

Article

# A Symmetry Particle Method towards Implicit Non-Newtonian Fluids

Yalan Zhang, Xiaojuan Ban\*, Xiaokun Wang, and Xing Liu

University of Science and Technology Beijing, Xueyuan Road 30, Haidian District, Beijing, China, 100083; b20150306@xs.ustb.edu.cn (Y.Z.); banxj@ustb.edu.cn (X.B.); wang1xiao2kun3@163.com (X.W.); ustbdante@gmail.com (X.L.)

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

Academic Editor: Yuhua Luo

Received: 17 Jan 2017; Accepted: 10 Feb 2017; Published: date

**Abstract:** In this paper, a symmetry particle method, the smoothed particle hydrodynamics (SPH) method, is extended to deal with non-Newtonian fluids. First, the viscous liquid is modeled by a non-Newtonian fluid flow and the variable viscosity under shear stress is determined by the Carreau-Yasuda model. Then a pressure correction method is proposed, by correcting density error with individual stiffness parameters for each particle, to ensure the incompressibility of fluid. Finally, an implicit method is used to improve efficiency and stability. It is found that the non-Newtonian behavior can be well displayed in all cases, and the proposed SPH algorithm is stable and efficient.

**Keywords:** symmetry; cooperation visualization; smoothed particle hydrodynamics; non-Newtonian fluid; implicit method

## 1. Introduction

Non-Newtonian fluid is widely presented in industry, mining engineering, food processing, and many other fields in cases such as pit filled paste, blood, honey, etc. However, the physical nature of non-Newtonian fluid is very complicated. It is pretty difficult to exactly simulate non-Newtonian fluid. A non-Newtonian fluid has characteristic properties of both solids and fluids [1]. Take the shear thinning non-Newtonian fluid for example, when it flows with low velocity, its viscosity becomes higher, and the fluid starts to exhibit properties of solids; while as it flows with high velocity, its viscosity gets smaller, and the non-Newtonian fluid exhibits liquid properties. In recent years, smoothed particle hydrodynamics [2] (SPH) has become an important Lagrangian method for industrial flow simulation [3] and computational fluid dynamics [4,5]. Because of the symmetry of SPH method with respect to translations, rotations, and time, the exact conservation of momentum, angular momentum and energy are guaranteed [6–9]. There are several difficulties to simulating non-Newtonian fluids. Firstly, how to model the non-Newtonian fluid. Due to the non-linear relationship between viscosity and shear force, the adequate physical equations and stable approximate simulation method should be re-established for non-Newtonian fluids. Secondly, how to ensure the incompressibility of non-Newtonian fluids. SPH method has some inherent problems which are the same as massless methods [10]. Enforcing incompressibility for particle methods is a challenging problem [11]. Thirdly, how to improve the efficiency of simulation. Due to the complicated physical equations of non-Newtonian fluids, a small time step is necessary, which proves to be rather time-consuming.

A novel non-Newtonian fluid simulation method for SPH is proposed in this article. The viscous liquid is modeled by a non-Newtonian fluid flow, and the variable viscosity under shear stress is achieved using a viscosity model known as the Carreau-Yasuda model. To improve numerical

stability, a pressure correction method is used to correct density error by setting up individual stiffness parameters for each particle. Furthermore, an implicit viscosity solver is proposed for non-Newtonian SPH fluid.

## 2. Related Works

There are two major approaches to simulate non-Newtonian fluid: Eulerian method and Lagrangian method. The Lagrangian method treats fluid as a particle system. Each point in the fluid is labeled as a separate particle, with a position and a velocity. Instead of tracking each particle, the Eulerian method can look at fixed points in space and see how the fluid quantities (such as density, velocity, etc.) measured at those points change in time [12].

For the Eulerian method, the main problems are how to model non-Newtonian fluid and how to deal with free surface. Goatskin et al. [13] proposed a quasi-linear plastic model to control the change of viscosity in the gradual transformation from a solid to a non-Newtonian fluid. This model meets the von Mises's yield condition. They simulated the melting of a piece of wax with this model by using explicit Euler method. Losasso et al. [14] modeled unified fluids with different viscoelasticities and densities by extending the particle level set. There are many different level sets in the same grid region. The exchanges between different level sets can be achieved by adding the pressure correction process before calculating viscosity. Batty and Bridson [15] found out that the Laplace operator and Strain rate items of non-Newtonian fluid should be computed under the velocity field with no divergence; therefore, the pressure correction process was added after the calculation of viscosity. Besides, they proposed an implied unconditionally stable simulation method based on marker-and-cell. Even though they proposed high precision free surface boundary conditions, it still cannot simulate high viscosity Newtonian fluid. Bergou et al. [16] proposed a method based on discrete differential geometry to simulate one dimensional linear viscous fluid. This approach can simulate one dimensional phenomenon vividly, but distortion often appeared when the flow layer got thicker. Batty and Uribe [17] simulated thin viscous layer with dimensionality deduce technology. They combined nonlinear surface tension and smallest-discrete-surface-area-based formulas and reserved the mass of unit triangular mesh with the local grid reconstruction technology. This approach can simulate the physical conditions of viscous thin layer, such as bending and draping.

Particle-based methods are superior when free surface and complex interface are to be dealt with. So mesh free methods represented by SPH method have been more commonly applied in the field of non-Newtonian fluid simulation [18, 19]. Müller et al. [20] simulated fluids of high elasticity and high plasticity by structuring a point-based animation modeling method. He combined continuum mechanics with von Mises' yield condition and calculated the displacement and the velocity using moving least squares. Afonso et al. [21] also completed simulations with a General Newtonian Fluid model, and the viscosity of fluid was controlled by the Jump Number. The melting process of a heated non-Newtonian fluid [22], which later becomes a low viscosity fluid, was also modeled by the general Newtonian fluid model. Rafiee et al. [23] proposed a free surface SPH method. This method can be applied to Newton fluid and viscoplastic fluid, and simulate the bending phenomena of viscous fluids. The relationship between the pressure and the viscosity of non-Newtonian fluids was described with the real polymer state equation [24]. Andrade et al. [25] can simulate the flow of polymer with high viscosity and high pressure. In addition, the unified molding of viscous Newtonian fluid and shear thinning non-Newtonian fluid were simulated, but this method can only simulate fluid with low viscosity; whereas unstable phenomenon would appear when the time step is larger. Zhang et al. [26] proposed an explicit method for shear thinning non-Newtonian fluid.

In addition to Euler and Lagrange methods, recently there is a novel alternative, the codimensional method [27], which is able to simulate a non-Newtonian fluid. This method enables the simulation of interaction between different dimensions of a non-Newtonian fluid, such as filamentous remaining marks left on a pool of paint after getting brushed over, which is difficult for Euler and Lagrange methods to achieve.

For Newtonian fluid simulation, the standard SPH has been gradually replaced by a predictive–corrective method to ensure incompressibility with larger time step. Predictive–corrective algorithm was first proposed by Solenthaler and Pajarola [28], which is to predict the state of a fluid in absence of pressure, then use pressure for its conduct correction. Ihmsen et al. [29] proposed an implicit SPH method that adjusts fluid’s density by pressure according to the relationship between density and velocity in the next time step. Bender and Koschier [30] proposed a divergence-free velocity method, which is essentially also a predictive–corrective method that prevents volume compression and enforces a divergence-free velocity field formulation of the physical laws. However, none of them involved non-Newtonian fluid simulation.

### 3. Material model

#### 3.1. Governing Equations

In the SPH scheme, the fluid is described as moving particles with physical attributes such as position  $\mathbf{X}$ , velocity  $\mathbf{V}$ , and mass  $m$ . Mass is used to determine the local geometry of the fluid and assumed to be constant. Non-Newtonian fluids are a continuous medium whose motion equations must follow the mass and momentum conservation principles. The motion equations are represented by simulating the advection of particles. Through the use of integral interpolant, the dependent field variables are expressed by integrals which are approximated by summation interpolant over neighboring particles.

Compared with Newtonian fluid [31], the strain tensor term should be added in non-Newtonian fluid. In the Lagrangian frame, the governing equations for the incompressible fluid with spatially constant viscosity can be written as the following

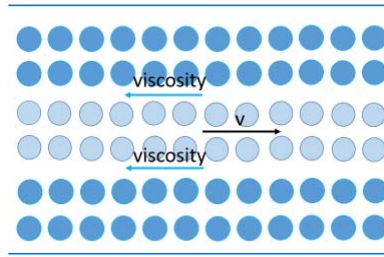
$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v} \quad (1)$$

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \boldsymbol{\sigma} + \frac{1}{m} \mathbf{F}^{ext} \quad (2)$$

where  $t$  denotes the time,  $\rho$  is the density,  $p$  is the pressure,  $\boldsymbol{\sigma}$  is the viscous stress tensor and  $\frac{1}{m} \mathbf{F}^{ext}$  denotes the acceleration due to external forces such as gravity, surface tension, and friction.

The Equation (1) is the continuity equation. Since the mass is carried by particles, the mass is conserved exactly. The Equation (2) is the momentum equation. The term  $-\frac{1}{\rho} \nabla p$  is related to particle acceleration due to pressure changes in the fluid. The term  $\frac{1}{\rho} \nabla \cdot \boldsymbol{\sigma}$  describes the viscous acceleration.

Viscosity, which is defined as the resistance of a fluid to flow, is essentially friction forces (Figure 1). For incompressible flow, the viscous stress tensor is  $\boldsymbol{\sigma} = \mu \boldsymbol{\gamma}$ , where  $\mu$  is the viscosity and  $\boldsymbol{\gamma} = \nabla \mathbf{v} + (\nabla \mathbf{v})^T$  is the strain rate tensor. The shear rate  $\gamma$  is approximated by using Frobenius norm  $\gamma = \sqrt{\boldsymbol{\gamma} : \boldsymbol{\gamma}}$  to model non-Newtonian fluids.



**Figure 1.** Viscous forces model. V: velocity.

### 3.2. Carreau-Yasuda Model

Both Newtonian and non-Newtonian fluid flows are considered in this paper, and the variable viscosity under shear stress is achieved using a rheological model known as Carreau-Yasuda model [32]. The relationship between shear rate and viscosity is defined by:

$$\mu(\gamma) = \mu_{\infty} + (\mu_0 - \mu_{\infty})(1 + (K\gamma)^{\alpha})^{\frac{n-1}{\alpha}} \quad (3)$$

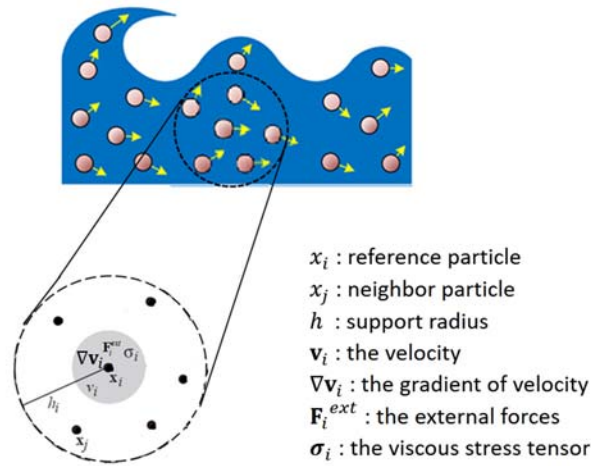
where  $K$  controls the shear rate and  $\alpha$  governs smoothness of transitions,  $\mu_0$  and  $\mu_{\infty}$  are given positive constants. When  $n=1$ , the fluid is a Newtonian fluid with viscosity  $\mu_0$ . When  $K=0$ , the kinematic viscosity is equivalent to  $\mu_0$ , which means under low shear rate, the viscosity of non-Newtonian fluid is governed by  $\mu_0$ . When  $n<1$ , the second term approaches zero and the viscosity approaches to  $\mu_{\infty}$  as the shear rate increases, the simulated fluid is shear thinning non-Newtonian fluid. When  $n>1$ , the viscosity increases with the amount of scaled by  $\mu_0 - \mu_{\infty}$  as the shear rate increases, and the simulated fluid is shear thickening non-Newtonian fluid.

### 3.3. Smoothed Particle Hydrodynamics

In the SPH approach, the basic idea is to discretize the fluid domain into a finite number of particles with physical attributes (Figure 2). A continuous field  $A(x_i)$  can be obtained by summation over all the sampling points  $x_j$  with mass  $m_j$  and density  $\rho_j$  in the support domain:

$$\langle A(x) \rangle = \sum_j \frac{m_j}{\rho_j} A_j W_{ij} \quad (4)$$

where  $W_{ij} = W(\mathbf{x} - \mathbf{x}_j, h)$  is the kernel function with  $h$  the particle radius, and  $j$  iterating all the neighbors. The chosen kernel function is limited by several conditions, such as normalization, compact condition, and delta function property. In this paper, the popular quantic spline kernel is used for all simulations.



**Figure 2.** Illustration for smoothed particle hydrodynamics (SPH) method. It is a symmetry particle method and the physical attributions of each particle is weighted in summation by its neighbors.

In the SPH interpolation scheme, the derivatives of field quantities only affect the smoothing kernel. The gradient of  $A$  can be calculated as:

$$\nabla A(\mathbf{x}) \approx \langle \nabla A(\mathbf{x}) \rangle = \sum_j \frac{m_j}{\rho_j} A_j \nabla W_{ij} \quad (5)$$

By substituting density into Equation (5), the density at location  $\mathbf{x}_i$  is:

$$\rho_i = \sum_j m_j W_{ij} \quad (6)$$

Generally, there are two formulas to relate pressure and density, namely the Poisson equation and the gas state equation. Solving the Poisson equation guarantees the incompressibility of fluid, but it is time-consuming. In contrast, the explicit calculation is adopted in an ideal gas equation, but results in the stiffness value having to be very large so that the density fluctuations will not exceed 1%. However, according to the Courant-Friedrichs-Lewy (CFL) condition, a large stiffness value will restrict the time step and make the simulation inefficient. In this paper, the equation of the gas state is used for efficiency and the problem of the stiffness value will be solved by pressure projection:

$$p_i = k(\rho_i - \rho_0) \quad (7)$$

where  $k$  is a gas constant that depends on the temperature, and  $\rho_0$  is the rest density. The value  $\rho_0 = 1000 \text{ kg/m}^3$  turns out to be suitable for all experiments.

To compute the pressure acceleration, substitute  $-\frac{1}{\rho} \nabla p$  into Equation (6):

$$-\frac{1}{\rho_i} \nabla p_i = -\sum_j m_j \frac{p_i + p_j}{\rho_j} \nabla W_{ij} \quad (8)$$

In order to compute the shear stress  $\sigma_i$  at particle  $i$ , the same SPH approximation as [33] is adopted for the deformation tensor  $\gamma_i = \nabla \mathbf{v}_i + (\nabla \mathbf{v}_i)^T$  with:

$$\nabla \mathbf{v}_i = \sum_j \frac{m_j}{\rho_j} \mathbf{v}_{ji} \otimes \nabla W_{ij} \quad (9)$$

where  $\mathbf{v}_{ji} = \mathbf{v}_i - \mathbf{v}_j$ . After updating shear stress at all particles, the viscous acceleration is approximated by:

$$\frac{1}{\rho} \nabla \cdot \boldsymbol{\sigma}_i = \sum_j m_j \left( \frac{\sigma_i}{\rho_i^2} + \frac{\sigma_j}{\rho_j^2} \right) \nabla W_{ij} \quad (10)$$

#### 4. Pressure Projection Algorithm

In the non-Newtonian fluid model proposed above, due to a rather high compressibility caused by the gas state equation, when a non-Newtonian fluid flows, the volume of the fluid might change [31]. In order to address this problem, a pressure projection algorithm is proposed to correct density by pressure.

Currently, there are many different pressure solvers that minimize the density error  $\rho - \rho_0$  [28,29]. The algorithm proposed in this paper is similar with [28]. Firstly, only external force  $\mathbf{F}^{ext}$  is considered to obtain the first intermediate velocity  $\mathbf{V}^*$ . Secondly, to enforce fluid incompressibility, a pressure projection algorithm is proposed in this section by using a new solver that sets a separate stiffness parameter  $k_i$  for each fluid particle  $i$  to satisfy local incompressibility, making the density of the entire fluid system remain unchanged and getting the second intermediate velocity  $\mathbf{V}^{**}$ . Next, an implicit method is proposed to solve the viscous stress tensor (see Section 5).

Finally, particle velocity and position are integrated using the Euler scheme, and the velocity can be rewritten as:  $\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \Delta t \frac{\mathbf{F}_i^{ext}(t) + \mathbf{F}_i^p(t) + \mathbf{F}_i^{vis}(t)}{m_i}$  with unknown pressure forces  $\mathbf{F}_i^p$  and viscosity forces  $\mathbf{F}_i^{vis}$  and known external forces  $\mathbf{F}_i^{ext}$  such as gravity and surface tension. Following the projection concept, intermediate (predicted) velocities are set to be  $\mathbf{V}_i^*$ :

$$\mathbf{v}_i^* = \mathbf{v}_i(t) + \Delta t \frac{\mathbf{F}_i^{adv}(t)}{m_i} \quad (11)$$

The second intermediate velocity  $\mathbf{V}^{**}$  is:

$$\mathbf{v}_i^{**} = \mathbf{v}_i^* + \Delta t \frac{\mathbf{F}_i^p(t)}{m_i} \quad (12)$$

The velocity in the next time step can be written as:

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i^{**} + \frac{\Delta t}{\rho} \nabla \cdot \boldsymbol{\sigma} \quad (13)$$

The pressure gradient is computed using the SPH formulation [29]:

$$\nabla p_i = k_i \nabla \rho_i = k_i \nabla \sum_j m_j \nabla W_{ij} \quad (14)$$

The pressure force of particle  $i$  caused by pressure acceleration is determined by:

$$\mathbf{F}_i^p = -m_i \frac{1}{\rho} \nabla p = -k_i \frac{m_i}{\rho_i} \sum_j m_j \nabla W_{ij} \quad (15)$$

$\mathbf{F}_{j \leftarrow i}^p$  is the pressure forces that act from particle  $i$  on the neighboring particles  $j$ . According to Newton's third law, its value is equal to the total pressure of all neighboring particles  $j$  affects  $i$ , which satisfies  $\mathbf{F}_i^p + \sum_j \mathbf{F}_{j \leftarrow i}^p = 0$ . Then we can get:

$$\mathbf{F}_{j \leftarrow i}^p = k_i \frac{m_i}{\rho_i} m_j \nabla W_{ij} \quad (16)$$

The velocity changes  $\Delta \mathbf{v}_i$  by pressure forces is:

$$\Delta \mathbf{v}_i = \Delta t \frac{\mathbf{F}_i^p(t)}{m_i} = -\Delta t \frac{k_i}{\rho_i} \sum_j m_j \nabla W_{ij} \quad (17)$$

Similarly:

$$\Delta \mathbf{v}_j = \Delta t \frac{\mathbf{F}_{j \leftarrow i}^p(t)}{m_j} = \Delta t \frac{k_i}{\rho_i} m_j \nabla W_{ij} \quad (18)$$

According to [28], the intermediate density caused by intermediate velocity is:

$$\rho_i^* = \rho_i(t) + \Delta t \sum_j m_j (\mathbf{v}_i^* - \mathbf{v}_j^*) \nabla W_{ij} \quad (19)$$

We now search for pressure forces to resolve the compression, i.e., the deviation from the rest density  $\Delta \rho_i$ :

$$\begin{aligned} \Delta \rho_i &= \Delta t \sum_j m_j (\Delta \mathbf{v}_i - \Delta \mathbf{v}_j) \nabla W_{ij} \\ &= \Delta t \sum_j m_j \left( -\Delta t \frac{k_i}{\rho_i} \sum_j m_j \nabla W_{ij} - \Delta t \frac{k_i}{\rho_i} m_j \nabla W_{ij} \right) \nabla W_{ij} \\ &= -\frac{k_i}{\rho_i} \Delta t^2 \left( \left( \sum_j m_j \nabla W_{ij} \right)^2 + \sum_j (m_j \nabla W_{ij})^2 \right) \\ &= \rho_0 - \rho_i^* \end{aligned} \quad (20)$$

Solving for  $k_i$  yields:

$$k_i = \frac{\rho_0 - \rho_i^*}{\Delta t^2} \alpha_i, \text{ with } \alpha_i = \frac{\rho_i}{\left( \left( \sum_j m_j \nabla W_{ij} \right)^2 + \sum_j (m_j \nabla W_{ij})^2 \right)} \quad (21)$$

---

#### Algorithm 1. Pressure Projection Algorithm

---

```

for all particles  $i$  do
  update  $\rho_i$  using Equation (7)
  update  $\alpha_i$  using Equation (21)
end for
for all particles  $i$  do
  update  $\mathbf{v}_i^*$  using Equation (12)
end for
while  $(\rho_{avg} - \rho_0 > \eta)$  or  $iter < 2$  do
  for all particles  $i$  do
    update  $\rho_i^*$  using Equation (20)
    update  $k_i$  using Equation (21)
  
```

---

---

```


$$\rho_i^* = \rho_i^* + \frac{k_i}{\alpha_i} \Delta t^2$$

end for
end while
for all particles  $i$  do

$$\mathbf{v}_i^{**} = \mathbf{v}_i^* - \Delta t \sum_j m_j \left( \frac{k_i}{\rho_i} + \frac{k_j}{\rho_j} \right) \nabla W_{ij}$$

end for

```

---

Algorithm 1 outlines the simulation using our novel method in detail. First, the particle neighborhoods  $N_i$  using compact hashing [34] are determined. Then for each particle, the density  $\rho_i$  and the factor  $\alpha_i$  are computed. Since this factor solely depends on the current positions, it is pre-computed before executing the solvers, which saves the solver from significant computational work. The third step in the simulation loop is to compute the strain tensor term of a non-Newtonian fluid, which is the main difference between a Newtonian fluid and a non-Newtonian fluid.

The forth and the fifth step of the algorithm are predictive-correction steps, whose purpose is to minimize the density error by the pressure solver, scilicet solving  $\rho - \rho_0 = f(p)$ . In this step, the non-pressure forces are used to compute predicted velocities  $\mathbf{V}_i^*$ . The constant density solver uses this prediction and the pre-computed factors  $\alpha_i$  to determine the pressure forces for each neighborhood in order to correct the density error.

Finally, the resulting velocity changes are used to update the velocity of the particle without viscosity force  $\mathbf{v}_i^{**}$ .

## 5. Implicit Viscosity Solver

The viscous acceleration is commonly computed from the divergence of viscous stress tensor, which is mentioned in Section 3.2. In this paper, the proposed solver first computes an intermediate velocity  $\mathbf{V}^*$  without considering pressure and viscosity. Then, the second updated intermediate velocity  $\mathbf{V}^{**}$  is computed by using the proposed pressure correction method, which is described in Section 4. Finally, the proposed viscosity solver computes the final velocity  $\mathbf{v}(t+\Delta t)$  and the position of particles. Algorithm 2 summarizes the proposed implicit non-Newtonian SPH fluid solver.

---

### Algorithm 2. Implicit non-Newtonian SPH fluid solver

---

```

for all particles  $i$  do
  find neighborhoods  $N_i$ 
  compute time step size  $\Delta t$ 
end for
for all particles  $i$  do
  compute the external force  $\mathbf{F}_i^{ext}$ 
end for
for all particles  $i$  do
    update velocity  $\mathbf{v}_i^* = \mathbf{v}_i(t) + \Delta t \frac{\mathbf{F}_i^{adv}(t)}{m_i}$ 
  end for
for all particles  $i$  do
    compute the external force  $\mathbf{F}_i^{ext}$ 

```

---

---

```

end for
for all particles  $i$  do
    update velocity  $\mathbf{v}_i^* = \mathbf{v}_i(t) + \Delta t \frac{\mathbf{F}_i^{adv}(t)}{m_i}$ 
end for
for all particles  $i$  do
    compute the pressure and pressure force
end for
for all particles  $i$  do
    update velocity  $\mathbf{v}_i^{**}$  // Pressure Projection
end for
for all particles  $i$  do
    solving viscosity system
end for
for all particles  $i$  do
         $\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i^{**} + \frac{\Delta t}{\rho} \nabla \cdot \boldsymbol{\sigma}$ 
         $\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \Delta t \cdot \mathbf{v}_i(t + \Delta t)$ 
    end for

```

---

According to Algorithm 2, the viscosity solver starts with the strain rate tensor to compute the final velocity. Substituting Equation (15) into the strain rate tensor  $\boldsymbol{\gamma}_i = \nabla \mathbf{v}_i + (\nabla \mathbf{v}_i)^T$ , and then getting the symmetric tensor for particle  $i$  as a six-dimensional vector yields:

$$\boldsymbol{\gamma} = \begin{pmatrix} \gamma_i^{xx} \\ \gamma_i^{yy} \\ \gamma_i^{zz} \\ \gamma_i^{xy} \\ \gamma_i^{xz} \\ \gamma_i^{yz} \end{pmatrix} = \frac{1}{\rho_i} \sum_j m_j \begin{pmatrix} 2\mathbf{v}_{ji}^x \nabla W_{ij}^x \\ 2\mathbf{v}_{ji}^y \nabla W_{ij}^y \\ 2\mathbf{v}_{ji}^z \nabla W_{ij}^z \\ \mathbf{v}_{ji}^x \nabla W_{ij}^y + \mathbf{v}_{ji}^y \nabla W_{ij}^x \\ \mathbf{v}_{ji}^x \nabla W_{ij}^z + \mathbf{v}_{ji}^z \nabla W_{ij}^x \\ \mathbf{v}_{ji}^y \nabla W_{ij}^z + \mathbf{v}_{ji}^z \nabla W_{ij}^y \end{pmatrix} \quad (22)$$

To solve the viscous acceleration  $\frac{1}{\rho} \nabla \cdot \boldsymbol{\sigma}_i$ , the gradient of viscous stress tensor should be computed:

$$\nabla \cdot \boldsymbol{\sigma}_i = \nabla \cdot \mu_i \boldsymbol{\gamma}_i = \nabla \cdot f(\boldsymbol{\gamma}_i) \boldsymbol{\gamma}_i \quad (23)$$

According to the Equation (4),  $\mu_i$  is the function of the strain rate tensor  $\boldsymbol{\gamma}_i$ , while  $K$ ,  $\alpha$ ,  $\mu_0$ , and  $\mu_\infty$  are given positive constants and will not influence the gradient computing. Thus, the gradient of viscous stress tensor can be written as:

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma}_i &= \nabla \cdot \mu_i \boldsymbol{\gamma}_i = \nabla \cdot K(\mu_0 - \mu_\infty) \boldsymbol{\gamma}_i^{n-1} \boldsymbol{\gamma}_i \\ &= \nabla \cdot K(\mu_0 - \mu_\infty) \boldsymbol{\gamma}_i^n \\ &= nK(\mu_0 - \mu_\infty) \boldsymbol{\gamma}_i^{n-1} \frac{\partial \boldsymbol{\gamma}_i}{\partial \mathbf{v}_i} \end{aligned} \quad (24)$$

Due to the formulation, the required derivatives of the strain rate tensor  $\boldsymbol{\gamma}_i$  with respect to the velocity  $\mathbf{v}_j$  are determined by the following matrix:

$$\frac{\partial \gamma_i}{\partial \mathbf{v}_j} = \frac{1}{2\rho_i} \begin{pmatrix} 2\nabla W_{ij}^x & 0 & 0 \\ 0 & 2\nabla W_{ij}^y & 0 \\ 0 & 0 & 2\nabla W_{ij}^z \\ \nabla W_{ij}^y & \nabla W_{ij}^x & 0 \\ \nabla W_{ij}^z & 0 & \nabla W_{ij}^x \\ 0 & \nabla W_{ij}^x & \nabla W_{ij}^z \end{pmatrix} \quad (25)$$

In the SPH scheme, the derivatives of the strain rate tensor  $\gamma_i$  with respect to the velocity  $\mathbf{v}_i$  can be written as:

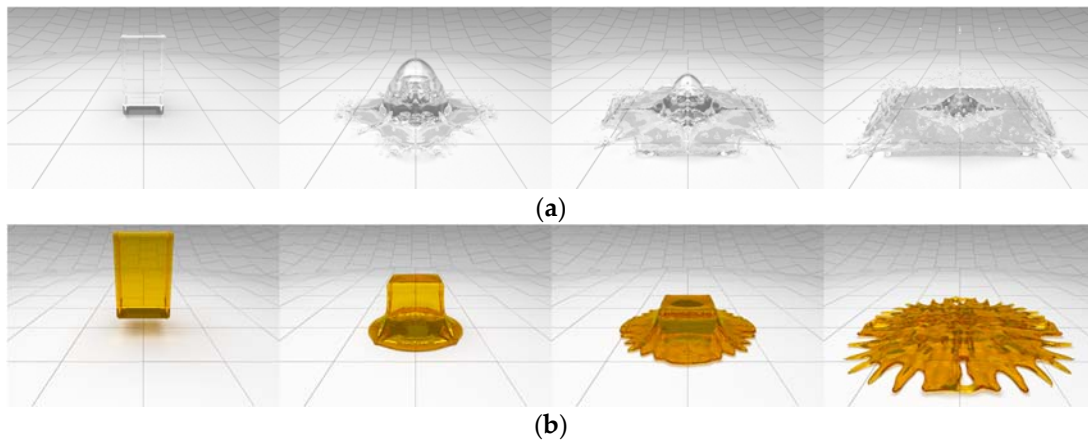
$$\frac{\partial \gamma_i}{\partial \mathbf{v}_i} = - \sum_j \frac{\partial \gamma_i}{\partial \mathbf{v}_j} \quad (26)$$

Now that all the variable quantities in Equation (14) can be solved, the final velocity and position of each particle are gained.

## 6. Implementation and Results

All timings are given for an Intel 3.50 GHz CPU with four cores. The simulation software is parallelized with Open MP. The simulation and surface reconstruction are actualized with C++ language and multi-threading technology. The density fluctuation was set to 0.01.

Figure 3 shows the Newtonian fluid and non-Newtonian fluid according to our method. As described in Section 3.2, when  $n=1$ , the non-Newtonian fluid model is simplified to a Newtonian fluid with constant kinematic viscosity (Figure 3a). When the fluid touches the container at the bottom of the scene (the container is set to be transparent for observers' convenience), the fluid particles that touch the container splash upward. In contrast, the non-Newtonian fluid particles (shown in Figure 3b) cannot splash when they collide with container due to their viscosity. The setting and statistics are illustrated in Table 1.

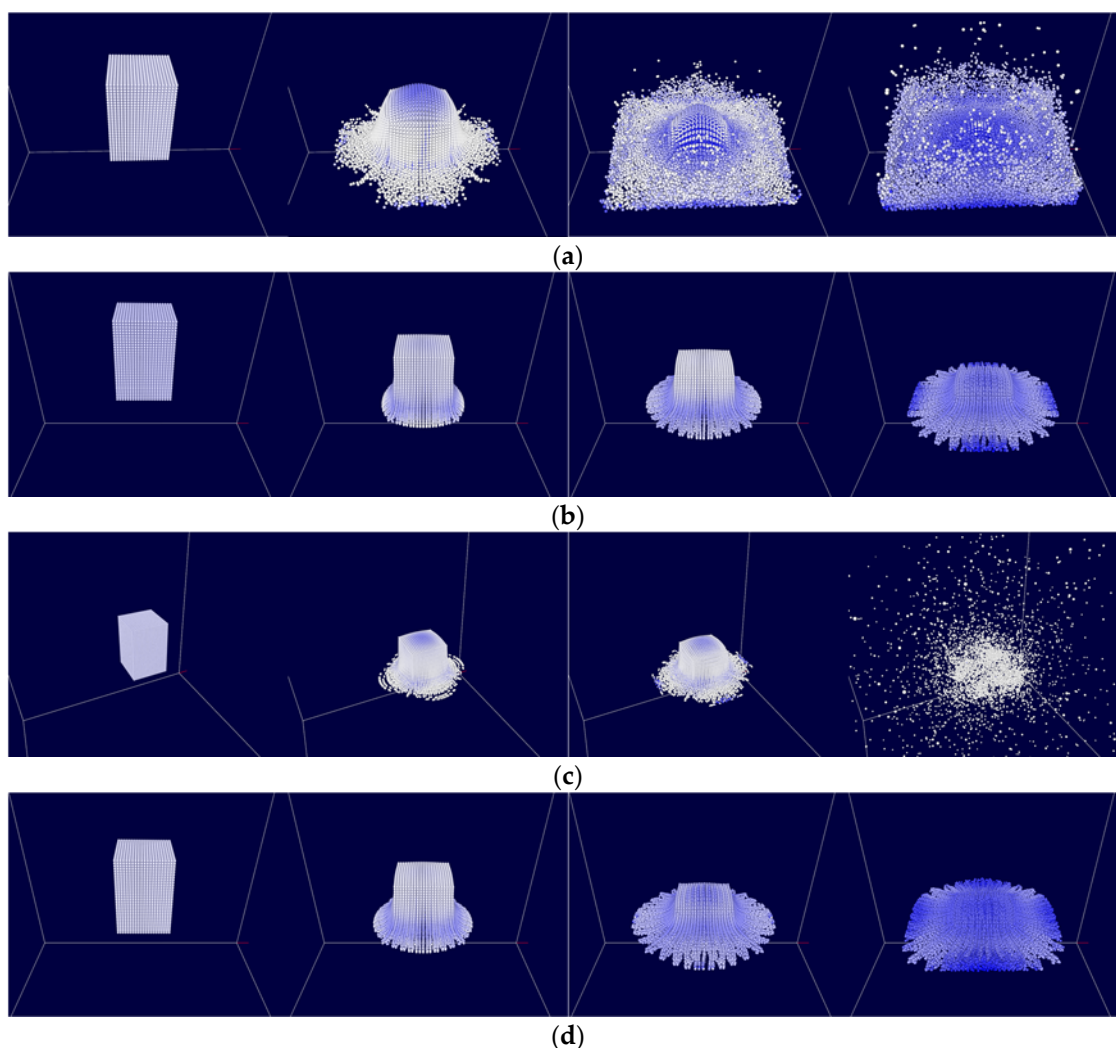


**Figure 3.** A falling water column with our method. (a) is Newtonian fluid; (b) is the non-Newtonian fluid.

**Table 1.** The simulation parameters for Figure 3. (Both Newtonian and non-Newtonian fluid).

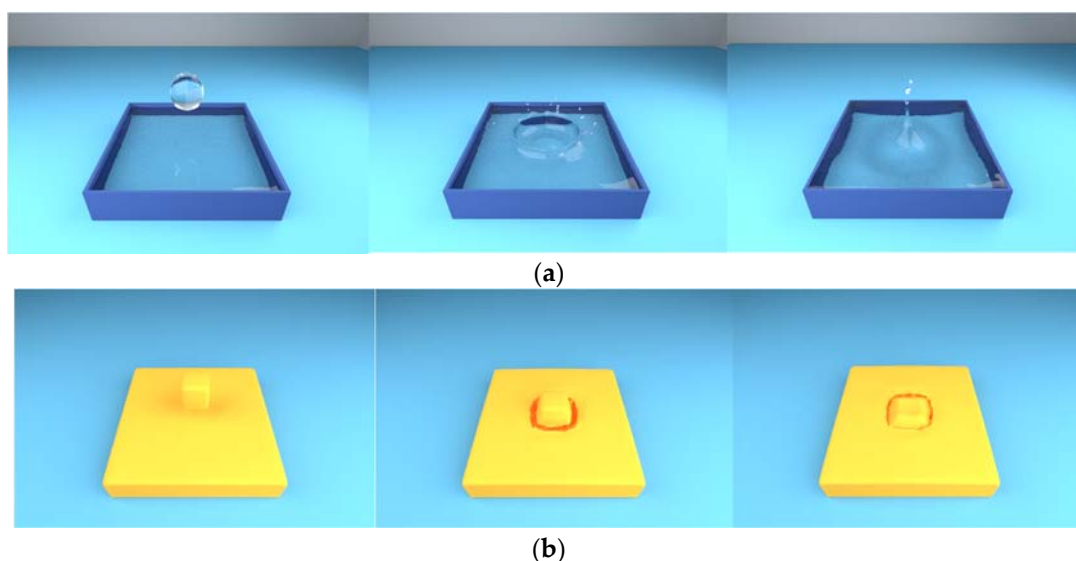
Simulation domain size	6 m × 6 m × 6 m
Fluid particles	13,671
Smoothing kernel function	cubic splines
Smoothing radius	0.1 m
Fluid particle width	0.05 m
Rendering time(per frame)	1.14 min

Figure 3 shows the free fall of a water column in particle state with 13,671 particles. In the same initial scene, both Newtonian fluid (Figure 4a) and non-Newtonian fluid (Figure 4b) could be achieved by governing the coefficients. We compared the non-Newtonian fluid simulation of our method with the method of Andrade et al. [25]. Andrade et al. used explicit integration without pressure projection. During the experiment, when the scene was small or the fluid particles moved gently, running the algorithm with pressure correction was found to be time and memory-consuming. However, when the scene is large or the fluid particles collide intensely, the method of Andrade et al. will be unstable with the larger time step. As shown in Figure 3c, when the time step was set to  $3.0 \times 10^{-4}$  s, fluid particles scattered in all directions with the method of Andrade et al. [25], while our algorithm still ran with the pressure projection (Figure 3d).



**Figure 4.** The free fall of a water column. (a) is Newtonian fluid (our method). (b) is the non-Newtonian fluid (Andrade et al.[25]) with  $\Delta t = 2.5 \times 10^{-6}$  s . (c) is the non-Newtonian fluid (Andrade et al.[25]) with  $\Delta t = 3.0 \times 10^{-4}$  s , (d) is the non-Newtonian fluid (our method) with  $\Delta t = 3.0 \times 10^{-4}$  s .

Figure 5 shows a water ball falling in a pool. The first line is Newtonian fluid (Figure 5a) and the second line is the non-Newtonian fluid (Figure 5b). As can be seen from the pictures, the wave of Newtonian fluid lashed against the water, sending up pearly spray. For the non-Newtonian fluid, there is no wave and the water sank into the pool directly. The setting and statistics are illustrated in Table 2.



**Figure 5.** A water block falling in a pool. (a) is Newtonian fluid. (b) is the non-Newtonian fluid.

**Table 2.** The simulation parameters for Figure 5. (Both Newtonian and non-Newtonian fluid)

Simulation domain size	8 m × 8 m × 8 m
Fluid particles	32,751
Smoothing kernel function	cubic splines
Smoothing radius	0.2 m
Fluid particle width	0.1 m
Rendering time(per frame)	3.85 min

At our test scene (Figure 4), we compared the simulation times and stability of our method with the method of Andrade et al. [25]. The physical parameters and performance measurements are summarized in Table 3. For the explicit method without pressure projection (Andrade et al. [25]), the time step is set according to the CFL condition [35] and the viscosity of fluid [25]. In the case of  $\eta = 0.01$ , the time step of our method could be running well with  $\Delta t = 3.0 \times 10^{-4}$  s, while the method of Andrade et al. [25] could not. As a result, for a two-second animation, our method needs 32 frames and 44.5 min for computing, while the method of Andrade et al. [25] needs 67 frames and 59.5 min for computing. The simulation parameters and performance of Figure 4 are also listed in Table 3.

**Table 3.** The simulation parameters and performance.  $\mu_0[m^2/s]$ ,  $\mu_\infty[m^2/s]$ ,  $K$ ,  $\alpha$  and  $n$ : fluid physical parameters,  $\Delta t[s]$ : time step,  $t^{total}[s]$ : simulation time per frame.

Figure	$\mu_0[m^2/s]$	$\mu_\infty[m^2/s]$	$K$	$\alpha$	$n$	$\Delta t[s]$	$t^{total}[s]$
4a	-	0.2	-	-	1	$3.0 \times 10^{-4}$	40.8
4b	2	0.2	1	0.5	-0.5	$2.5 \times 10^{-6}$	53.3
4c	2	0.2	1	0.5	-0.5	$3.0 \times 10^{-4}$	-
4d	2	0.2	1	0.5	-0.5	$3.0 \times 10^{-4}$	83.4
5a	-	0.2	-	-	1	$6.0 \times 10^{-4}$	178
5b	0.2	2	1	0.5	-0.5	$6.0 \times 10^{-4}$	216

## 7. Conclusions

In this paper, we proposed a novel symmetry particle-based method for non-Newtonian fluid, where the variable viscosity is decided by the Carreau-Yasuda Model. Our method includes a pressure correction step to reduce the density errors and ensure the incompressibility of fluid. An implicit method is used for the non-Newtonian fluid simulation to improve efficiency and stability. With our method, the fluid simulation runs well in large time steps compared with previous work.

One limitation of our method is that it does not consider temperature in our non-Newtonian fluid model. By taking this into fluid model, one could begin to simulate phenomena of solid-liquid conversion, like wax melt in high temperature. Additionally, work remains to be done on the topic of multi-phase coupling between different non-Newtonian fluids. This might be a potential direction for future work.

**Acknowledgments:** This work was supported by National Natural Science Foundation of China (No. 61572075) and The National Key Research and Development Program of China (Grant No.2016YFB0700502, 2016YFB1001404)

**Author Contributions:** Yalan Zhang conceived the framework and wrote the paper; Xing Liu 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. Ellero, M.; Tanner, R.I. SPH simulations of transient viscoelastic flows at low Reynolds number. *J. Non-Newton. Fluid Mech.* **2005**, *132*, 61–72.
2. 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, July 26–27 2003; Eurographics Association Press.: Aire-la-Ville, Switzerland, 2003; pp. 154–159.
3. Shadloo, M.S.; Oger, G.; Le Touzé, D. Smoothed particle hydrodynamics method for fluid flows, towards industrial applications: Motivations, current state, and challenges. *Comput. Fluids* **2016**, *136*, 11–34.
4. Sadek, S.H.; Yildiz, M. Modeling Die Swell of Second-Order Fluids Using Smoothed Particle Hydrodynamics. *J. Fluids Eng.* **2013**, *135*, 051103.
5. 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.
6. Price, D.J. Modelling discontinuities and Kelvin–Helmholtz instabilities in SPH. *J. Comput. Phys.* **2008**, *227*, 10040–10057.
7. Hayhurst, C.J.; Clegg, R.A. Cylindrically symmetric SPH simulations of hypervelocity impacts on thin plates. *Int. J. Impact Eng.* **1997**, *20*, 337–348.
8. Omang, M.; Børve, S.; Trulsen, J. Alternative kernel functions for Smoothed Particle Hydrodynamics in cylindrical symmetry. *Shock Waves* **2005**, *14*, 293–298.
9. Ren, J.; Ouyang, J.; Jiang, T.; Li, Q. A corrected symmetric SPH method to simulate viscoelastic free surface flows based on the PTT model. *Int. J. Numer. Methods Fluids* **2012**, *70*, 1494–1517.
10. Monaghan, J.J. Simulating free surface flows with SPH. *J. Comput. Phys.* **1994**, *110*, 399–406.
11. Shao, S.; Lo, E.Y.M. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Adv. Water Resour.* **2003**, *26*, 787–800.
12. Bridson, R.; Fedkiw, R.; Müller-Fischer, M. Fluid simulation: SIGGRAPH 2006 course notes Fedkiw and Muller-Fischer presentation videos are available from the citation page[C]//ACM SIGGRAPH 2006 Courses. *ACM* **2006**, 1–87, doi:10.1145/1185657.1185730.
13. Goktekin, T.G.; Bargteil, A.W.; O’Brien, J.F. A Method for Animating Viscoelastic Fluids. *ACM Trans. Graph.* **2004**, *23*, 461–466.
14. Losasso, F.; Shinar, T.; Selle, A.; Fedkiw, R. Multiple interacting liquids. *ACM Trans. Graph.* **2006**, *25*, 812–819.

15. Batty, C.; Bridson, R. Accurate Viscous Free Surfaces for Buckling, Coiling, and Rotating Liquids. In Proceedings of the 2008 Eurographics/ACM SIGGRAPH Symposium on Computer Animation, SCA 2008, Dublin, Ireland, 7–9 July 2008; pp. 219–228.
16. Bergou, M.; Audoly, B.; Vouga, E.; Wardetzky, M.; Grinspun, E. Discrete viscous threads. *ACM Trans. Graph.* **2010**, *29*, 157–166.
17. Batty, C.; Uribe, A.; Audoly, B.; Grinspun, E. Discrete Viscous Sheets. *ACM Trans. Graph.* **2012**, *31*, 13–15.
18. Xu, X.; Ouyang, J.; Yang, B.; Liu, Z. SPH simulations of three-dimensional non-Newtonian free surface flows. *Comput. Methods Appl. Mech. Eng.* **2013**, *256*, 101–116.
19. Pan, W.; Tartakovsky, A.M.; Monaghan, J.J. Smoothed particle hydrodynamics non-Newtonian model for ice-sheet and ice-shelf dynamics. *J. Comput. Phys.* **2013**, *242*, 828–842.
20. Müller, M.; Keiser, R.; Nealen, A. et al. Point based animation of elastic, plastic and melting objects. In Proceedings of the ACM Siggraph/eurographics Symposium on Computer Animation, Grenoble, France, 27–29 August 2004; Eurographics Association: Aire-la-Ville, Switzerland, 2004; pp. 141–151.
21. Afonso, P.; Fabiano, P.; Thomas, L.; Geovan, T. Particle-based viscoplastic fluid/solid simulation. *Comput.-Aided Des.* **2009**, *41*, 306–314.
22. Paiva, A.; Petronetto, F.; Lewiner, T.; Tavares, G. Particle-based non-Newtonian fluid animation for melting objects. In Proceedings of the 2006 19th Brazilian Symposium on Computer Graphics and Image Processing, Manaus, Brazil, 8–11 October 2006; pp. 78–85.
23. Rafiee, A.; Manzari, M.T.; Hosseini, M. An incompressible SPH method for simulation of unsteady viscoelastic free-surface flows. *Int. J. Non-Linear Mech.* **2007**, *42*, 1210–1223.
24. Fan X, J.; Tanner R, I.; Zheng, R. Smoothed particle hydrodynamics simulation of non-Newtonian moulding flow. *J. Non-Newton. Fluid Mech.* **2010**, *165*, 219–226.
25. De Andrade, L.F.S.; Sandim, M.; Petronetto, F.; Pagliosa, P.; Paiva, A. SPH Fluids for Viscous Jet Buckling. In Proceedings of the Graphics, Patterns and Images, Rio de Janeiro, Brazil, 26–30 August 2014; pp. 65–72.
26. Zhang, Y.; Ban, X.; Wang, X. et al. A Density-Correction Method for Particle-Based Non-Newtonian Fluid. In *Cooperative Design, Visualization, and Engineering*. Springer; Heidelberg, Germany, 2016.
27. Zhu, B.; Lee, M.; Quigley, E.; Fedkiw, R. Codimensional non-Newtonian fluids. *ACM Trans. Graph.* **2015**, *34*, 1–9.
28. Solenthaler, B.; Pajarola, R. Predictive-corrective incompressible SPH. *ACM Trans. Graph.* **2009**, *28*, 341–352.
29. Ihmsen, M.; Cornelis, J.; Solenthaler, B. et al. Implicit Incompressible SPH. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 426–435.
30. Bender, J.; Koschier, D. Divergence-free smoothed particle hydrodynamics[C]//Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation. *Eurographics Association Press, Aire-la-Ville, Switzerland*. **2015**, 147–155.
31. Becker, M.; Teschner, M. Weakly compressible SPH for free surface flows. In Proceedings of the ACM Siggraph/eurographics Symposium on Computer Animation, SCA 2007, San Diego, CA, USA, 2–4 August 2007; pp. 209–217.
32. Macosko, C.W.; Larson, R.G. *Rheology: Principles, Measurements, and Applications*; Wiley, New York, NY, USA, 1994.
33. Morris, J.P.; Fox, P.J.; Zhu, Y. Modeling low Reynolds number incompressible flows using SPH. *J. Comput. Phys.* **1997**, *136*, 214–226.
34. Ihmsen, M.; Akinci, N.; Becker, M.; Matthias, T. A Parallel SPH Implementation on Multi-Core CPUs. *Comput. Graph. Forum* **2011**, *30*, 99–112.
35. Monaghan, J.J. Smoothed particle hydrodynamics. *Annu. Rev. Astron. Astrophys.* **1992**, *30*, 543–574.

