

Multiphase Viscoelastic Non-Newtonian Fluid Simulation

paper1038

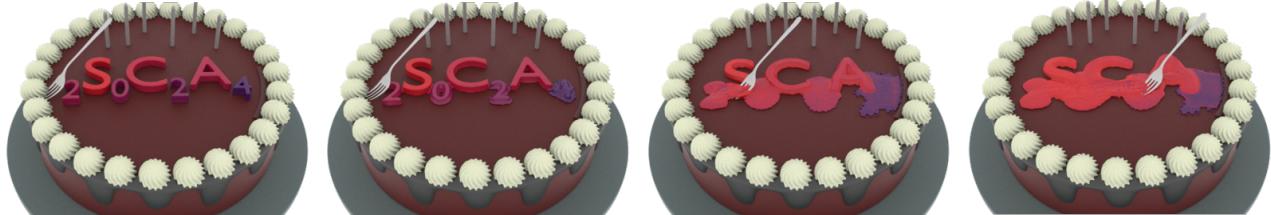


Figure 1: Seven different non-Newtonian fluids are used in one single simulation (2.06M particles in total). The letters in S-C-A come with decreasing shear thickening, whereas the numbers in 2-0-2-4 use gradually enhanced shear thinning (see Section 6 for details). Moving a fork across the seven characters shows different flow behaviors and dynamic interactions after fluid contact.

Abstract

We propose a method for simulating viscoelastic non-Newtonian fluids within a multiphase framework. For this, we use mixture models to handle component transport and conformation tensor methods to handle the fluid's viscoelastic stresses. In addition, we consider a bonding effects network to handle the impact of microscopic chemical bonds on phase transport. Our method supports the simulation of both steady-state viscoelastic fluids and discontinuous shear behavior. Compared to previous work on single-phase viscous non-Newtonian fluids, our method can capture more complex behavior, including material mixing processes that generate non-Newtonian fluids. We adopt a uniform set of variables to describe shear thinning, shear thickening, and ordinary Newtonian fluids while automatically calculating local rheology in inhomogeneous solutions. In addition, our method can simulate large viscosity ranges under explicit integration schemes, which typically requires implicit viscosity solvers under earlier single-phase frameworks.

Keywords: fluid simulation, mixture models, Viscoelastic Non-Newtonian fluids

CCS Concepts

• Computing methodologies → Physical simulation;

1. Introduction

Fluid simulation has always been a research hotspot in computer graphics [WXL^{*}24]. Many techniques have been proposed, including Newtonian fluid simulation, detail enhancement such as turbulence [FLX^{*}22], and multiphase fluid simulation [XWW^{*}23]. In Newtonian fluid simulations, previous work has covered surface tension [JWL^{*}23], pressure solvers [BK15], and viscous solvers [TDF^{*}15, WKBB18, LHW^{*}21, BK16] [PICT15]; as well as numerical accuracy improvements for different calculation methods [HKA21]. In multiphase fluid simulations, simulated phenomena include bubbles [HK03, IBAT11, TFK^{*}03], melting [MKN^{*}04, CBL^{*}09, FM07], and crystallization [KL03, KHL04, KAL06], among several others.

Newtonian fluids can be considered time-invariant with constant viscosity under certain conditions. In contrast, non-Newtonian fluids show a nonlinear relationship between viscosity and shear strain rate. Non-Newtonian fluids are actually more common in industrial

production and in nature, e.g., blood, chocolate, paint, and glass melts. At present, single-phase techniques can simulate simple non-Newtonian fluids [LHW^{*}21, ASP^{*}14] by computing the fluid's rheology by adding macroscopic viscosity models such as the Cross model [Cro65] to existing fluid simulators. For example, when configuring a specific fluid, such as the shear thinning type, the parameters of the Cross model are fixed in one single simulation. As the local rheological properties of a mixture may be different due to the solute concentration, such methods cannot simulate complex mixtures. Rather, when substances are mixed to form a non-Newtonian fluid (and since almost all non-Newtonian fluids are mixtures), simulation mechanisms with a multiphase framework are preferred.

This paper proposes a non-Newtonian fluid simulation method based on multiphase flow. Our method uses the conformation tensor method of non-Newtonian fluid dynamics to calculate the fluid's viscoelastic stress, and designs a non-Newtonian fluid mechanism based on the mixture model, which can simulate shear thinning flu-

ids, shear thickening fluids, and Newtonian fluids in a single framework. Our method can process single-phase non-Newtonian fluids, deal with dynamic mixing scenes, and compute the local rheological properties of the fluid separately. Our method has simple-to-set parameters and can be computed efficiently. Separately, in previous multiphase fluid models, interphase momentum exchange depends on suitably setting various constants (*e.g.* k_d in [JL21]; and C_d in [XWW^{*}23]). Such settings imply a *constant* degree of the mixture effect on phase motion for an entire simulation – which does not hold for systems with significant combining reactions. To the best of our knowledge, this is the first action model to effectively alleviate the limitations of previous multiphase fluid models where interphase momentum exchange is constrained by constant settings. Our model uniquely calculates the effect of concentration on phase transfer by simulating the dynamic network of chemical bonds formed in the solution, thus providing a more accurate representation of the mixing dynamics in non-Newtonian fluids.

2. Related Work

Much previous work on fluid simulation focuses on Newtonian fluids. Simulating non-Newtonian fluids is mainly achieved by viscosity solvers combined with simple non-Newtonian fluid viscosity models. In such schemes, a core problem is how to solve the viscous term in the Navier-Stokes equations and the viscosity of non-Newtonian fluids.

Viscosity computation: Currently, there are three main methods to solve the viscosity term in the SPH method: artificial viscosity (XSPH), viscosity Laplacian-based methods, and strain rate tensor-based methods. Schechter *et al.* [SB12] used XSPH to compute viscosity, which updates the velocity field by controlling the velocity difference with artificial parameters. However, such artificial parameters lead to the problem that phenomena such as curling and coiling cannot be correctly simulated. Many methods exist that handle the Laplacian of viscosity. A simple approach is to discretize the Laplacian by standard SPH. While easy to compute, this method is highly sensitive to the lack of particles at the boundary, and it cannot guarantee the sign correctness of the second derivative of the Gaussian-like kernel function. To avoid second derivatives, Takahashi *et al.* [TDF^{*}15] solved the first derivative twice, but the computational overhead increased. Brookshaw *et al.* [Bro85] used an SPH discretization combined with finite differences to process the Laplacian and conserve the linear and angular momenta. This method was later adopted by subsequent schemes [WKBB18, LHWW21].

Improving quality and speed: To achieve faster and stabler simulation of highly viscous fluids, many methods proposed implicit viscosity solvers based on strain rate tensors. Takahashi *et al.* [TDF^{*}15] used implicit time integration in a reverse Euler framework based on the original strain rate tensor representation. Though this scheme allowed for larger time steps, it led to pseudo-realistic artifacts in some scenes due to missing particles at the boundary. Bender *et al.* [BK16] proposed a velocity constraint function with the strain rate tensor to correct the velocity field, which is, however, expensive to compute. Peer *et al.* [PICT15] decomposed the velocity gradient tensor into three parts, with an ar-

tificial regulator imposed on the shear rate tensor; SPH discretization and Taylor expansion were used to numerically approximate the velocity field. However, this scheme cannot smoothly reconstruct the velocity field for low-viscous fluids. Also, this velocity field reconstruction method produces artifacts due to the above-mentioned missing particle issue. The Laplacian-based optimization scheme in [WKBB18] addressed the issue of boundary artifacts of the above methods.

To further improve visual realism in large viscous fluid simulations, some researchers have studied the interference problem between incompressibility and viscosity solutions. Liu *et al.* [LHWW21] first implemented a convergent sticky-incompressibility iterative algorithm in the SPH method by using the SIMPLE algorithm in an Eulerian setting, which reduces the interference between solvers; however, the iterative nature of this method can yield high computational overhead. Based on such SPH solvers, some studies further extended simulations to non-Newtonian fluids. Andrade *et al.* [ASP^{*}14] combined the symmetric form of Laplacian discretization with a cross-viscosity model to simulate simple shear-thinning non-Newtonian fluids. Liu *et al.* [LHWW21] used the Cross model to describe simple shear thinning and shear thickening scenarios.

Refined frameworks: Some viscous solvers achieve good visual results using more refined frameworks. The codimension method [ZLQF15] used the Carreau model to describe shear thinning and shear thickening fluids, achieving a good thin film effect; yet, this method involves complex calculations. Barreiro *et al.* [BGFAO17] were the first to implement real-time simulation of viscoelastic Newtonian fluids using polymer conformation methods combined with Position Based Dynamics (PBD); however, they did not consider non-Newtonian fluids. Goldade *et al.* [GWAB19] used an octree adaptive variational finite difference method to optimize the viscosity solution problem, which greatly improved the method's speed. Fei *et al.* [FBGZ19] developed a multi-scale framework for coupling wire and shear-related fluids, where the elastoplastic fluid was modeled by the Herscher-Barkley model. Panuelos *et al.* [PGG^{*}23] used the functional method to solve the unsteady Navier-Stokes equations to simulate highly viscoelastic fluids. The advantage of this method is that it preserves the coupling between the viscosity and pressure solutions and improves numerical accuracy. Su *et al.* [SXH^{*}21] used the MPM method to construct a second-order accurate viscosity solver that achieves diverse single-fluid effects while maintaining good numerical accuracy. By integrating previously simulated models of different materials into a unified particle framework, Li *et al.* [LGH^{*}23] created a comprehensive engine able to simulate a wide variety of non-Newtonian fluids.

Multiphase simulations: The above mentioned methods use a single-phase framework and cannot deal with multivariate dynamic scenarios, such as material mixing and precipitation phenomena. For such cases, multiphase simulations are needed. Multiphase simulation frameworks aim to address this case, which is consistent with the fact that fluids almost always exist as mixtures. Moreover, single-phase models can only simulate fluids with fixed properties; and non-Newtonian viscosity models can only realize simple non-Newtonian fluid simulations. The formation of non-Newtonian flu-

ids after material mixing and the simulation of the mixing process can only be achieved using a multiphase framework.

The mixture model, introduced by Ren *et al.* [RLY*14] to graphics, is a simple to compute and easy to extend multiphase model. Jiang *et al.* [JL21] further reformulated the computation of the hybrid model, abandoned the local equilibrium assumption, and enhanced simulation efficiency and stability, allowing the simulation of more dynamic scenes. Xu *et al.* [XWW*23] proposed an implicit mixture model and used a volume-incompressible method to enforce fluid incompressibility, leading to a more stable phase transport and a highly improved numerical stability.

The phenomena investigated by current multiphase fluid simulations are very broad, including bubble simulation [HK03, IBAT11, TFK*03], melting [MKN*04, CBL*09, FM07], and crystal formation [KL03, KHL04, KAL06]. However, in the field of computer graphics, there is no current approach that attempts to model non-Newtonian fluids with multiphase models. We present a method that enables efficient computation of dynamically non-Newtonian fluid scenarios under a multiphase model.

3. Implicit Mixture Model

The *Mixture model* [MnTK96] is a well-known technique to simulate multiphase fluids. It uses the volume fraction to represent the concentration ratio of each phase in a multiphase particle; computes the dynamic parameters of each phase through the multiphase fluid dynamics; and finally reconstructs the velocity field of the mixed particle. We now detail the mixture model's main terms.

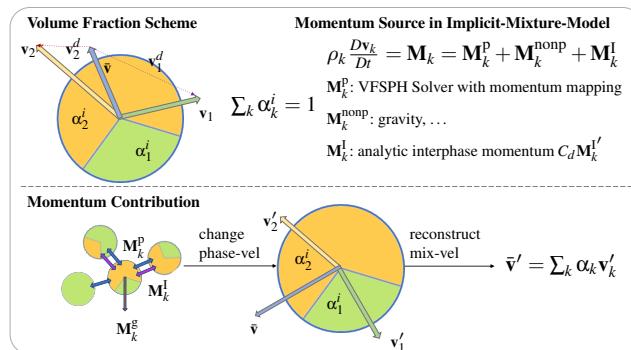


Figure 2: The volume fraction scheme and momentum sources in the implicit mixture model [XWW*23].

A mixed particle i with phases $k \in [1, \dots, K]$ contains the volume fractions α_k^i of each phase which sum up to 1, that is

$$\alpha^i = \sum_k \alpha_k^i = 1. \quad (1)$$

The mixture model uses the drift velocity v_k^d to measure the shift of the phase velocity v_k relative to the mixture velocity \bar{v} as

$$v_k^d = v_k - \bar{v}, \quad (2)$$

where

$$\bar{v} = \sum_k \alpha_k v_k. \quad (3)$$

Finally, the particle rest density $\bar{\rho}$ is computed as

$$\bar{\rho} = \sum_k \alpha_k \rho_k^0 \quad (4)$$

where ρ_k^0 is the rest density of phase k .

The recent implicit mixture model in [XWW*23] is designed to improve the numerical stability of mixture models. It uses a constant rest volume of particles and adopts the pressure profile $p_k = \bar{p}$ for mixture fluids. The basic implicit model has five phase-level momentum sources: pressure \mathbf{M}_k^p , viscosity \mathbf{M}_k^v , gravity \mathbf{M}_k^g , advection \mathbf{M}_k^a , and interphase \mathbf{M}_k^I ; the corresponding velocity fields and mixture momenta are denoted accordingly, e.g., v_k^v and \bar{v}^v for \mathbf{M}_k^v . In detail: \mathbf{M}_k^a is caused by the phase drift velocity; \mathbf{M}_k^I obeys a strict force balance model $\sum_k \mathbf{M}_k^I = 0$; according to Eqn. 2, we get $\bar{M}^* = \mathbf{M}_k^* - \mathbf{M}_k^d$ with $* \in \{p, v, g, a, I\}$; finally, the mixture momentum sources can be obtained by summing up all momentum sources of all phases as

$$\alpha_k \rho_k^0 \frac{Dv_k}{Dt} = \mathbf{M}_k^p + \mathbf{M}_k^v + \mathbf{M}_k^g + \mathbf{M}_k^a + \mathbf{M}_k^I, \quad (5)$$

$$\bar{\rho} \frac{D\bar{v}}{Dt} = \bar{\mathbf{M}}^p + \bar{\mathbf{M}}^v + \bar{\mathbf{M}}^g + \bar{\mathbf{M}}^a + \bar{\mathbf{M}}^I. \quad (6)$$

From Eqns. 2, 5, and 6, we get the change rate of phase drift velocity as

$$\frac{Dv_k^d}{Dt} = \frac{\mathbf{M}_k^*}{\alpha_k \rho_k^0} - \frac{\bar{\mathbf{M}}^*}{\bar{\rho}}. \quad (7)$$

The relationship between \mathbf{M}_k^I and \mathbf{M}_k^d can be deduced from the no-drift and free-drift situations to finally get $\mathbf{M}_k^I = C_d \mathbf{M}_k^I'$, where \mathbf{M}_k^I is the number interphase momentum and $C_d \in [0, 1]$ is a regulatory factor: $C_d = 0$ recovers free-drift, which means that the interphase momentum is zero, and $C_d = 1$ recovers no-drift, which means that the phase drift velocity is zero.

Computing the momentum sources v_k^v and v_k^p uses C_d and the corresponding momentum sources, as

$$\frac{Dv_k^v}{Dt} = C_d \frac{\bar{\mathbf{M}}^v}{\bar{\rho}} + (1 - C_d) \frac{\mathbf{M}_k^v}{\alpha_k \rho_k^0}, \quad (8)$$

$$\frac{Dv_k^p}{Dt} = C_d \frac{\bar{\mathbf{M}}^p}{\bar{\rho}} + (1 - C_d) \frac{\mathbf{M}_k^p}{\alpha_k \rho_k^0} = \frac{\bar{\mathbf{M}}^p}{\bar{\rho}} \left(C_d + (1 - C_d) \frac{\bar{\rho}}{\rho_k^0} \right). \quad (9)$$

Advection and drift momentum sources are reasonably ignored in the calculation, and pressure profile $p_k = \bar{p}$ is adopted, p_k is the pressure of phase k and \bar{p} is the mixture pressure. These makes the pressure treatment revert to the single-phase solver. A detailed derivation of momentum sources can be found in [XWW*23].

The phase transport is affected by two factors, namely drift velocity and diffusion, given by

$$\frac{D\alpha_k^i}{Dt} = - \sum_j V_0 (\alpha_k^i v_k^{i,d} + \alpha_k^j v_k^{j,d}) \cdot \nabla W^{ij}, \quad (10)$$

$$\nabla^2 \alpha_k^i = C_f \sum_j (\alpha_k^i - \alpha_k^j) \frac{\mathbf{x}^{ij} \cdot \nabla W^{ij}}{\|\mathbf{x}^{ij}\|^2 + 0.01h^2}, \quad (11)$$

where $\mathbf{v}_k^{i,d}$ is \mathbf{v}_k^d of particle i ; C_f is the diffusion coefficient; $\mathbf{x}^{ij} = \mathbf{x}^i - \mathbf{x}^j$ is the distance between the positions of particles i and j ; h is the SPH support radius; and W is the kernel function, which is common to be cubic spline function.

Volume incompressible SPH changes the density estimation step in DFSPH to a volume compression ratio estimation, and corrects the velocity field according to compressibility. This has the advantage of supporting the simulation of fluids with non-uniform density fields. The volume compression ratio is computed as

$$\phi^i = V^0 \sum_j W^{ij} \quad (12)$$

where V^0 is the rest volume of a particle and j denotes all neighbors of particle i within the kernel radius. Similar to DFPSH, ϕ is iteratively corrected to ensure that $\phi \leq 1$.

4. Proposed Method

In view of the shortcomings of existing single-phase solvers and the mixture model, we propose a mixture-model-based non-Newtonian fluid simulation method to simulate dynamic and highly viscoelastic fluid scenarios. The viscoelastic stress calculation is presented in Sec. 4.1, the extended non-Newtonian simulation mechanism in Sec. 4.2, and the dynamic phase transport control in Sec. 4.3.

4.1. Conformation Tensor Based Viscoelastic Stress

Polymer fluids provide one way for modeling non-Newtonian fluids. The polymer provides viscoelasticity to the fluid, and its concentration affects the lubrication of the fluid on the polymer and the friction between polymers, leading to an overall viscoelastic effect. In non-Newtonian fluid dynamics, the conformation tensor [BTRD11] can be exploited to control viscoelastic stress of polymer fluids.

The polymer fluid model defines the polymer stress tensor, which is the viscoelastic stress of the fluid, as

$$\sigma = c\eta_s \mathbf{s}(\mathbf{C}), \quad (13)$$

where η_s is the mixture viscosity; c is the mixture concentration. In the mixture model, $c = \alpha_2$, and the symmetric positive definite polymer conformation tensor \mathbf{C} evolves as

$$\frac{D\mathbf{C}}{Dt} = \mathbf{C}\nabla\mathbf{v} + (\nabla\mathbf{v})^T \mathbf{C} - \frac{1}{\lambda} \mathbf{s}(\mathbf{C}). \quad (14)$$

where λ is relaxation time. This model can only describe Newtonian viscoelastic fluids. To simulate non-Newtonian fluids, a non-linear term $\beta(\mathbf{C} - \mathbf{I})\mathbf{C}$ must be added, yielding

$$\frac{D\mathbf{C}}{Dt} = \mathbf{C}\nabla\mathbf{v} + (\nabla\mathbf{v})^T \mathbf{C} - \frac{1}{\lambda} \mathbf{s}(\mathbf{C}) - \beta(\mathbf{C} - \mathbf{I})\mathbf{C}, \quad (15)$$

where $\beta \in [0, 1]$ is a thinning factor that regulates the fluid's shear thinning capacity (larger β yields stronger shear thinning).

The tensor $\mathbf{s}(\mathbf{C})$ takes different forms in various non-Newtonian models. In our work, we use the Oldroyd-B case given by

$$\mathbf{s}(\mathbf{C}) = \mathbf{C} - \mathbf{I}. \quad (16)$$

The mixture viscoelastic stress is calculated as

$$\bar{\sigma} = c\eta_s(\mathbf{C} - \mathbf{I}), \quad (17)$$

In the multiphase framework, the velocity field of the mixture is reconstructed by the phase velocity. We use the implicit mixture model (Sec. 3) as a base model and replace the viscosity momentum source \mathbf{M}^v with the viscoelastic stress force momentum \mathbf{M}^{vise}

$$\mathbf{M}_k = \mathbf{M}_k^p + \mathbf{M}_k^{vise} + \mathbf{M}_k^g + \mathbf{M}_k^a + \mathbf{M}_k^l. \quad (18)$$

Following the mixture theory, we have that

$$\bar{\mathbf{M}}^{vise} = \sum_k \mathbf{M}_k^{vise}. \quad (19)$$

In our work, we only focus on two-phase fluids made up of water (α_1) and polymers ($\alpha_2 = 1 - \alpha_1$). Since the conformation tensor method focuses on the mixture as a whole, the momentum updates of each phase are not explicitly given. So, when computing momenta, we do something similar to the pressure profile:

$$\sigma_k = \bar{\sigma} \quad (20)$$

where σ_k is the stress of phase k , $\bar{\sigma}$ is the mixture stress. And according to Eqn. 19, we get

$$\mathbf{M}_k^{vise} = \alpha_k \bar{\mathbf{M}}^{vise}, \quad (21)$$

$$\bar{\mathbf{M}}^{vise} = \nabla \cdot \bar{\sigma}. \quad (22)$$

Finally, we obtain the viscoelastic phase momentum update equation

$$\frac{D\mathbf{v}_k^{vise}}{Dt} = \frac{\bar{\mathbf{M}}^{vise}}{\bar{\rho}} \left(C_d + (1 - C_d) \frac{\bar{\rho}}{\rho_k^0} \right). \quad (23)$$

This setting does not lead to the disappearance of the multiphase characteristics, and it ensures the normal phase transport and viscoelastic properties calculation.

The relaxation time λ in Eqn. 15 can affect the fluid's viscoelastic behavior. As λ approaches infinity, the mixture will approach an elastic solid; as λ approaches 0, the mixture will approach a pure viscous fluid. The solution's viscosity will affect the overall viscous behavior of the mixture, which differs from the general viscosity.

Another benefit of using a multiphase framework is that the fluid's local rheological properties can be computed independently. This can simulate the discontinuous shear scenario of a fluid, which is not possible with previous models (see Figure 3).

4.2. Non-Newtonian Mechanism

In the conformation tensor model (Sec. 4.1) there are already non-linear terms that can describe the fluid's shear thinning ability. Still, the model's descriptive ability is limited to Newtonian and shear thinning. We add shear thickening capability to the model by creating a non-Newtonian mechanism, combined with the mixture model, to dynamically calculate local fluid properties.

For the three fluid scenarios described in Figure 3, we know that the key parameter affecting the local rheological properties is the solution concentration, which can be conveniently replaced by the

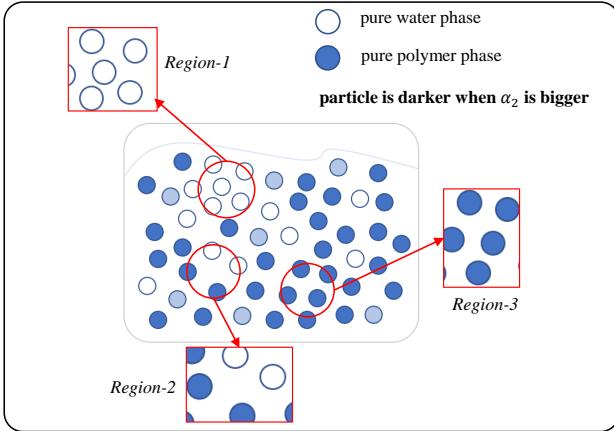


Figure 3: Solution dynamics. Particle colors are a mixture of phase fractions – darker colors show higher fractions of polymer phase. The fluid's local rheological properties are affected by the solution's local concentration. For a dynamic solution system, the properties of the local fluid elements may also be different if the polymer's concentration in the above three regions is different: Region-1 shows properties of pure water; Region-3 shows viscoelastic polymer properties; Region-2 is somewhere in between.

volume fraction of the polymer phase in the mixture model. Hence, an artificial threshold can be set to adjust the stress calculation model, we use a volume fraction threshold α_0 to handle this.

For the shear thinning fluid, we update the thinning factor β in Eqn. 15 by

$$\beta = \beta_0 \frac{\max(0, \alpha_0 - \alpha_2)}{\alpha_0 + \epsilon}, \quad (24)$$

where β_0 is the basic thinning factor. When $\alpha_2 < \alpha_0$, shear thinning will be stronger, and if α_2 it will be weaker; $\alpha_0 = 0$ recovers the basic viscoelastic model and describes a Newtonian fluid; ϵ is a regularization factor used to ensure numerical stability.

For the shear thickening, we use the Cross model, a kind of non-Newtonian viscosity model [Cro65], to calculate shear-related viscosity changes. The classical Cross model is formulated as

$$\mu(\gamma) = \mu_\infty + (\mu_0 - \mu_\infty) \frac{1}{1 + (l\gamma)^N}, \quad (25)$$

where $\gamma = \sqrt{\frac{1}{2} \text{tr}((\nabla \bar{v} + (\nabla \bar{v})^T)^2}$ is the shear rate; μ_0 and μ_∞ are limiting values of viscosity; l is a scaling factor; and N is a smoothing factor.

In our model, we need to consider two aspects to change the viscosity of the fluid: the change caused by the polymer concentration and the shear rheology.

Inspired by the classical suspension model [WC14], we describe the non-Newtonian properties of fluids by *microstructures*. Instead of explicitly computing and storing microstructures, we scale the rest dynamic viscosity by establishing a nonlinear relationship between the polymer particle phase fraction and the scaling factor.

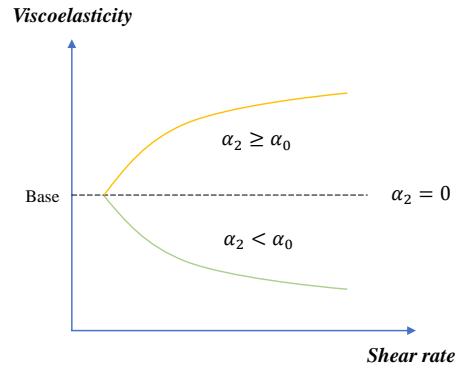


Figure 4: The Viscoelasticity-ShearRate curve is divided into the shear thinning interval and the shear thickening interval, and the rheological properties are calculated dynamically through α_2 .

Following Royer [RBH16] and Hermes [HGP*16] on the relationship between the volume fraction and the viscosity of the polymer particle component, we compute the scaling factor

$$\kappa^i = (1 - \alpha_2^i)^{-n} \quad (26)$$

and apply it to the solution viscosity, where α_2^i is the volume fraction of the polymer phase in particle i , and n denotes the strength of the nonlinear relationship (set empirically in our work to $n = 1.2$). The rest dynamic viscosity is computed as

$$\eta_s^0 = \kappa \eta_0, \quad (27)$$

where η_0 is the basic viscosity of the solution.

Finally, we compute the effective dynamic viscosity as

$$\eta_s(\gamma) = \eta_s^0 + (\eta_{\max} - \eta_s^0) \frac{1}{1 + (l\gamma)^N} \frac{\max(0, \alpha_2 - \alpha_0)}{1 - \alpha_0 + \epsilon}. \quad (28)$$

The above model can be seen as two acting ‘intervals’ which maintain the same shear properties in each; the interval’s upper and lower bounds give the upper and lower limits of shear strength. Shear thinning properties generally occur at low solution concentrations, when water lubrication in the solution exceeds friction between polymer molecules. In Eqn. 24, $\frac{\max(0, \alpha_0 - \alpha_2)}{\alpha_0 + \epsilon}$ shows that when α_2 gradually increases in the shear thinning interval, the increase of local concentration weakens the thinning effect, thus reducing the thinning factor of the nonlinear term. When the concentration rises and exceeds the threshold, the nonlinear term in Eqn. 15 vanishes and activates the calculation of shear thickening. In Eqn. 28, $\frac{\max(0, \alpha_2 - \alpha_0)}{1 - \alpha_0 + \epsilon}$ shows that the higher the concentration, the larger the scaling factor and the higher the solution viscosity. When $\alpha_0 = 0$, we simply set $\eta_s = \eta_0$.

The viscoelastic capacity of our non-Newtonian model as a function of shear is plotted in Fig. 4.

4.3. Bonding Effects Network

In the mixture model, the phase transfer step is only affected by drift velocity and diffusion. In the implicit mixture model, C_d is

used to adjust the effect of mixture on the phase motion, but it is a constant. This means that, regardless of the dynamic state of the fluid, the mixture affects the phase in the same proportion. This is not true in solutions with significant bonding interactions, such as starchy water, which are easy to mix and difficult to separate, and cannot be simulated if *constant* C_d values are used. To alleviate this, we propose a bond cooperation network that dynamically modifies the parameter C_d at each time step according to the fluid's local concentration, as follows.

For a simulation, we set a basic drag parameter C_d^0 , making next C_d depend on it. The core concept of the bonding effects network is *obstruction*: For particle i , the polymer concentration carried by its neighbors hinders the momentum exchange between the phases, which causes the C_d value to rise there. We use the SPH method to estimate particle i 's value C_d (see also Figure 5) as

$$C_d = C_d^0 + (1 - C_d^0) \sum_{j \neq i}^j V^j \alpha_2^j W^{ij}, \quad (29)$$

where V^j is the volume of particle j , and $\sum_{j \neq i}^j V^j \alpha_2^j W^{ij}$ denotes the blocking factor.

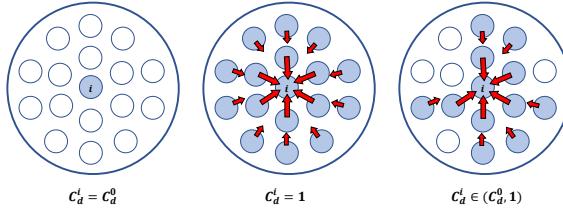


Figure 5: Three bonding effect cases. Red arrows show the blocking effect – the longer the arrows, the stronger this effect. When particle i 's neighbors are all pure water, no shift occurs and $C_d^i = C_d^0$. When neighbors are all pure polymer, momentum exchange blocking occurs, yielding a no-drift scenario and $C_d^i = 1$. A more common scenario is a neighbor with an unsaturated phase which activates the blocking effect. Particles in the figure are drawn using pure water or pure polymer phases; the case of particles with unsaturated phases can already be described by these three examples.

5. Implementation

We implemented our model in CUDA 12.3. All experiments discussed next were run on a single NVidia RTX 3050Ti card. All rel-

evant parameters are summarised in Table 1. The first two parameters adjust the mixing process; the latter ones control the desired non-Newtonian fluid effects. The model's elastic calculation depends mainly on parameter λ and viscous behavior on $\eta_f, \eta_0, \eta_{\max}$. When $\lambda \rightarrow 1$, the model is purely elastic; when $\eta \rightarrow \infty$ or $\lambda \rightarrow 0$, the model is purely viscous.

Table 2 shows an overview of our algorithm and its main steps. We start by setting the conformation tensor to \mathbf{I} for all particles. We compute particle neighbors (here and next) by a unified grid-based neighbor search [Gre10]. Next, the simulation loop starts, consisting of six steps. **Step 1** updates the current state of the fluids and precomputes the viscoelastic solver related parameters. **Step 2** uses a DFSPH solver to compute the pressure momentum and next uses this to update the phase velocities and mixture velocity to keep the velocity field divergence-free. **Step 3** updates the fluid field based on gravity. Surface tension forces also can be added here. **Step 4** uses DFSPH to enforce the fluid's incompressibility. **Step 5** employs our viscoelastic solver to compute viscoelastic stress forces, used after the pressure solver to reduce the latter's influence on the viscoelastic solver. Finally, **Step 6** updates the volume fractions, particle positions, and particles' neighbors.

6. Results

In this section, we present several results obtained using our proposed multiphase model. It is noteworthy that our method enables the adjustment of fluid flow using a uniform set of parameters, eliminating the need to switch between different models. All experiments discussed herein employ an explicit scheme to update the fluid's conformation tensor.

Viscoelastic capacity: Figure 6 shows a set of multiphase Newtonian fluids ($\alpha_0 = 0$) with different viscosity and relaxation time, where we set $C_d^0 = 1$ to stabilize the rheological properties. Except for the small ball scenario, the other scenarios adopt a fixed time step of 1.5ms, which cannot be achieved by the viscosity calculation scheme in the implicit mixture model or by the existing explicit viscosity solvers, such as explicit Laplacian viscosity. In the upper part of Figure 6, η_0 and η_{\max} range in $[0.5, 20]$, and λ ranges in $[0.005, 0.3]$ in the lower part.

Non-Newtonian fluid: Figures 7 and 8 show a shear-dependent non-Newtonian fluid. Our non-Newtonian mechanism assigns computational flows primarily by polymer phase fractions. In Figure 7, for an intuitive comparison, we set three different types of fluids simultaneously **in one single simulation**: **Left**: Newtonian, **Middle**: shear thinning, and **Right**: shear thickening. It is worth noting that in existing non-Newtonian solvers based on specific non-Newtonian viscosity models (Cross, Carreau, et al.), simulating different types of fluids requires manually changing the parameters of the non-Newtonian viscosity model, which means that different types of fluids cannot be considered in one single simulation, and more dynamic scenarios cannot be achieved. The scenario mainly examines the static non-Newtonian properties and hence we set $C_d^0 = 1$ to keep a homogeneous solution. Figure 8 shows a simple comparison of our model with a non-Newtonian solver based on the Cross model for another set of **single simulations**. This scenario examines dynamic flow effects with phase transfer, where the

Table 1: Model Symbols

Parameter	Meaning	Range
C_f	diffusion coefficient	$[0, 1]$
C_d^0	rest drag coefficient	$[0, 1]$
β_0	rest shear thinning factor	$[0, 1]$
α_0	phase volume fraction threshold	$[0, 1]$
η_f	rest viscosity of water	$(0, 0.02)$
η_0	basic viscosity of solution	$(0, 20)$
η_{\max}	max viscosity of solution	$(0, 20)$
λ	relaxation time	$(0, 1)$

Table 2: Non-Newtonian fluid simulation algorithm

Initialization:
set $\mathbf{C} = \mathbf{I}$
Update Neighbors
Loop:
1. Precomputation:
for all particles:
compute mixture velocity $\bar{\mathbf{v}}$ using Eqn. 3
compute phase drift velocities \mathbf{v}_k^d using Eqn. 2
compute mixture rest density $\bar{\rho}$ using Eqn. 4
compute β and κ using Eqns. 24 and 26
compute drag coefficient C_d using Eqn. 29
2. VFSPH divergence-free Solver:
for all particles:
compute pressure momenta $\mathbf{M}_k^{p_{\text{div}}}$ using DFSPH
update phase velocities \mathbf{v}_k using Eqn. 9
recompute mixture velocity $\bar{\mathbf{v}}$ using Eqn. 3
3. Advection:
for all particles:
update phase velocity from \mathbf{M}_k^g using $\mathbf{v}_k + = \mathbf{g}\Delta t$
recompute mixture velocity $\bar{\mathbf{v}}$ using Eqn. 3
4. VFSPH incompressible Solver:
for all particles:
compute pressure momentum $\mathbf{M}_k^{p_{\text{incomp}}}$ using DFSPH
update phase velocity \mathbf{v}_k using Eqn. 9
recompute mixture velocity $\bar{\mathbf{v}}$ using Eqn. 3
5. Viscoelastic Solver:
for all particles:
update conformation tensor \mathbf{C} using Eqn. 15
update phase velocity from \mathbf{M}_k^{vise} using Eqns. 17 and 23
recompute mixture velocity $\bar{\mathbf{v}}$ using Eqn. 3
6. Finalize:
Phase Transfer:
update volume fraction using Eqns. 10 and 11
for all particles:
recompute each phase drift velocity \mathbf{v}_k^d using Eqn. 2
update particle positions \mathbf{x}^i using $\mathbf{x}^i + = \bar{\mathbf{v}}\Delta t$
Update Neighbors

rheological properties of the fluid may change as the phase fraction changes. We set $C_d^0 = 0.8, \alpha_0 = 0.65, \beta_0 = 0.1$. For these two scenes, $\eta_0 = 2, \eta_{\max} = 5$.

Buckling effect: The buckling effect is a complex flow phenomenon of highly viscous fluids, which mainly manifests itself by the maintenance of angular momentum of the model. Our model reproduces this experiment, which proves that our model is able to guarantee the conservation of angular momentum; see Figure 9. We set $C_d^0 = 0.85, \lambda = 0.0075, \alpha_0 = 0.1, \beta_0 = 0.1, \eta_0 = 10, \eta_{\max} = 13$.

Bonding effect network: The network is mainly used to mitigate phase separation, which helps in easy to mix, hard to separate scenarios. In the original implicit mixture model, the constant C_d leads to the same mixing and separating mechanism, which means that mixing and separating occur at the same rate. For scenarios like starch water, relatively easy mixing can be achieved by stirring, but we also need to maintain the mixing state for a certain amount of

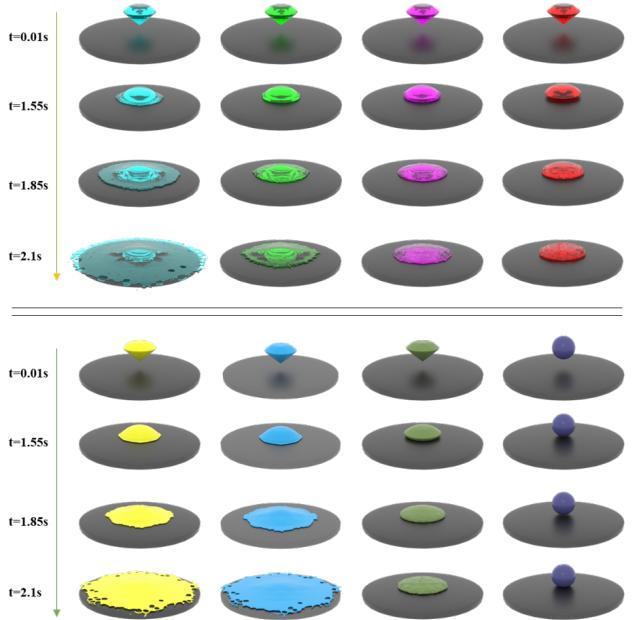


Figure 6: Diamonds and a ball with different viscoelasticity values and 281k particles. The timeline in the upper part shows the flow of different viscosity parameters (η_0, η_{\max}) with equal elastic coefficients (relaxation time λ). The first three diamond simulations in the timeline in the lower part show the flow with different elastic parameters with equal viscosity. The last small ball demonstrates the bouncing effect at high viscoelasticity.

time, so the constant setting of C_d is not applicable. We visually observe the blocking effect of the bonding network on phase separation through a simple scenario; see Figure 10. We set volume fraction $\alpha = (0.3, 0.7)$ and $C_d^0 = 0.3, C_f = 0.0005, \lambda = 0.004, \alpha_0 = 0, \eta_0 = \eta_{\max} = 1$.

Mix into cement: This experiment mainly investigates the comprehensive ability of the model. The process of mixing water and sediment to form cement reflects easy mixing, difficult separation, local rheological differences and viscoelastic ability. This scene can also be extended to more dynamic scenes, such as making dough; see Figure 11. We set $C_d^0 = 0.4, C_f = 0.015, \lambda = 0.006, \alpha_0 = 0.65, \beta_0 = 0.2, \eta_0 = 6, \eta_{\max} = 13$.

"SCA2024" non-Newtonians: This experiment demonstrates a dynamic non-Newtonian scene. We set 7 different non-Newtonian fluids in one single simulation. Different non-Newtonian fluids are obtained by setting different initial phase fractions. We set $C_d^0 = 0.6, C_f = 0.015, \lambda = 0.006, \alpha_0 = 0.55, \beta_0 = 0.25, \eta_0 = 6, \eta_{\max} = 10$. For the letters/numbers, we set 'S' to $\alpha = (0.1, 0.9)$, 'C' to $\alpha = (0.3, 0.7)$, 'A' to $\alpha = (0.4, 0.6)$, the first '2' to $\alpha = (0.5, 0.5)$, '0' to $\alpha = (0.6, 0.4)$, the second '2' to $\alpha = (0.7, 0.3)$, and finally '4' to $\alpha = (0.9, 0.1)$; see Figure 1.

Table 3 outlines our method's capabilities vs with three other established fluid simulation solvers. Table 4 compares four viscoelastic scheme timings under different viscosities. Table 4 shows that in

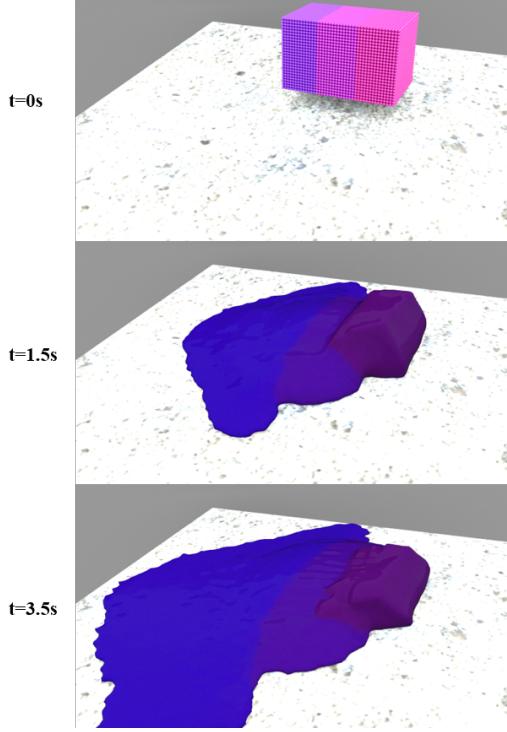


Figure 7: A fluid block (52k particles) falls on an inclined plate. Left block of particles: Newtonian, Middle block: shear thinning, Right block: shear thickening. After contact with the inclined plate, the Newtonian fluid always maintains its own flow. 0–1.5s: the shear thinning fluid reduces the viscosity due to shear effect, and promotes the flow; the shear thickening fluid increases viscosity and blocks flow. 1.5–3.5s: as the shear effect weakens, the shear thinning fluid viscosity increases, and the flow slows down. The shear thickening fluid viscosity decreases, facilitating flow.

high viscosity scenarios, our approach has significant advantages over other explicit schemes and can also include a wider viscosity range.

Table 3: Comparison of method capabilities

Solver	High viscosity ($\Delta t \geq 1.5ms$)	Dynamic Mixing	Non-Newtonian fluids	Easy mixing but separation blocked	Multitype fluids in one simulation
Ours	✓	✓	✓	✓	✓
IMM	✗	✓	✗	✗	✗
DFSPH _{ev}	✗	✗	✓	✗	✗
DFSPH _{iv}	✓	✗	✓	✗	✗

IMM: Implicit Mixture Model

DFSPH_{ev}: DFSPH with Explicit Viscosity

DFSPH_{iv}: DFSPH with Implicit Viscosity

7. Conclusion

We proposed a new simulation technique for non-Newtonian fluids in a multiphase framework that makes use of the concept of mixtures so as to closely mimic the real behavior of fluids. Our method can adapt to different non-Newtonian fluid models and can

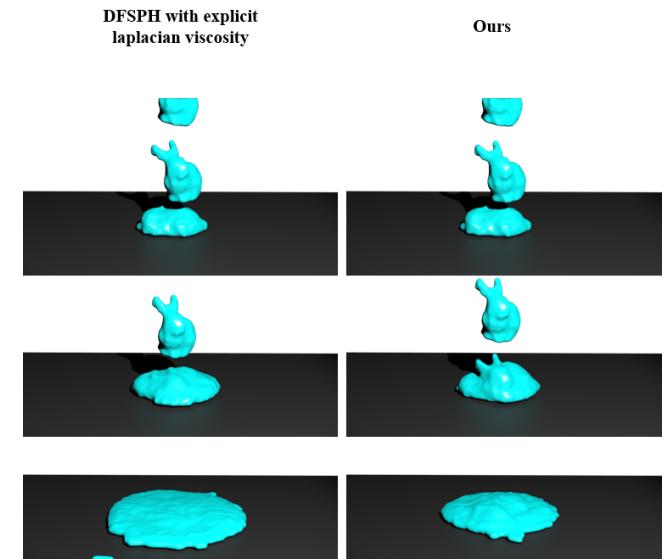


Figure 8: Three bunnies falling to the ground. 73k particles. For our model, they are **Below**: newton, **Middle**: shear thinning, **Top**: shear thickening. And three same shear thinning bunnies in another scenario. In our scenario, the flow is composite and dynamic, and influenced by the polymer phase fraction, the fluid more closely resembles a viscoelastic shear thickening fluid.

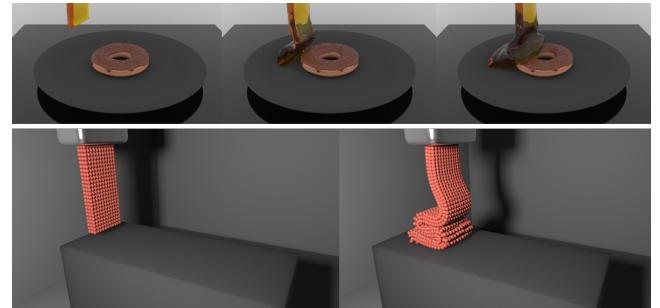


Figure 9: The top scene shows honey poured over a donut and the bottom scene shows another buckling scene rendered as a particle state with 158k particles. The scenes have a stable and obvious buckling effect, which proves the stability of our viscoelastic computation.

reproduce their behavior for a wide range of viscosities. Also, our method is able to model a certain degree of elastomer behavior. Compared with general single-phase solvers with explicit viscosity and the original implicit mixture model, our method is more computationally efficient in highly viscous scenarios.

There are a few directions worth exploring. In this work, we used the explicit solution. An implicit solution of the conformation tensor method under a Lagrangian method may be able to improve the stability of the algorithm and improve its time stepping. The problem brought by the explicit solution is that numerical errors may occur under high stress, resulting in undesirable fracture be-

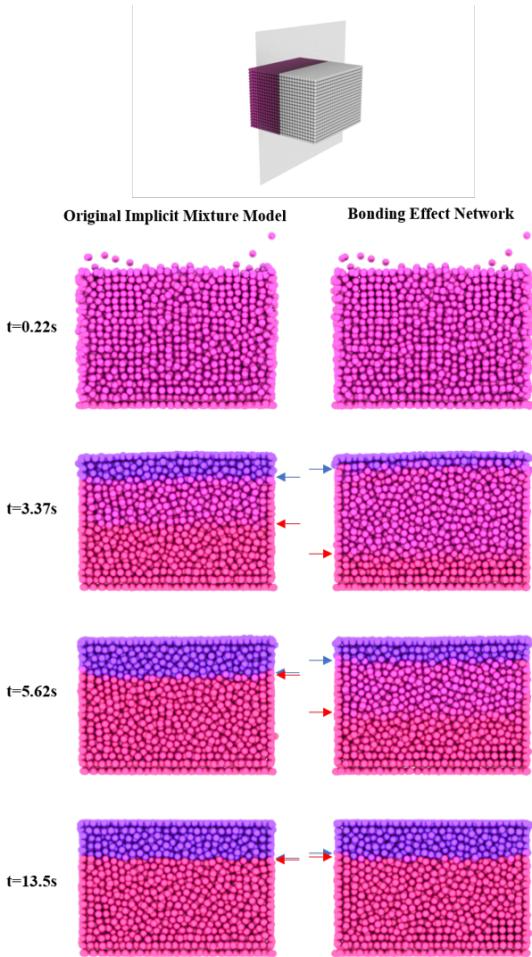


Figure 10: The blocking effect of Bonding Effect Network on phase transfer (98k particles). We use a plane to segment the fluid to observe the internal state of the fluid. The arrows indicate distinct phase interfaces. The left shows phase separation of the original implicit mixture model, $C_d = 0.3$. Our model’s result with $C_d^0 = 0.3$ is on the right. It can be clearly seen that our model has a blocking effect on phase separation, which contributes to the scenario where the fluid achieves easy mixing and remains mixed at low C_d^0 values.

havior. This problem mostly occurs in ultra-high elastic scenes, and the resolution of the background grid may also affect the calculation. Multiscale multiphase flow simulation techniques should have higher potential for non-Newtonian fluid simulation. In our experiments, we also found that our model has the potential to be exploited in elastomer simulations as well.

References

- [ASP*14] ANDRADE L. F. D. S., SANDIM M., PETRONETTO F., PAGLIOSA P., PAIVA A.: Sph fluids for viscous jet buckling. In *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images* (2014), IEEE, pp. 65–72. [1](#), [2](#)
- [BGF*17] BARREIRO H., GARCÍA-FERNÁNDEZ I., ALDUÁN I.,
- [OTADUY17] OTADUY M. A.: Conformation constraints for efficient viscoelastic fluid simulation. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–11. [2](#)
- [BK15] BENDER J., KOSCHIER D.: Divergence-free smoothed particle hydrodynamics. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics symposium on computer animation* (2015), pp. 147–155. [1](#)
- [BK16] BENDER J., KOSCHIER D.: Divergence-free sph for incompressible and viscous fluids. *IEEE Transactions on Visualization and Computer Graphics* 23, 3 (2016), 1193–1206. [1](#), [2](#)
- [Bro85] BROOKSHAW L.: A method of calculating radiative heat diffusion in particle simulations. *Publications of the Astronomical Society of Australia* 6, 2 (1985), 207–210. [2](#)
- [BTRD11] BALCI N., THOMASES B., RENARDY M., DOERING C. R.: Symmetric factorization of the conformation tensor in viscoelastic fluid models. *Journal of Non-Newtonian Fluid Mechanics* 166, 11 (2011), 546–553. [4](#)
- [CBL*09] CHANG Y., BAO K., LIU Y., ZHU J., WU E.: A particle-based method for viscoelastic fluids animation. In *Proceedings of the 16th ACM symposium on virtual reality software and technology* (2009), pp. 111–117. [1](#), [3](#)
- [Cro65] CROSS M. M.: Rheology of non-Newtonian fluids: a new flow equation for pseudoplastic systems. *J. Colloid Sci.* (1965). [1](#), [5](#)
- [FBGZ19] FEI Y., BATTY C., GRINSPUN E., ZHENG C.: A multi-scale model for coupling strands with shear-dependent liquid. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–20. [2](#)
- [FLX*22] FENG F., LIU J., XIONG S., YANG S., ZHANG Y., ZHU B.:

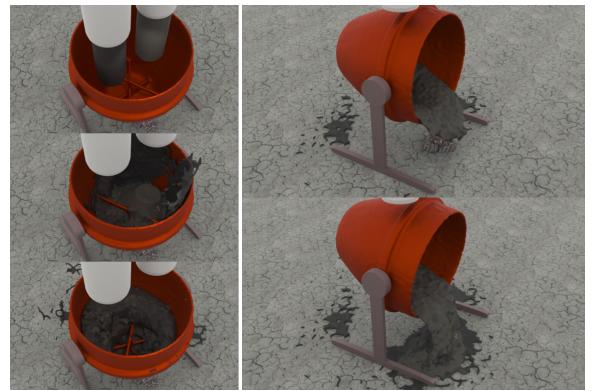


Figure 11: Mix into cement (150k particles). The liquid column on the left is the pure water phase and the liquid column on the right is the pure polymer phase. To create the cement effect, we added a 1 : 2 volume ratio of water to polymer, and set a low $C_d^0 = 0.4$ and a high viscosity. It can be seen that the two liquid columns can be stably mixed and maintain a high viscoelastic mixing state.

Table 4: Timings of methods. The table lists the maximum time step Δt (in ms) for each method in the corresponding viscosity range; η is listed in m^2/s .

Solver	$\eta \leq 0.001$	$\eta \in [0.001, 1]$	$\eta \in [1, 10]$	$\eta \in [10, 20]$
Ours	≤ 3	≤ 3	≤ 1.5	≤ 0.1
IMM	≤ 3	≤ 0.1	-	-
DFPSH _{ev}	≤ 10	≤ 0.5	-	-
DFPSH _{iv}	≤ 10	≤ 10	≤ 5	≤ 1

- Impulse fluid simulation. *IEEE Transactions on Visualization and Computer Graphics* (2022). 1
- [FM07] FUJISAWA M., MIURA K. T.: Animation of ice melting phenomenon based on thermodynamics with thermal radiation. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia* (2007), pp. 249–256. 1, 3
- [Gre10] GREEN S.: Particle simulation using cuda. *NVIDIA whitepaper* 6 (2010), 121–128. 6
- [GWAB19] GOLDADE R., WANG Y., AANJANEYA M., BATTY C.: An adaptive variational finite difference framework for efficient symmetric octree viscosity. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14. 2
- [HGP*16] HERMES M., GUY B. M., POON W. C., POY G., CATES M. E., WYART M.: Unsteady flow and particle migration in dense, non-brownian suspensions. *Journal of Rheology* 60, 5 (2016), 905–916. 5
- [HK03] HONG J.-M., KIM C.-H.: Animation of bubbles in liquid. In *Computer Graphics Forum* (2003), vol. 22, Wiley Online Library, pp. 253–262. 1, 3
- [HKA21] HIRASAWA N., KANAI T., ANDO R.: A flux-interpolated advection scheme for fluid simulation. *The Visual Computer* 37, 9–11 (2021), 2607–2618. 1
- [IBAT11] IHMSEN M., BADER J., AKINCI G., TESCHNER M.: Animation of air bubbles with sph. In *International Conference on Computer Graphics Theory and Applications* (2011), vol. 2, SCITEPRESS, pp. 225–234. 1, 3
- [JL21] JIANG Y., LAN Y.: A dynamic mixture model for non-equilibrium multiphase fluids. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 85–95. 2, 3
- [JWL*23] JESKE S. R., WESTHOFEN L., LÖSCHNER F., FERNÁNDEZ-FERNÁNDEZ J. A., BENDER J.: Implicit surface tension for sph fluid simulation. *ACM Transactions on Graphics* 43, 1 (2023), 1–14. 1
- [KAL06] KIM T., ADALSTEINSSON D., LIN M. C.: Modeling ice dynamics as a thin-film stefan problem. In *Symposium on Computer Animation: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation: Vienna, Austria* (2006), vol. 2, pp. 167–176. 1, 3
- [KHL04] KIM T., HENSON M., LIN M. C.: A hybrid algorithm for modeling ice formation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 305–314. 1, 3
- [KL03] KIM T., LIN M. C.: Visual simulation of ice crystal growth. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003), Citeseer, pp. 86–97. 1, 3
- [LGH*23] LI C., GAO Y., HE J., CHENG T., LI S., HAO A., QIN H.: A unified particle-based solver for non-newtonian behaviors simulation. *IEEE Transactions on Visualization and Computer Graphics* (2023). 2
- [LHWLW21] LIU S., HE X., WANG W., WU E.: Adapted simple algorithm for incompressible sph fluids with a broad range viscosity. *IEEE Transactions on Visualization and Computer Graphics* 28, 9 (2021), 3168–3179. 1, 2
- [MKN*04] MÜLLER M., KEISER R., NEALEN A., PAULY M., GROSS M., ALEXA M.: Point based animation of elastic, plastic and melting objects. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 141–151. 1, 3
- [MnTK96] MANNINEN M., TAIVASSALO V., KALLIO S.: On the mixture model for multiphase flow. *VTT Publications*, 288 (1996), 3–67. 3
- [PGG*23] PANUELOS J., GOLDADE R., GRINSPUN E., LEVIN D., BATTY C.: Polystokes: A polynomial model reduction method for viscous fluid simulation. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–13. 2
- [PICT15] PEER A., IHMSEN M., CORNELIS J., TESCHNER M.: An implicit viscosity formulation for sph fluids. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–10. 1, 2
- [RBH16] ROYER J. R., BLAIR D. L., HUDSON S. D.: Rheological signature of frictional interactions in shear thickening suspensions. *Physical review letters* 116, 18 (2016), 188301. 5
- [RLY*14] REN B., LI C., YAN X., LIN M. C., BONET J., HU S.-M.: Multiple-fluid sph simulation using a mixture model. *ACM Transactions on Graphics (TOG)* 33, 5 (2014), 1–11. 3
- [SB12] SCHECHTER H., BRIDSON R.: Ghost sph for animating water. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–8. 2
- [SXH*21] SU H., XUE T., HAN C., JIANG C., AANJANEYA M.: A unified second-order accurate in time mpm formulation for simulating viscoelastic liquids with phase change. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–18. 2
- [TDF*15] TAKAHASHI T., DOBASHI Y., FUJISHIRO I., NISHITA T., LIN M. C.: Implicit formulation for sph-based viscous fluids. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 493–502. 1, 2
- [TFK*03] TAKAHASHI T., FUJII H., KUNIMATSU A., HIWADA K., SAITO T., TANAKA K., UEKI H.: Realistic animation of fluid with splash and foam. In *Computer Graphics Forum* (2003), vol. 22, Wiley Online Library, pp. 391–400. 1, 3
- [WC14] WYART M., CATES M. E.: Discontinuous shear thickening without inertia in dense non-brownian suspensions. *Physical review letters* 112, 9 (2014), 098302. 5
- [WKBB18] WEILER M., KOSCHIER D., BRAND M., BENDER J.: A physically consistent implicit viscosity solver for sph fluids. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 145–155. 1, 2
- [WXL*24] WANG X., XU Y., LIU S., REN B., KOSINKA J., TELEA A. C., WANG J., SONG C., CHANG J., LI C., ZHANG J. J., BAN X.: Physics-based fluid simulation in computer graphics: Survey, research trends, and challenges. *Computational Visual Media* (Apr 2024). 1
- [XWW*23] XU Y., WANG X., WANG J., SONG C., WANG T., ZHANG Y., CHANG J., ZHANG J. J., KOSINKA J., TELEA A., ET AL.: An implicitly stable mixture model for dynamic multi-fluid simulations. In *SIGGRAPH Asia 2023 Conference Papers* (2023), pp. 1–11. 1, 2, 3
- [ZLQF15] ZHU B., LEE M., QUIGLEY E., FEDKIW R.: Codimensional non-newtonian fluids. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–9. 2