modeled by the entropy of the flow field, which is calculated from the velocity difference and kinetic energy. When the disorder, or randomness of the flow field is large, the diffuse material generation rate of certain fluid particles will be greater than zero, and these particles will be designated as diffusible particle emitters. Diffuse particles are generated near these emitters randomly. To enhance efficiency, the coupling between diffuse materials and fluid is achieved by the velocity field. The weight of the coupling depends on the degree of the interaction between fluid and diffuse materials. This step is a post-processing step, ignoring the extremely small forces exerted by the diffuse material on the fluid, and reduces the time spent on the process of finding neighbors.

2. Related Work

Diffuse materials play an important role in fluid simulations, metallurgical industry and other fields. Although the regular movements of fluid can be successfully achieved with both Eulerian and Lagrangian techniques [4]. The Lagrange method is more suitable for simulating the fluid with great deformation because its particle-based system does not have the limitation of fixed mesh [12]. The schematic diagrams of Eulerian and Lagrangian method shown in Figure 2. In this paper, one of the Lagrange methods named Smoothed Particle Hydrodynamics (SPH) is extended to deal with diffuse fluid.

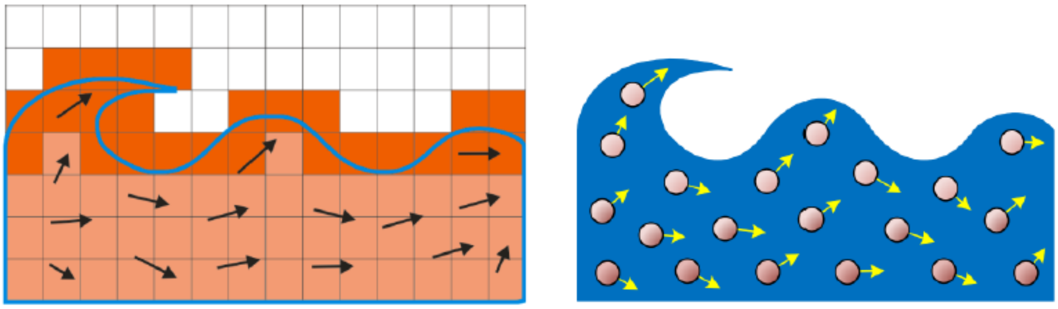


Figure 2. Schematics of two methods for discretizing the flow field. Grid-based Euler method (**left**). Particle-based Lagrange method (**right**).

SPH method was first used in the field of astrophysics [13,14]. It was later introduced by Desbrun et al. [15] into the field of computer graphics to simulate the deformable body. In 2003, Müller et al. [12] used SPH method for real-time interactive simulation of fluid. However, since the ideal gas equation was employed to solve the pressure, the incompressibility of the fluid cannot be guaranteed. In the field of Computational Fluid Dynamics (CFD), Monaghan [16] proposed a SPH method for free surface flow, replacing the ideal gas state equation with Tate equation. This method achieved a weak compressibility of the fluid by assuming a large sound speed and stiffness coefficients, which is very robust and easy to program. In 2007, Becker et al. [17] introduced this idea into the field of fluid animation. Weakly Compressible SPH (WCSPH) significantly increases the realism of the simulation, but it is at the expense of limiting the time step, which affects the efficiency of the algorithm. Physics-based fluid simulation and Computational Fluid Dynamics seem to be similar in mathematical expression, but the two are essentially different. The most significant difference is that fluid simulation as a branch of computer animation pursues realistic picture effects and computational efficiency [18], while CFD as a branch of numerical calculation pursues computational accuracy [19]. In 2009, Solenthaler et al. [20] proposed the Predictive-Corrective Incompressible SPH (PCISPH) approach, using a predictive-corrective method to adjust the pressure by density which avoiding solving complex equations. PCISPH method improved the fidelity and computational efficiency, and it made a significant progress in the field of fluid simulation based on SPH. When updating the position of the particle, the explicit method was originally used for the calculation [12,17,20], which is, using the speed of this time-step to solve the position of following time-step. In 2014, Ihmsen et al. [21] proposed an implicit method to calculate the particle's position of the next time-step with the particle's velocity of the next time-step which making the calculation more accurate. At the same time, the paper gives a new iterative method, and the simulation efficiency has been further improved. In 2015, Bender et al. [22] drew on the idea divergence-free velocity method, enforcing incompressibility of fluid both on position level and velocity level. This method was first proposed by Cummins et al. [23] in 1999, projecting the velocity field onto a divergence-free space by solving a pressure Poisson equation. To improve the ISPH modeling capacity, Gui et al. [24] combined the standard density-invariant and the velocity divergence-free formulations in a weighted average form and proposed a mixed source term model. This model can obtain more accurate impact pressure and force as compared with the results obtained by using either the density-invariant or the velocity divergence-free ISPH model. In 2017, Winchenbach et al. [25] proposed an adaptive SPH method, which changed the previous fixed-mass setting of particles and set the best mass value according to the distance of the particles from the free surface, so that the simulation result effectiveness increased.

The SPH method has been continuously revised to improve its symmetry and accuracy. In the 1980s, Monaghan [26] presented a symmetrization formula to achieve better results. The method is applied to the derivation of the SPH equations which conserve linear and angular momentum exactly. In 1996, Johnson et al. [27] set forth a normalized smoothing function (NSF) algorithm. This approach can obtain exact normal strain rates for conditions of constant linear velocity distributions. In 1999, Bonet et al. [28] presented a discrete variational SPH framework which ensures the balance of linear and angular momentum. In 2008, Khayyer et al. [29] proposed a corrected incompressible SPH (CISPH) method to track water-surface in breaking waves accurately. Corrective terms are derived for the incompressible SPH formulations based on a variational approach, which ensures the preservation of angular momentum. At present, the SPH method can well simulate general fluid dynamics problems.

As a particle-based method, the SPH method is often used to simulate turbulence and sea waves. Back in 2004, Gotho et al. [30] combined the incompressible Smoothed Particle Hydrodynamics (SPH) method with a Large Eddy Simulation (LES) model to study the wave transmission and reflection by a half-immersed curtain breakwater. For the plunging waves, Shao et al. [31] presented a 2-D large eddy simulation (LES) modelling approach in 2006. The computations are in good agreement with the documented data, especially the computed turbulence quantities under the breaking waves. In 2015, Shadloo et al. [32] simulated tsunami runup on idealized geometries for the validation and exploration of three-dimensional flow structures in tsunamis. This study proves that SPH is able to reproduce the runup of long waves for different initial and geometric conditions. In June of 2017, Bender et al. [33] introduced a novel micropolar material model for the simulation of turbulent inviscid fluids. This model is linear and angular momentum conserving, and its computational overhead is negligible.

For diffuse materials, there are many simulation models focused on the behavior of microscopic bubbles [8,34]. In 2011, Ihmsen present a particle-based model for the complex bubble flow with respect to deformation and merging of bubbles and volume-dependent buoyancy [35]. Chentanez used a hybrid grid representation composed of regular cubic cells on top of a layer of tall cells, and focused effort on the area near the surface where it most matters [36]. In these methods, air is considered as a separate phase and the behavior of diffuse fluids is understood as interaction between multiphase flows. However, the simulation of large-scale macro-diffusible fluids remains to be studied. In 2003, Takahashi et al. [37] proposed a method for foam simulation, which combining grid and particle methods to add foam at the crest of wave formation. Although this method had certain visual effects, it ignored the physical reality and do not modeling from the perspective of bubble formation. Due to the lack of theoretical support, the resulting effect has some limitations. In 2009, Mihalef et al. [38] used kinetic energy and surface energy to determine the generation of small-scale splash, which could simulate droplets and bubbles realistically. In 2016, Meric proposed to analyze the acceleration and curvature values to identify the fast and complicated water flows, and foam particle is advected by its classified type based on its velocity [39]. This paper is inspired by those methods mentioned above and a symmetric SPH-based simulation model for diffuse fluid is proposed. Experiments demonstrate that the addition of diffuse materials makes the results more vivid. The simulation results of our method can show the shape and movement trends of fluid more clearly with a negligible computational overhead.

3. Fluid Simulation Model

The addition of diffusion material can enrich the abundance of fluid simulation scene and enhance the fidelity of simulation results. In this paper, the calculation of diffusion material is a post-processing step. The physics-based fluid simulation process needs to be modeled first. This section expounds the physical constraints of the fluid and the SPH simulation method.

3.1. Physical Constraint

Smoothed Particle Hydrodynamics is a particle-based method for fluid simulation. In this scheme, fluid is modeled by a number of particles with attribute such as mass m, position x, and velocity v. Each moving particle is constrained by three conservation laws, including conservation of mass, conservation of momentum and conservation of energy. These constraints are expressed as governing equations of fluid in the field of fluid mechanics:

$$\frac{d\rho}{dt} = -\rho \nabla \boldsymbol{v} \tag{1}$$

$$\frac{d\boldsymbol{v}}{dt} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \boldsymbol{\sigma} + \boldsymbol{a}^{ext} \tag{2}$$

where ρ denotes the density of fluid, t denotes the time, $\boldsymbol{v}$ is the velocity, p is the pressure, $\boldsymbol{\sigma}$ is the viscous stress tensor, and a^{ext} is the acceleration generated by the external force, including gravity, surface tension, adhesion and other forces.

Equation (1) is derived from the law of conservation of mass to ensure the continuity of fluid. Since mass is an inherent property of the particle and the number of particles in the scene remains constant, the law of conservation of mass is automatically satisfied.

Equation (2) is the Navier-Stockes equation (N-S Equation) derived from the law of conservation of momentum, and it is used to describe the fluid's motion. It can be seen that the acceleration of a particle is mainly contribution by the pressure acceleration $a^{pressure} = -\frac{1}{\rho}\nabla p$, the viscous acceleration $a^{viscous} = \frac{1}{\rho}\nabla\sigma$, and external acceleration a^{ext} [40]. The external force is applied to the particles directly, the viscous force is often calculated by a user-defined artificial viscosity, and the pressure is calculated by the equation of state derived from the law of conservation of energy:

$$P = -\frac{\rho_0 c^2}{\gamma}\left(\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right) \tag{3}$$

where $\gamma = 7$, and c denotes the velocity of light.

Equation (3) is the Tate equation for ensuring the incompressibility of fluid, which can be calculated efficiently. Compared with the ideal gas state equation $p = k(\rho - \rho_0)$, this equation solves the problem of high compressibility. Although this equation requires smaller time steps compared to the Poisson equation $\nabla^2 p = \rho\frac{\nabla v}{\Delta t}$, the computation per time step is significantly faster and the implementation easily fits into the SPH framework [17].

The stress state of any point in the continuous flow field can be obtained by solving the above three equations. In addition, to implement fluid simulations, the discretization method is also needed to solve the stress information at each discrete point.

3.2. Smoothed Particle Hydrodynamics

The SPH simulation scheme is essentially an interpolation method. Each particle will carry attributes including mass, position, velocity, etc (Figure 3). The continuous flow field is dispersed as a large number of moving fluid particles. The interpolation result A_i of particle i at position x_i is contributed by all the particles j in the neighborhood, with the weight W related to the position:

$$A_i = \sum_j \frac{m_j}{\rho_j} A_j W_{ij} \tag{4}$$

where m is the mass which is constant, $W_{ij} = W(x_i - x_j, h)$ is the smooth kernel function with homogeneity, and h is the support radius. In this paper, we take W as a B-cubic spline function to ensure the stability, accuracy and efficiency of SPH method.

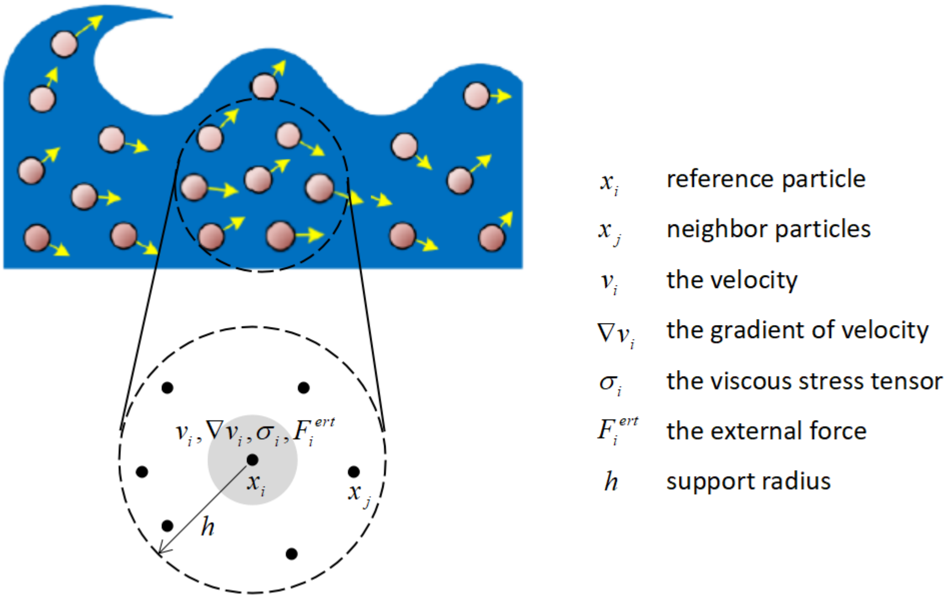

$v_i, \nabla v_i, \sigma_i, F_i^{ert}$
x_i
x_j
h
x_i reference particle
x_j neighbor particles
v_i the velocity
∇v_i the gradient of velocity
σ_i the viscous stress tensor
F_i^{ert} the external force
h support radius

Figure 3. Schematic for the symmetric particle-based method, Smoothed Particle Hydrodynamics (SPH). Fluid is considered to be composed of movable particles that carry physical attributions. The physical attributions of a particle are weighted in summation by its neighbors.

By substituting density into Equation (4), we can get the density at location x_i, which is the last unsolved attribute:

$$\rho_i = \sum_j \frac{m_j}{\rho_j} \rho_j W_{ij} = \sum_j m_j W_{ij} \tag{5}$$

The SPH method has the inherent problem of force asymmetry. Muller [12] solve the problem by using the idea of arithmetic averaging. The new interpolation function is as follows:

$$A_i = \sum_j \frac{m_j}{\rho_j} \frac{A_j + A_i}{2} W_{ij} = \sum_j \frac{m_j}{\rho_j} A_j^{\text{ave}} W_{ij} \tag{6}$$

Equation (6) takes the property values of particle i into account, ensuring the force symmetry. In addition, the gradient of A_i is written as $\nabla A_i = \sum\limits_j \frac{m_j}{\rho_j} A_j^{\text{ave}} \nabla W_{ij}$. Discretize the N-S Equation (2) by Equation (6), and the force exerted by neighbor particles on one particle can be obtained.

In general, the conservativeness of governing equations of fluid is guaranteed by the symmetry of space translation, time translation and rotation transformation. The symmetry of force in the SPH method is guaranteed by the combination of interpolation method and arithmetical mean. Dispersing the flow field into a finite number of moving particles and solving the governing equations of fluid by SPH method, the state of flow field will be obtained. The calculation of diffuse materials is a post-processing step, which will be elaborated in the next section.

4. Diffuse Material Simulation System

Diffuse materials are usually generated by the rushing fluid flow. When the disorder, or randomness of the system become larger, the air will be dragged out and resulting in diffuse fluids. Although current SPH-based simulation methods could describe regular movements of fluid, details such as diffuse materials were often ignored or simplified during modeling process. There have been many studies focusing on the process of bubble in small scenes. However, there are still some problems to be solved for the macroscopic movement of diffuse materials in large scenes.

In this paper, an efficient diffuse fluid simulation scheme based on SPH is presented. Diffuse materials are considered when the diffuse material generation rate calculated by the entropy of the flow field is greater than zero. They are generated by the qualified diffuse particle emitters in fluid particles. The coupling between diffuse materials and fluid is a post-processing step achieved by the velocity field. Three coupling parameters are used to indicate the degree of coupling between the diffuse material and the fluid. Therefore, when the calculation of the flow field by the model above is completed, the generation and termination of diffuse materials and the coupling with other materials will be processed. This approach avoids the time-consuming process of finding neighbors and enhance the efficiency.

4.1. Generation of Diffuse Material

Diffusion Entropy. In some early approaches, diffuse materials are added at the wave crest empirically based on visual effects. However, these methods do not obey the physical laws. In nature, air will be dragged into turbulent fluid flow, resulting in spray, foam and bubble. Slowly flowing streams are always crystal clear, while a huge splash will be sparked when the wave crash into the rock in high speed. Therefore, we consider fluids with more kinetic energy and more chaotic internalities to have a greater ability to produce diffuse materials. The degree of disorder can be expressed by the velocity difference between particles. The simulation of diffuse fluid is processed from the perspective of physics.

The concept of entropy is introduced to describe the potential of flow field to generate diffuse materials. Entropy is a measure of disorder within a macroscopic system or of the availability of the energy in a system to do work [41]. In statistical mechanics, entropy is a measure of the number of ways in which a system may be arranged, often taken to be a measure of ”disorder”. The higher the entropy, the higher the disorder [42]. For fluid, entropy is the degree of disorder within the flow field and the energy it contains. Therefore, the diffusion entropy S_D of the fluid particle i is related to the kinetic energy E and velocity difference $\boldsymbol{v}^{diff}$:

$$S_D \propto E_i \tag{7}$$

$$S_D \propto \boldsymbol{v}_i^{diff} \tag{8}$$

The kinetic energy of particle i is as follows:

$$E_i = 0.5 m_i \boldsymbol{v}_i^2 \tag{9}$$

The velocity difference of particle i can be calculated by interpolation:

$$v_i^{diff} = \sum_j \parallel \boldsymbol{v}_{ij} \parallel \cdot W_{ij}^D \tag{10}$$

where $\boldsymbol{v}_{ij} = \boldsymbol{v}_i - \boldsymbol{v}_j$ is the velocity difference between particle i and particle j, W^D is the kernel function with normality:

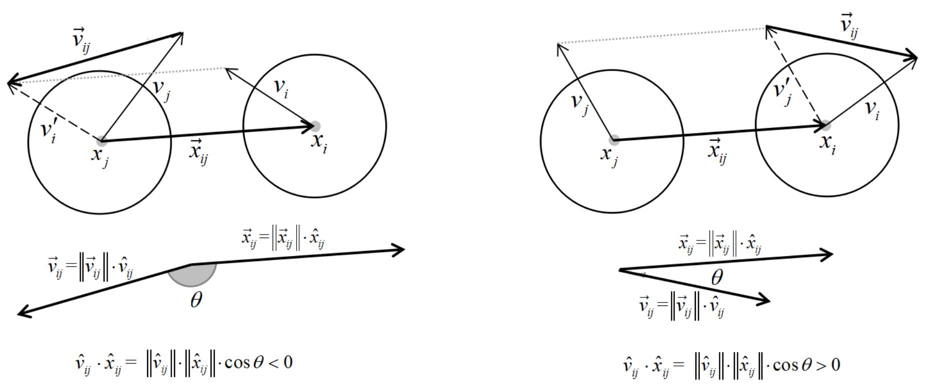
$$W_{ij}^D = W(\boldsymbol{x}_{ij}, h) = \begin{cases} 1 - \frac{\|\boldsymbol{x}_{ij}\|}{h}, & \parallel \boldsymbol{x}_{ij} \parallel < h; \\ 0, & \parallel \boldsymbol{x}_{ij} \parallel \geq h \end{cases} \tag{11}$$

where $\boldsymbol{x}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$ is the position difference vector, h is the support radius. W is an even function whose function image is symmetric about the longitudinal axis.

Equation (10) can calculate the velocity difference between particles i and its neighbors, but it cannot accurately represent the generation probability of the diffuse material. Under the same velocity difference, more air will be mixed in as the two particles move toward each other. Conversely, if two particles are moving away from each other, the amount of diffuse material produced will be less. Therefore, Equation (10) is not sufficient enough to express the amount of bubbles generated by the velocity difference.

In the SPH scheme, $\hat{\boldsymbol{v}}_{ij} \cdot \hat{\boldsymbol{x}}_{ij}$ can be used to indicate whether the two particles are moving away from each other or close to each other, where $\hat{\boldsymbol{v}}_{ij} = \frac{\boldsymbol{v}_i - \boldsymbol{v}_j}{\|\boldsymbol{v}_i - \boldsymbol{v}_j\|}$ is the direction vector of the velocity difference vector, and $\hat{\boldsymbol{x}}_{ij} = \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}$ is the direction vector of the position difference vector (Figure 4). The value of $\hat{\boldsymbol{v}}_{ij} \cdot \hat{\boldsymbol{x}}_{ij}$ is negative when particles are in opposite directions, and positive when particles are backwards. Therefore, we use the parameter λ to determine the additional weight of the velocity difference:

$$\lambda = 1 - \hat{\boldsymbol{v}}_{ij} \cdot \hat{\boldsymbol{x}}_{ij} \tag{12}$$



$\vec{v}_{ij}$
v_j
v_i
v'_i
x_j
$\vec{x}_{ij}$
x_i
$\vec{x}_{ij} = \|\vec{x}_{ij}\| \cdot \hat{x}_{ij}$
$\vec{v}_{ij} = \|\vec{v}_{ij}\| \cdot \hat{v}_{ij}$
θ
$\hat{v}_{ij} \cdot \hat{x}_{ij} = \|\hat{v}_{ij}\| \cdot \|\hat{x}_{ij}\| \cdot \cos\theta < 0$
$\vec{v}_{ij}$
v_j
v'_j
v_i
x_j
$\vec{x}_{ij}$
x_i
$\vec{x}_{ij} = \|\vec{x}_{ij}\| \cdot \hat{x}_{ij}$
θ
$\vec{v}_{ij} = \|\vec{v}_{ij}\| \cdot \hat{v}_{ij}$
$\hat{v}_{ij} \cdot \hat{x}_{ij} = \|\hat{v}_{ij}\| \cdot \|\hat{x}_{ij}\| \cdot \cos\theta > 0$

Figure 4. The effect of $\hat{\boldsymbol{v}}_{ij} \cdot \hat{\boldsymbol{x}}_{ij}$. When two fluid particles move towards each other, $\hat{\boldsymbol{v}}_{ij} \cdot \hat{\boldsymbol{x}}_{ij}$ is less than zero (**left**). When the particle moves backwards, $\hat{\boldsymbol{v}}_{ij} \cdot \hat{\boldsymbol{x}}_{ij}$ is greater than zero (**right**).

The corrected velocity difference equation is as follows:

$$v_i^{diff} = \sum_j \parallel \boldsymbol{v}_{ij} \parallel \cdot \lambda \cdot W_{ij}^D \tag{13}$$

The diffusion entropy of each fluid particle is calculated by the product of kinetic energy and velocity difference. Obviously, the higher the diffusion entropy is, the higher the probability of generating diffuse particles in the vicinity of the fluid particles is. Therefore, the number of diffuse particles generated around each fluid particle is related to the entropy.

Number of Diffuse Particles. In order to solve the number of generated diffuse particles, the mapping function ϕ is used to map diffusion entropy S_D to [0, 1] interval to obtain the diffuse material generation rate R_D. The mapping function for a scalar quantity M is defined as:

$$\phi(M, \varphi^{min}, \varphi^{max}) = \begin{cases} 0, & M \leq \varphi^{min}; \\ \frac{M-\varphi^{min}}{\varphi^{max}-\varphi^{min}}, & \varphi^{min} < M \leq \varphi^{max}; \\ 1, & M > \varphi^{max} \end{cases} \tag{14}$$

where φ^{min} is the upper limit, φ^{max} is the lower limit. Velocity difference and kinetic energy are mapped respectively to obtain accurate calculation results. The diffuse material generation rate R_D is the product of these two value:

$$R_D = \phi(v_i^{diff}, \varphi_v^{min}, \varphi_v^{max}) \cdot \phi(E_i, \varphi_E^{min}, \varphi_E^{max}) \tag{15}$$

where φ_v^{min}, φ_v^{max}, φ_E^{min}, φ_E^{max} are the user-defined parameters.

Throughout the flow field, the diffuse material generation rate R_D of each moving particle is calculated. When the R_D is greater than zero, the particle is considered as a diffuse particle emitter. Diffuse particles are generated in the vicinity of these emitters randomly. The number of diffuse particles produced in each time step is:

$$n_d = R_D \cdot k_{max} \cdot \Delta t \tag{16}$$

where Δt is the time step, k_{max} is the maximum sampling rate, that is, the maximum number of diffuse particles that can be generated in each time step.

Diffuse Particle Emitter. Diffuse particles are randomly generated near the fluid particle selected as diffuse particle emitter. In one time-step, the fluid particles move from position $\boldsymbol{x}_f(t)$ to position $\boldsymbol{x}_f(t + \Delta t)$. In addition, the diffuse particles are generated in the cylinder centered on the line connecting $\boldsymbol{x}_f(t)$ and $\boldsymbol{x}_f(t + \Delta t)$. Its default height is $\parallel \Delta \boldsymbol{x}_f \parallel = \parallel \boldsymbol{v}_f \cdot \Delta t \parallel$, and the default bottom radius r is related to the support radius of fluid particles.

In order to make the generation of diffuse material more realistic, the initialization parameters need to be set randomly. Firstly, a vector $\vec{n}_1$ perpendicular to vector $\Delta \boldsymbol{x}_f$ is chosen randomly. Then, three random variables k_h, k_r, k_θ are set to produce another vector $\vec{n}_2$ that is not linearly related with $\vec{n}_1$ and $\Delta \boldsymbol{x}_f$. The exact location of diffuse particle i is determined by the following parameter:

$$H_i = k_h \cdot \parallel \Delta \boldsymbol{x}_f \parallel \tag{17}$$

$$\theta_i = k_\theta \cdot 2\pi \tag{18}$$

$$r_i = k_r \cdot r \tag{19}$$

where h_i is the height, θ_i is the azimuth, r_i is the distance to the cylinder (Figure 5). When the values of k_h, k_r and k_θ are between 0 and 1, the diffuse particles are generated inside the cylinder. In experiments, when the support radius and time-step is small, the size of the cylinder becomes very small. Newly generated diffuse particles will accumulate together, causing undesirable visual effects. Therefore, the value of the above three parameters can be adjusted according to experiments in order to achieve the best results.

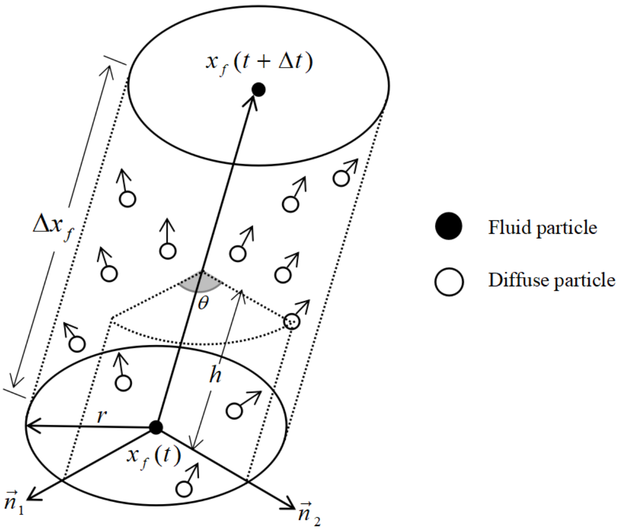


Figure 5. Diffuse particle emitter. Diffuse particles are generated by the qualified fluid particle in the cylinder with radius r and height $\Delta \boldsymbol{x}_f$. The initial result of diffuse particles is determined by the velocity and position of the fluid particles.

The vector $\vec{n}_2$ is calculated by Rodrigues' rotation formula [43]:

$$\vec{n}_2 = cos\theta \cdot \vec{n}_i + (1 - cos\theta)(\vec{n}_i \cdot \hat{\boldsymbol{v}}_f) \cdot \hat{\boldsymbol{v}}_f + sin\theta \cdot \hat{\boldsymbol{v}}_f \times \vec{n}_i \tag{20}$$

where $\hat{\boldsymbol{v}}_f$ is the direction vector of $\boldsymbol{v}_f$.

The initial position and velocity of a diffuse particle are as follows:

$$\boldsymbol{x}_d(t + \Delta t) = \boldsymbol{x}_f(t) + rcos\theta \cdot \vec{n}_i + r \cdot sin\theta \cdot \vec{n}_2 + h\hat{\boldsymbol{v}}_f \tag{21}$$

$$\boldsymbol{v}_d(t + \Delta t) = rcos\theta \cdot \vec{n}_i + r \cdot sin\theta \cdot \vec{n}_2 + \hat{\boldsymbol{v}}_f \tag{22}$$

where, $\vec{n}_1$ and $\vec{n}_2$ act as position in Equation (21), and velocity in Equation (22).

In summary, the number of newly generated diffuse particles per unit time-step is determined by the entropy of the fluid particles, which is calculated from the velocity difference and kinetic energy of the fluid particles. The diffuse material emitter will be designated, and its diffuse material generation rate will be determined by mapping the entropy to [0, 1] interval. The maximum sampling rate is set to control the number of diffuse particles generated. Random parameters are used to initialize diffuse particles, making the distribution of particles more random and experimental results more realistic.

4.2. Interaction with Fluid

The traditional force analysis of microscopic diffuse fluids simulation is complex. Many surface properties are considered including surface tension, boundary handling, etc. In a small-scale scenario, a two-way coupling model needs to be established for the interaction between the fluid and the air, including the description of the surface characteristics at the two-phase interface. The time-consuming cross-neighbor lookup process between fluid particles and foam particles is unavoidable. However, large-scale scenes emphasize the tendency of macroscopic movements, while the requirements for surface characteristics are not high. Therefore, we proposed a simplified one-way coupling model designed to quickly predict macroscopic motion of diffusible materials.

Diffuse particles are essentially air particles. The mass carried by these particles is much smaller than the fluid particles nearby. Therefore, it is difficult for diffuse materials to possess enough momentum to have a significant effect on the movement of the fluid. The force that the diffuse material exerts on the fluid is ignorable. In our method, the calculation process ignores the negligible effect of the diffuse material on the fluid and only solves the effect of the fluid on the diffuse particles. Besides, the number of particles in large scenes increases exponentially, and the process of two-way coupling will consume much time. Our model addresses macroscopic simulations of large-scale scenes and deals with the interaction of diffusible materials with fluids through a one-way coupling model. The coupling between the fluid with the diffuse materials is achieved by the velocity field. In our model, only the fluid neighbors of the fluid and the fluid neighbors of the diffuse materials need to be searched, which greatly saves the simulation time. This process does not require density calculation, avoiding the problem of distortion of the simulation result due to the density instability at the two-phase interface in two-way coupling models. In addition, it is a post-processing step in which the movement of the diffuse materials is not processed until after the state of the fluid has been calculated. Therefore, our method can be easily integrated in existing SPH simulation methods.

The velocity of a diffuse particle is affected by external forces, buoyancy and fluid coupling forces. In the broken wave, part of diffuse particles detaches from the particle cluster as independent flying particles, known as spray. Sometimes, air is trapped in fluid and forms bubbles that rise gradually under buoyancy. Bubbles are also transported by the fluid due to the frictional force. The foam is formed when rising bubbles reach the surface, and it is advected by the water. Therefore, we set three parameters k_{ext}, k_b, k_{cp} to represent the weight of these forces on the velocity of the diffuse particle. The weighted average velocity of the fluid particles around a diffuse particle d at $\boldsymbol{x}_d$ is:

$$\tilde{\boldsymbol{v}}_f(\boldsymbol{x}_d, t+\Delta t)) = \frac{\sum_f \boldsymbol{v}_f(t+\Delta t) W(\boldsymbol{x}_d(t) - \boldsymbol{x}_f(t), h)}{\sum_f W(\boldsymbol{x}_d(t) - \boldsymbol{x}_f(t), h)} \tag{23}$$

where $\boldsymbol{x}_f$ is the position of fluid particles, $\boldsymbol{v}_f(t+\Delta t) = \frac{x_f(t+\Delta t) - x_f(t)}{\Delta t}$. The velocity of a diffuse particle is:

$$\begin{aligned} \boldsymbol{v}_d(t+\Delta t) &= \boldsymbol{v}_d(t) + \boldsymbol{v}_{ext} + \boldsymbol{v}_b + \boldsymbol{v}_{cp} \\ &= \boldsymbol{v}_d(t) + k_{ext} \cdot (\boldsymbol{g} \cdot \Delta t) + (-k_b \cdot \boldsymbol{g} \cdot \Delta t) + k_{cp} \cdot \left(\tilde{\boldsymbol{v}}_f(t+\Delta t) - \boldsymbol{v}_d(t)\right) \end{aligned} \tag{24}$$

where $k_{ext} = 1$ consistently, k_b is the rising parameter to determine the speed of the rise of diffuse particles, k_{cp} is the following parameter to indicate how much the diffuse particles are affected by the neighboring fluid particles. When $k_{cp} = 1$, diffuse particles move completely with the fluid particles.

The position of the foam particle is updated as follows:

$$\boldsymbol{x}_d(t+\Delta t) = \boldsymbol{x}_d(t) + \Delta t \boldsymbol{v}_d(t+\Delta t) \tag{25}$$

Due to the different states of spray, foam, and bubble, the three parameters in the velocity formula have different values. As shown in Figure 6, the diffuse particles are divided into three categories according to the number of neighbor fluid particles. When the number of fluid neighbors is between 6 and 20, the diffuse particle is considered as foam. Diffuse particles with too few fluid neighbors are classified as spray, while others are considered as bubbles.

Spray is the free-flying particle that breaks away from the fluid during exercise. Spray particles move along a parabola by gravity, and their trajectories are unaffected by fluid particles (Figure 7). The velocity of the spray particle is only affected by the acceleration of the external force $\boldsymbol{F}_{ext}(x) = m_d \cdot \boldsymbol{g}$, $k_b = k_{cp} = 0$:

$$\boldsymbol{v}_{spray}(t+\Delta t) = \boldsymbol{v}_{spray}(t) + \boldsymbol{g} \cdot \Delta t \tag{26}$$

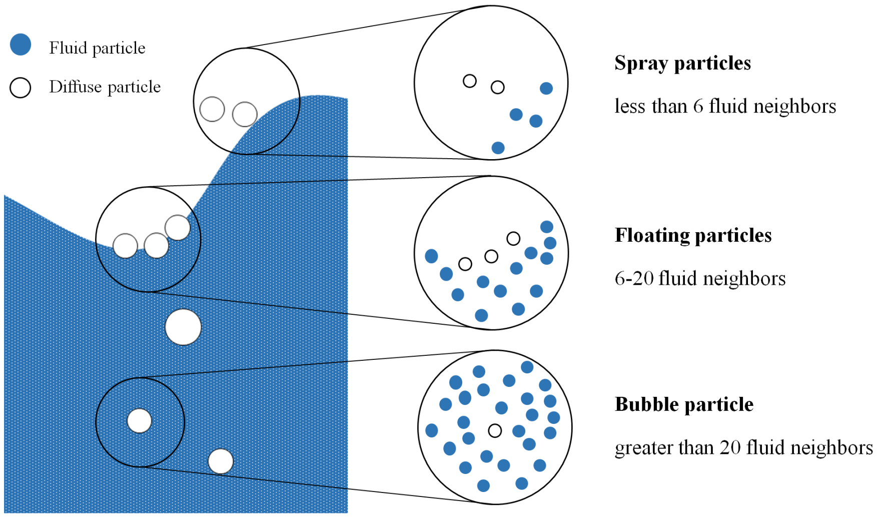


Figure 6. The different between spray, foam and bubble.

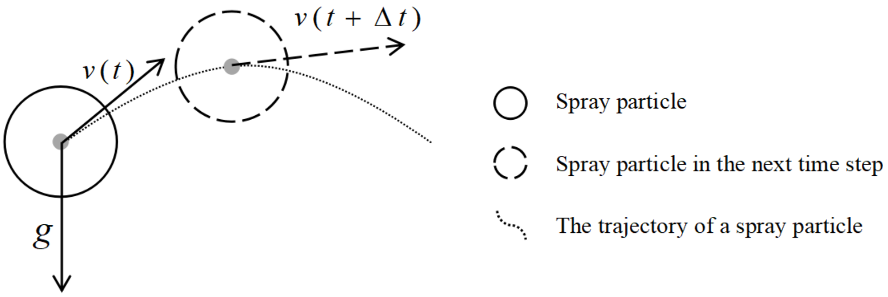

$v(t+\Delta t)$
$v(t)$
g
Spray particle
Spray particle in the next time step
The trajectory of a spray particle

Figure 7. The trajectory of a spray particle.

Unlike spray, foam is on the surface of the water. The buoyancy and the gravity of foam particles are balanced, and there is no need to account for buoyancy and gravity changes. Therefore, $k_{ext} = k_b = 0$. However, the foam contacts the free surface of the fluid, and its velocity is affected by the fluid neighbors (Figure 8). The velocity of a foam particles is update as:

$$v_{foam}(t+\Delta t) = v_{foam}(t) + k_{cp} \cdot \left(\tilde{v}_f(t+\Delta t) - v_{foam}(t) \right) \tag{27}$$

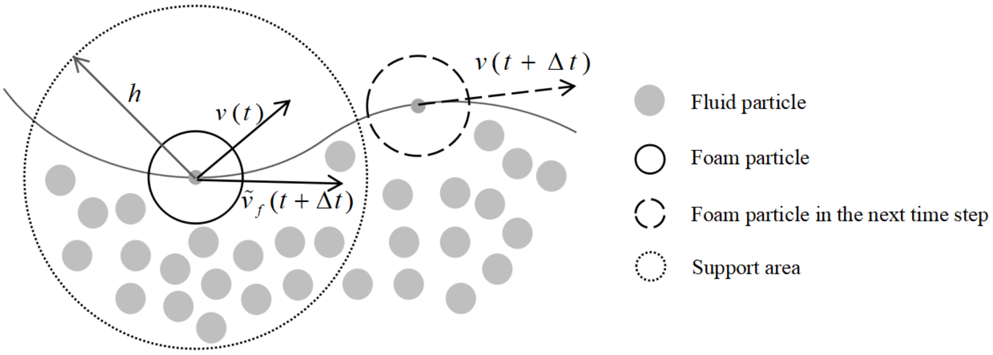


Figure 8. Interaction between a foam particle and fluid particles.

The movement of bubble is more complicated. Bubble particles are not only affected by the neighbor fluid particles, but also lifted by the buoyancy (Figure 9). Therefore, the velocity update formula of bubble particles is expressed as:

$$\begin{aligned} \boldsymbol{v}_{bubble}(t+\Delta t) &= \boldsymbol{v}_{bubble}(t) + k_{ext}\cdot(\boldsymbol{g}\cdot\Delta t) + (-k_b\cdot\boldsymbol{g}\cdot\Delta t) + k_{cp}\cdot(\tilde{\boldsymbol{v}}_f(t+\Delta t) - \boldsymbol{v}_{bubble}(t)) \\ &= \boldsymbol{v}_{bubble}(t) + (1-k_b)\cdot\boldsymbol{g}\cdot\Delta t + k_{cp}\cdot(\tilde{\boldsymbol{v}}_f(t+\Delta t) - \boldsymbol{v}_{bubble}(t)) \end{aligned} \tag{28}$$

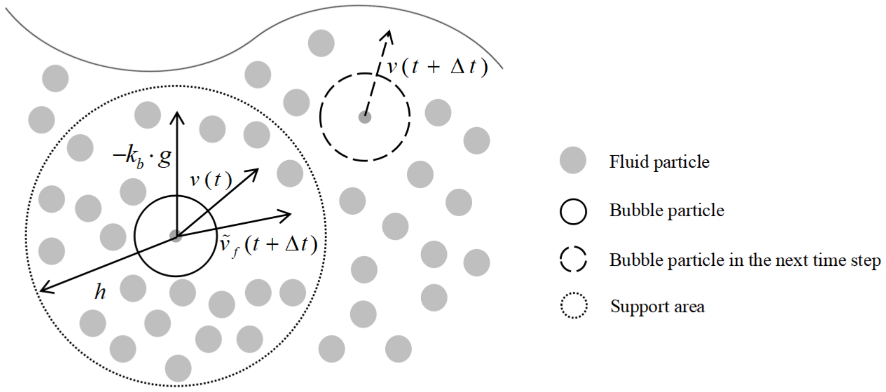


Figure 9. Interaction between a bubble particle and fluid particles.

In this paper, the velocity field is used to deal with the coupling between the fluid and the diffuse materials. It avoids the searching for the neighbor of diffuse particles and greatly reduces the consumption of computing resources. Diffuse particles are classified into three categories when dealing with the interaction of diffuse materials with fluid. Spray, foam and bubble have different properties, and their interactions with the fluid is significantly different. The coupling parameters are set to different values according to the degree of coupling with the fluid, and the physical properties of different diffuse materials have been fully described.

4.3. Termination of Diffuse Material

In nature, diffuse material only has a short uncertain lifetime. Therefore, diffuse particles are defined as dynamically surviving particles in this paper. In order to describe the random disappearance of diffuse materials, each diffuse particle is assigned a random remaining lifetime t_{remain} during initialization. Within each time step, t_{remain} will decrease Δt. The diffuse particles are deleted dynamically when the remaining lifetime runs out.

It is not hard to find out that foams disappear more slowly where they accumulate. In addition, bubble particles live longer than foam particles. Thus, the parameter k_t is used to denote the weight of the reduction in each time step. The value of k_t depends on the type of diffuse materials. For the foam particles, $k_t = 1$ consistently. Experimental results will be better when the parameters of the spray and bubble are less than 1. The remaining lifetime for a diffuse particle is:

$$\begin{aligned} t_{remain} &= t_{remain} - \sum(k_t\cdot\Delta t) \\ &= t_{remain} - k_t\sum\Delta t \end{aligned} \tag{29}$$

5. Implementation

In the SPH scheme, we employ the cubic spline kernel [5] as the smooth kernel in interpolations. We use compact hashing proposed in [44] to find neighbor particles. For the adaptive time-stepping schemes, we use the method explained in [45]. We employed the fluid-solid coupling method proposed in [46] and Bullet [47] for simulating rigid bodies. In following experiments, surface is reconstructed by anisotropic method proposed in [48]. The fluid is reconstructed into grid file according to the position of particles, and it is rendered as transparent liquid using the software named Blender [49]. Foam particles are assigned to a number of tiny spheres and rendered as tiny particles according to their position. The application of our symmetric particle-based simulation scheme towards large scale diffuse fluids is presented in Algorithm 1.

Algorithm 1: Large Scale Diffuse Fluids Simulation Scheme.

```
while animating do
    for each fluid particle $i$ do
        find neighbors
    end for
    for each fluid particle $i$ do
        compute the density $\rho_{f_i}(t)$, pressure $p_{f_i}(t)$
        compute acceleration $a_i$
    end for
    for each fluid particle i do
        update $\boldsymbol{v}_i$, $\boldsymbol{x}_i$
    end for
    for each fluid particle $i$ do
        compute the velocity difference $v_i^{diff}$ and kinetic energy $E_i$
        compute the diffuse material generation rate $R_D$
        compute the number of newly generated diffuse particles $n_d$
        cerate diffuse particles and set $\boldsymbol{v}_d$, $\boldsymbol{x}_d$,$t_{remain}$
    end for
    for each fluid particle $i$ do
        if $t_{remain} < 0$ then
            delete diffuse particle
        else
            find fluid neighbors and classify
            update $\boldsymbol{v}_d$, $\boldsymbol{x}_d$
            set $t_{remain} = t_{remain} - \Delta t$
        end if
    end for
    set $t = t + \Delta t$
end while
```

6. Experimental Section and Results

The following experiments were performed in order to verify the effectiveness of the algorithm. The platform for these experiments is Intel (R) Xeon (R) CPU E5-2637W v2 @ 3.50 GHz 80.0 GB memory (Intel, Santa Clara, CA, USA), 64-bit Windows operating system, rendering platform for Intel (R) Xeon (R) CPU E5-2687W v4 @ 3.00 GHz 72.0 GB memory (Intel, Santa Clara, CA, USA), 64-bit Windows operating system.

6.1. Sea Water Hit Static Rigid Body

The experimental scenes were based on a dam burst collides static rigid body. Rigid bodies of different shapes were selected to validate different aspects of our algorithm. In these experiments, the fluid is rendered as dark blue transparent liquid. Diffuse particles are rendered as tiny white spheres. Rigid bodies are rendered as white solids. In this subsection, we compare the visual effect of no-diffuse fluid, our method and adding foam at the wave crest. We also verified the influence of three coupling parameters on the experimental results.

The result of Figure 1 (page 2) shows the effects of diffuse materials on the simulation results. Seawater flows from right to left and hits the ball. In non-diffuse-material results, the boundary of the fluid is not clear, and the flow tendency is hard to determine. With the addition of diffuse materials, the curved water ring appearing around the sphere can be observed. As the experimental results show, the rendering result with diffuse materials can clearly capture details such as direction and shape of the fluid flow, and the simulation results are more realistic compared with the other method.

Figure 10 shows the comparison between the diffuse materials generated at location with larger entropy and the diffuse materials generated at wave crests. The water was set on the right side of the scene, colliding with the rigid body and scene boundaries in turn and bouncing to form waves. Obviously, the amount of diffuse materials generated by this method is more than that in previous methods, and the initial positions of diffuse particles are more accurate. The parameters in the experiment are shown in Table 1.

Figure 11 verifies the effects of three coupling parameters. In the experimental results without considering the coupling parameters, the distribution of diffuse particles is scattered. The calculation of the diffuse particle velocity is inaccurate, and the resulting position has obvious deviations. In the experimental results obtained, diffuse materials adhere to the fluid boundary. The trend of the flow and the boundary shape of fluid can be observed clearly.

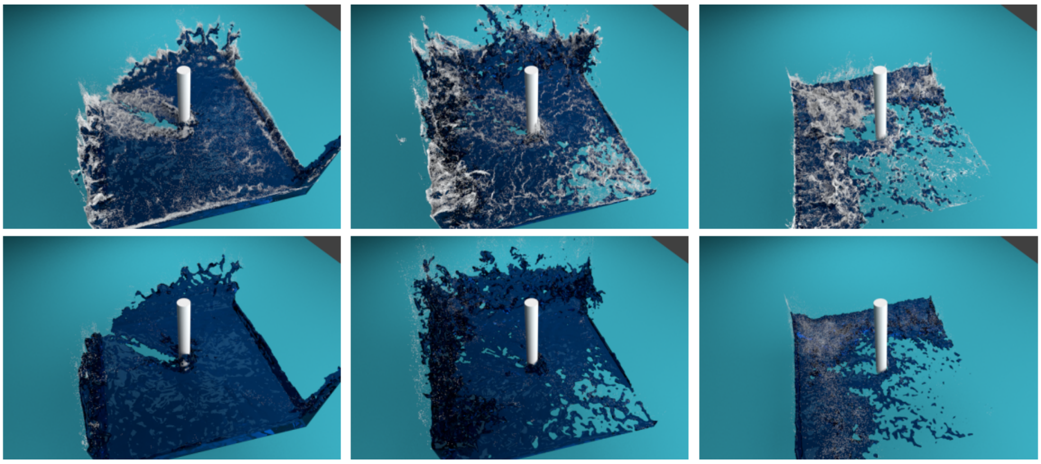


Figure 10. Comparisons of the experimental results of our method (**upper row**) and adding foam at the wave crest (**lower row**). A dam burst collides a static cylinder. From left side to the right side in order of the 30th, 42th, 60th frames.

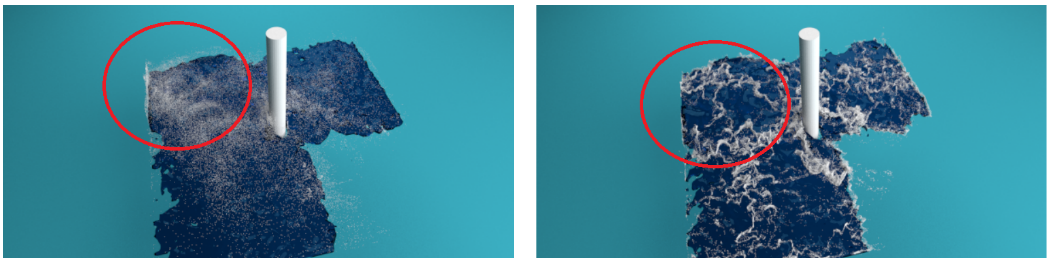


Figure 11. Comparisons of the experimental results without considering coupling parameters (**left**) and our method (**right**). The 80th frame of a dam burst collides a static cylinder. Considering the coupling parameters, the diffuse material is more in line with the fluid boundary, the experimental effect is more realistic.

Table 1. The setting and statistics of a dam burst colliding static cylinder simulation.

Name	Value	Unit
Simulation domain	$128 \times 128 \times 128$	m^3
Initial water body	$38.4 \times 128 \times 38.4$	m^3
Fluid particles	1.887×10^8	-
Foam particles(max)	9.90022×10^5	-
Fluid particle radius	0.5	m
Support radius	1	m
Program run time	113	s/frame
Frame rate	30	s/frame
Rendering time	97	min/frame
Sampling frequency	200	-
φ_v^{min}	2.5	-
φ_v^{max}	10	-
φ_E^{min}	0	-
φ_E^{max}	60	-
Mass force acceleration	$(0.0, -25.0, 0.0)$	32751
Survival time (max)	3.0	s

6.2. Dam Burst Hit Movable Body

The phenomenon of fluid interacting with movable rigid bodies and creating diffuse materials is often observed in nature. Two movable rigid cylinders were employed to impact with a dam break to demonstrate that our model can simulate this effect. Water flows from below to above, hitting two cylinders and interacting with them to form splash and droplets. The method described in [46] is used to deal with the fluid-solid coupling, so the rigid body needs to be sampled. The sampling result is shown in Figure 12. In this experiment, the fluid is rendered as light blue transparent liquid. Diffuse particles are rendered as tiny translucent spheres. Figure 12 shows the experimental results of this scene. Fluid and rigid bodies collide to produce clusters of spray, foam and bubbles. As time move forward, diffuse particles splash around and eventually disappear. In this scenario, we found the following thresholds which produce expected results: $k_b = 5.6$, $k_{cp} = 0.5$, $k_t^{spray} = 0.5$, $k_t^{foam} = 1.0$, $k_t^{bubble} = 0.33$. The emission range of diffuse material emitters is magnified twenty times, $k_h = k_r = 20$. The parameters in the experiment are shown in Table 2.

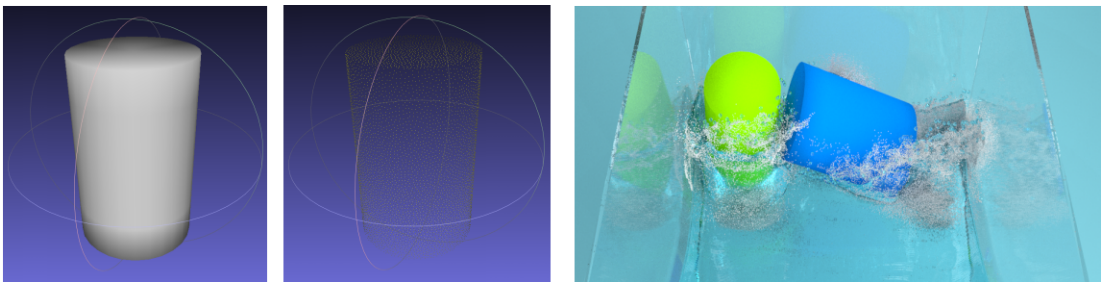


Figure 12. A dam burst collide two moving cylinders. The 31th frame of this experiment (**rifht**). The grid file (**left**) and the sample file (**middle**) of the solid cylinder.

Table 2. The setting and statistics of a dam burst colliding movable cylinder simulation.

Name	Value	Unit
Simulation domain	$240 \times 240 \times 240$	m^3
Initial water body	$33 \times 64 \times 96$	m^3
Fluid particles	1.622×10^6	-
Foam particles (max)	6.5×10^5	-
Fluid particle radius	0.5	m
Support radius	1	m
Program run time	128.7	s/frame
Rendering time	15	min/frame
Sampling frequency	30	-
φ_v^{min}	1.5	-
φ_v^{max}	5	-
φ_E^{min}	2	-
φ_E^{max}	7	-
Mass force acceleration	$(0.0, -9.8, 0.0)$	$\mathrm{kg/m}^3$
Survival time (max)	3.0	s

6.3. Diffuse Generated by Moving Body

Our diffuse module is a post-process step applied to particle-based fluid. Diffuse materials are divided into spray, foam and bubbles. In this experiment, nine rigid bodies of different shapes fall into the water under the action of gravity and produce a large number of diffuse particles. Different diffuse materials are marked with different colors in the rendering result (Figure 13). We can clearly observe the difference between them. The visual effect of the experimental results can be controlled by limiting the maximum number of diffuse particles. The number of diffuse particles as a function of time is shown in Figure 14, with a max diffuse particle number of 5.5×10^5. In life, bubbles will be buoyant and reach the surface of the water, and the droplets will fall back to the surface under the influence of gravity. Both of the two diffuse materials eventually become floating foam. Our experimental results show that all diffuse materials will become floating foam and eventually disappear over time. Therefore, our model is in line with objective laws. In this scenario, we found the following thresholds which produce expected results: $k_b^{foam} = 0.5$, $k_b^{bubble} = 0.3$, $k_{cp} = 0.6$, $k_t^{spray} = 0.67$, $k_t^{foam} = 1.0$, $k_t^{bubble} = 0.33$. The emission range of diffuse material emitters is magnified twenty times, $k_h = k_r = 8.5$. The parameters in the experiment are shown in Table 3.

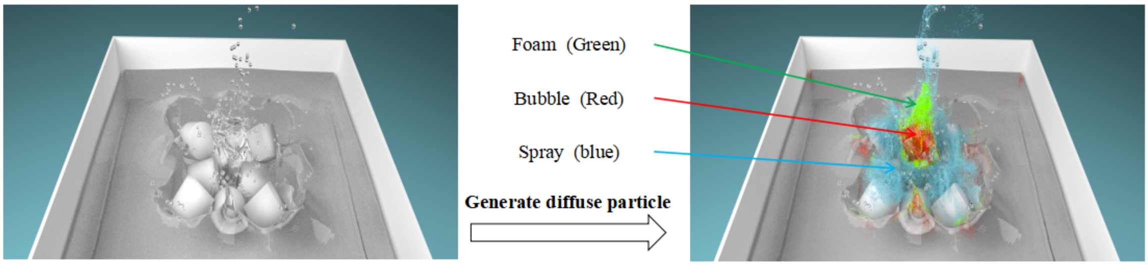


Figure 13. Add diffuse materials to the fluid simulation by post-processing steps. The **left** one shows the fluid simulation and its rendering results. The **right** one shows the experimental results with the addition of diffuse materials. Different diffuse materials are marked with different colors.

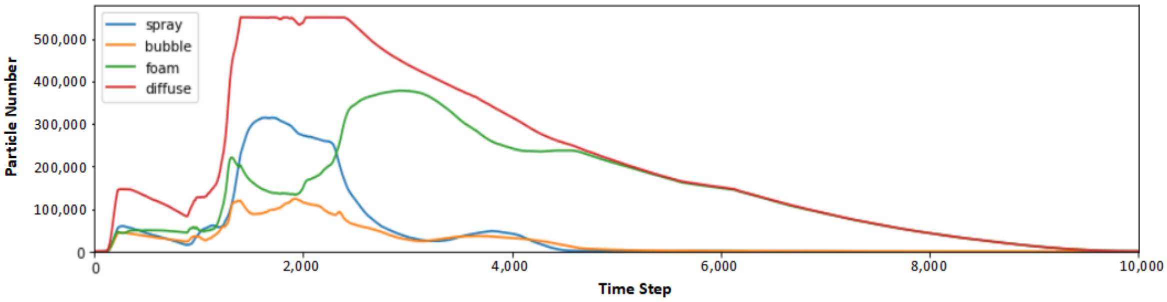


Figure 14. The number of diffuse particles over time. The max diffuse particle number is set to 5.5×10^5. As time passes, all diffuse materials will become floating foam and eventually disappear.

Table 3. The setting and statistics of nine rigid bodies fall into the water.

Name	Value	Unit
Simulation domain	240 × 240 × 240	m^3
Initial water body	96 × 10 × 96	m^3
Fluid particles	7.373×10^8	-
Foam particles (max)	5.5×10^5	-
Fluid particle radius	0.05	m
Support radius	0.2	m
Program run time	36	s/frame
Rendering time	30	min/frame
Sampling frequency	30	-
φ_v^{min}	2	-
φ_v^{max}	8	-
φ_E^{min}	2	-
φ_E^{max}	8	-
Mass force acceleration	(0.0, −50, 0.0)	kg/m^3
Survival time (max)	3.0	s

We establish a one-way coupling model to handle the coupling process between fluids and diffusible materials. The interaction between the two is achieved through the velocity field, where the movement of the fluid is not affected by the diffuse materials. In two-way coupling, a cross neighbor finding is performed between the fluid particles and the diffuse particles. This process is very time consuming. In our model, only the fluid neighbors of the fluid and the fluid neighbors of the diffuse materials need to be searched, which greatly saves the simulation time (Figure 15).

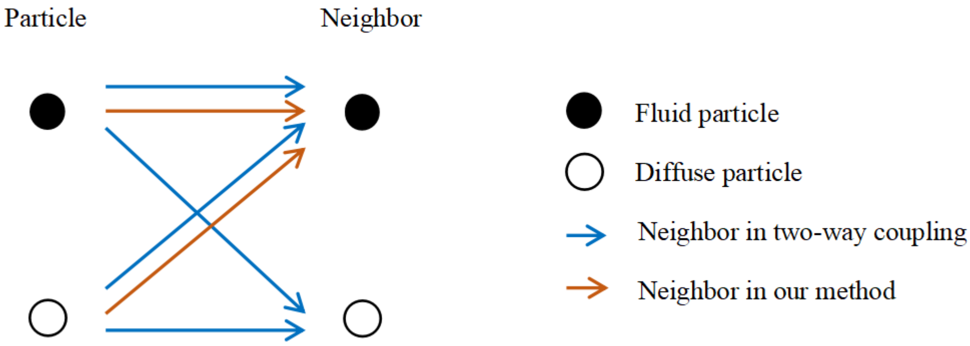


Figure 15. Comparison of the neighbor finding process in two-phase coupling and our method.

In order to verify the efficiency of our model, we compared the computational time of fluid simulation without diffuse materials, our method and the two-way coupling algorithm (Figure 16). In the experiment, the time of the two-way coupling algorithm is obtained by calculating the time of cross-neighbor finding. When the diffusible particles have not yet been generated, the calculation time of the three methods is generally the same. As the number of diffusible particles decreases, the time required for biphase coupling gradually decreases. Experimental results show that our method can handle fluid coupling with diffusible materials more efficiently. The computational overhead of our method is negligible.

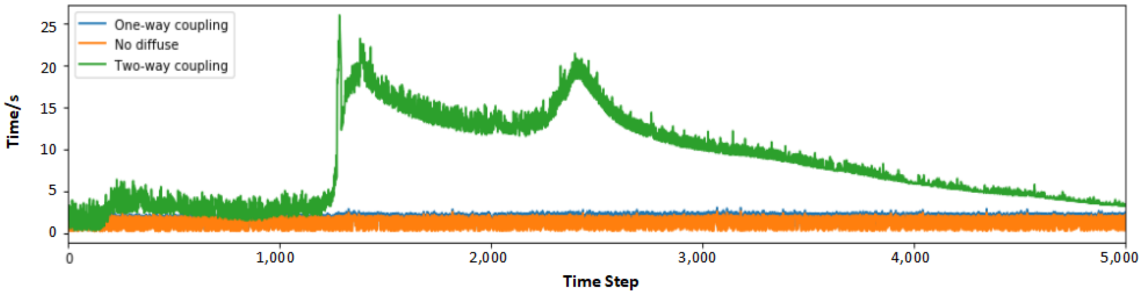


Figure 16. Comparison of the processing time of fluid simulation (yellow), our method (blue) and two-phase coupling method (green). When dealing with the coupling of fluids with diffusible materials, the computational overhead of our method is negligible. The calculation time required for the two-phase coupling algorithm is positively related to the number of diffusible particles.

7. Conclusions

In this paper, a symmetric particle-based simulation scheme is proposed for diffuse materials. The entropy of flow field is used to estimate the number of new generated diffuse particles. The velocity differences and kinetic energy are considered to determine the entropy of fluid particles. Comparing to original methods based on experience, our model is more in line with the laws of physics. The coupling between diffuse materials and fluid is achieved by the velocity field. This post-processing step avoid the time-consuming process of finding neighbors and improve the efficiency. Equations with different weights are used to describe the degree of coupling between diffuse materials and fluid, which improved the accuracy of the position of diffuse particles and improved the quality of simulation significantly. It is found that the diffuse material can be well displayed in all cases, and the proposed SPH algorithm is stable and effective.

We further believe when more details like the temperature are added to model, one could begin to simulate phenomena of air-liquid conversion, e.g. water boils at high temperatures.

Acknowledgments: The authors acknowledge the financial support from the National Science Foundation of China (No. 61572075, No. 61702036), and the National Key Research and Development Program of China (No. 2016YFB0700500).

Author Contributions: Sinuo Liu conceived the framework and wrote the paper; Ben wang performed the experiments; Xiaokun Wang checked the results; all the work was done under the guidance of Xiaojuan Ban.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shadloo, M.S.; Oger, G.; Le Touzé, D. Smoothed particle hydrodynamics method for fluid flows, towards industrial applications: Motivations, current state, and challenges. *Comput. Fluid.* **2016**, *136*, 11–34. [CrossRef]
2. Zainali, A.; Tofighi, N.; Shadloo, M.S.; Yildiz, M. Numerical investigation of Newtonian and non-Newtonian multiphase flows using ISPH method. *Comput. Methods Appl. Mech. Eng.* **2013**, *254*, 99–113. [CrossRef]
3. Munson, B.R.; Okiishi, T.H.; Rothmayer, A.P. Fundamentals of Fluid Mechanics; John Wiley and Sons: Hoboken, NJ, USA, **2014**. [CrossRef]

4. Liu, G.R.; Liu, M.B. Smooth particle hydrodynamics: A meshless particle method. *World Sci.* **2003**, 33, *491*.
5. Monaghan, J.J. Smoothed particle hydrodynamics. *Rep. Prog. Phys.* **2005**, *68*, 1703. [CrossRef]
6. Tabib, M.V.; Roy, S.A.; Joshi, J.B. CFD simulation of bubble column—An analysis of interphase forces and turbulence models. *Chem. Eng. J.* **2008**, *139*, 589–614. [CrossRef]
7. Busaryev, O.; Dey, T.K.; Wang, H.; Ren, Z. Animating bubble interactions in a liquid foam. *ACM Trans. Graph. (TOG)* **2012**, *31*, 63. [CrossRef]
8. Kim, B.; Liu, Y.; Llamas, I.; Jiao, X.; Rossignac, J. Simulation of bubbles in foam with the volume control method. *ACM Trans. Graph. (TOG)* **2007**, *26*, 98. [CrossRef]
9. Durian, D.J. Foam mechanics at the bubble scale. *Phys. Rev. Lett.* **1995**, *75*, 4780. [CrossRef]
10. Losasso, F.; Talton, J.; Kwatra, N.; Fedkiw, R. Two-way coupled SPH and particle level set fluid simulation. *IEEE Trans. Vis. Comput. Graph.* **2008**, *14*, 797–804. [CrossRef]
11. Thurey, N.; Muller-Fischer, M.; Schirm, S.; Gross, M. Real-time breaking waves for shallow water simulations. In Proceedings of the 15th Pacific Conference on Computer Graphics and Applications, Maui, HI, USA, 29 October–2 November 2007; pp. 39–46. [CrossRef]
12. 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. [CrossRef]
13. Lucy, L.B. A numerical approach to the testing of the fission hypothesis. *Astron. J.* **1977**, *82*, 1013–1024. [CrossRef]
14. 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]
15. Desbrun, M.; Gascuel, M.P. Smoothed particles: A new paradigm for animating highly deformable bodies. In Proceedings of the Eurographics Workshop on Computer Animation and Simulation 96, Poitiers, France, August 31–September 1 1996; pp. 61–76. [CrossRef]
16. Monaghan, J.J. Simulating free surface flows with SPH. *J. Comput. Phys.* **1994**, *110*, 399–406. [CrossRef]
17. 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. [CrossRef]
18. Bridson, R. *Fluid Simulation for Computer Graphics*; CRC Press: Hoboken, NJ, USA, 2008 [CrossRef]
19. Anderson, J.D.; Wendt, J. *Computational Fluid Dynamics*; McGraw-Hill: New York, NY, USA, 1995. [CrossRef]
20. Solenthaler, B.; Pajarola, R. Predictive-corrective incompressible sph. *ACM Trans. Graph.* **2009**, *28*, 341–352. [CrossRef]
21. Ihmsen, M.; Cornelis, J.; Solenthaler, B.; Horvath, C.; Teschner, M. Implicit incompressible sph. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 426–435. [CrossRef]
22. 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. [CrossRef]
23. Cummins, S.J.; Rudman, M. An SPH projection method. *J. Comput. Phys.* **1999**, *152*, 584–607. [CrossRef]
24. Gui, Q.; Dong, P.; Shao, S. Numerical study of PPE source term errors in the incompressible SPH models. *Int. J. Numer. Methods Fluid.* **2015**, *77*, 358–379. [CrossRef]
25. Winchenbach, R.; Hochstetter, H.; Kolb, A. Infinite continuous adaptivity for incompressible SPH. *ACM Trans. Graph. (TOG)* **2017**, *36*, 102. [CrossRef]
26. Monaghan, J.J. An introduction to SPH. *Comput. Phys. Commun.* **1988**, *48*, 89–96. [CrossRef]
27. Johnson, G.R.; Beissel, S.R. Normalized smoothing functions for SPH impact computations. *Int. J. Numer. Methods Eng.* **1996**, *39*, 2725–2741.
28. Bonet, J.; Lok, T.S.L. Variational and momentum preservation aspects of smooth engineering. *Comput. Methods Appl. Mech. Eng.* **1999**, *180*, 97–115. [CrossRef]
29. Khayyer, A.; Gotoh, H.; Shao, S.D. Corrected incompressible SPH method for accurate water-surface tracking in breaking waves. *Coast. Eng.* **2008**, *55*, 236–250.[CrossRef]
30. Gotoh, H.; Shao, S.; Memita, T. SPH-LES model for numerical investigation of wave interaction with partially immersed breakwater. *Coast. Eng. J.* **2004**, *46*, 39–63. [CrossRef]
31. Shao, S.; Ji, C. SPH computation of plunging waves using a 2-D sub-particle scale (SPS) turbulence model. *Int. J. Numer. Methods Fluid.* **2006**, *51*, 913–936. [CrossRef]

32. Shadloo, M.S.; Weiss, R.; Yildiz, M.; Dalrymple, R.A. Numerical simulation of long wave runup for breaking and nonbreaking waves. *Int. J. Offshore Polar Eng.* **2015**, *25*, 1–7.[CrossRef]
33. Bender, J.; Koschier, D.; Kugelstadt, T.; Weiler, M. A micropolar material model for turbulent sph fluids. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Los Angeles, CA, USA, 28–30 July 2017; p. 4. [CrossRef]
34. Hong, J.M.; Lee, H.Y.; Yoon, J.C.; Kim, C.H. Bubbles alive. *ACM Trans. Graph. (TOG) ACM* **2008**, *27*, 48. [CrossRef]
35. Ihmsen, M.; Bader, J.; Akinci, G.; Teschner, M. Animation of air bubbles with SPH. In Proceedings of the International Conference on Computer Graphics Theory and Applications, Vilamoura, Portugal, 5–7 March 2011. [CrossRef]
36. Chentanez, N.; Müller, M. Real-time Eulerian water simulation using a restricted tall cell grid. *ACM Trans. Graph. (TOG) ACM* **2011**, *30*, 82. [CrossRef]
37. Takahashi, T.; Fujii, H.; Kunimatsu, A.; Hiwada, K.; Saito, T.; Tanaka, K.; Ueki, H. Realistic animation of fluid with splash and foam. *Comput. Graph. Forum* **2003**, *22*, 391–400. [CrossRef]
38. Mihalef, V.; Metaxas, D.; Sussman, M. Simulation of two-phase flow with sub-scale droplet and bubble effects. *Comput. Graph. Forum* **2009**, *28*, 229–238. [CrossRef]
39. Meric, D.; Gedikli, H. Energy absorption behavior of tailor-welded tapered tubes under axial impact loading using coupled FEM/SPH method. *Thin-Walled Struct.* **2016**, *104*, 17–33. [CrossRef]
40. Zhang, Y.; Ban, X.; Wang, X.; Liu, X. A Symmetry Particle Method towards Implicit Non-Newtonian Fluids. *Symmetry* **2017**, *9*, 26. [CrossRef]
41. Gribbin, J. *Q is for Quantum: An Encyclopedia of Particle Physics*; Simon Schuster: New York, NY, USA, 2000.
42. Sethna, J. *Statistical Mechanics: Entropy, Order Parameters, and Complexity*; Oxford University Press: Oxford, UK, 2006.
43. Koks, D. *Explorations in Mathematical Physics: The Concepts Behind an Elegant Language*; Springer Science and Business Media: Berlin/Heidelberg, Germany, 2006.
44. Ihmsen, M.; Akinci, N.; Becker, M.; Teschner, M. A Parallel SPH Implementation on Multi-Core CPUs. *Comput. Graph. Forum* **2011**, *30*, 99–112. [CrossRef]
45. He, L.; Ban, X.; Liu, X.; Wang, X. Individual time stepping for SPH fluids. *Eurograph. Assoc.* **2015**, doi:10.2312/egsh.20151010. [CrossRef]
46. Akinci, N.; Ihmsen, M.; Akinci, G.; Solenthaler, B.; Teschner, M. Versatile rigid-fluid coupling for incompressible sph. *ACM Trans. Graph. (TOG)* **2012**, *31*, 62. [CrossRef]
47. Coumans, E. Bullet Physics Library. Available online: http://bulletphysics.org/wordpress/ (accessed on 2 February 2018).
48. Wang, X.; Ban, X.; Liu, X.; Zhang, Y.; Wang, L. Efficient extracting surfaces approach employing anisotropic kernels for SPH fluids. *J. Vis.* **2016**, *19*, 301–317. [CrossRef]
49. Kent, B.R. *3D Scientific Visualization with Blender*; Morgan and Claypool: San Rafael, CA, USA, 2015. [CrossRef]