An Implicitly Stable Mixture Model for Dynamic Multi-fluid Simulations

YANRUI XU, Beijing Advanced Innovation Center for Materials Genome Engineering, and School of Intelligence Science and Technology, University of Science and Technology Beijing, China, and University of Groningen, Netherlands
XIAOKUN WANG*, School of Intelligence Science and Technology, University of Science and Technology Beijing, China, and Bournemouth University, United Kingdom
JIAMIN WANG, CHONGMING SONG, TIANCHENG WANG, and YANLAN ZHANG, School of Intelligence Science and Technology, University of Science and Technology Beijing, China
JIAN CHANG and JIAN JUN ZHANG, Bournemouth University, United Kingdom
JIŘÍ KOSINKA, University of Groningen, Netherlands
ALEXANDRU TELEA, Utrecht University, Netherlands
XIAOJUAN BAN*, Beijing Advanced Innovation Center for Materials Genome Engineering, and School of Intelligence Science and Technology, University of Science and Technology Beijing, China

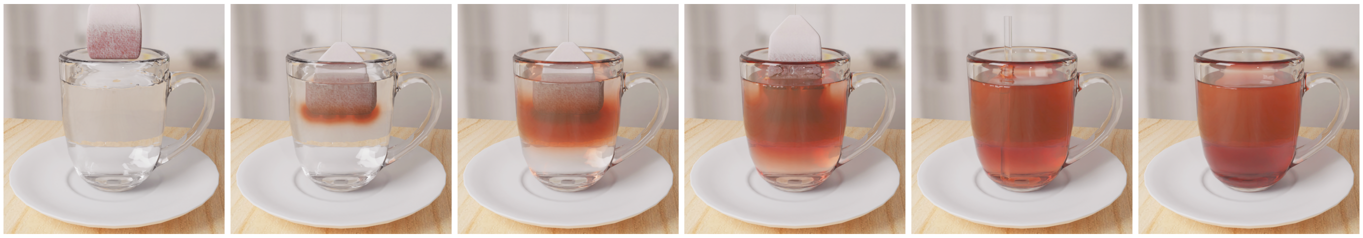


Fig. 1. Tea diffuses from a teabag into a cup of water. After the teabag is removed, the mixture is stirred with a (glass) rod.

Particle-based simulations have become increasingly popular in real-time applications due to their efficiency and adaptability, especially for generating highly dynamic fluid effects. However, the swift and stable simulation of interactions among distinct fluids continues to pose challenges for current mixture model techniques. When using a single-mixture flow field to represent all fluid phases, numerical discontinuities in phase fields can result in significant losses of dynamic effects and unstable conservation of mass and momentum. To tackle these issues, we present an advanced implicit mixture model for smoothed particle hydrodynamics. Instead of relying on an explicit mixture field for all dynamic computations and phase transfers between particles, our approach calculates phase momentum sources from the mixture model to derive explicit and continuous velocity phase fields. We then implicitly obtain the mixture field using a phase-mixture momentum-mapping mechanism that ensures conservation of incompressibility, mass, and momentum. In addition, we propose a mixture viscosity model and establish viscous effects between the mixture and individual fluid phases to avoid instability under extreme inertia conditions. Through a series of experiments, we show that, compared to existing mixture models, our method effectively improves dynamic effects while reducing critical instability factors. This makes our approach especially well-suited for long-duration, efficiency-oriented virtual reality scenarios.

CCS Concepts: • **Computing methodologies → Physical simulation**.

Additional Key Words and Phrases: fluid simulation, mixture models, incompressible fluids

ACM Reference Format:
Yanrui Xu, Xiaokun Wang, Jiamin Wang, Chongming Song, Tiancheng Wang, Yanlan Zhang, Jian Chang, Jian Jun Zhang, Jiří Kosinka, Alexandru Telea, and Xiaojuan Ban. 2023. An Implicitly Stable Mixture Model for Dynamic Multi-fluid Simulations. In *SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers '23), December 12–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3610548.3618215

1 INTRODUCTION

Simulating multiphase flows [Brennen 2005] has gained interest in the graphics community due to its ability to create distinct diffusion and stratification effects, such as brewing tea bags (Fig. 1) and crafting cocktails. Simplified mixture models [Manninen et al. 1996] have been widely adopted to produce stunning visuals [Jiang and Lan 2021; Ren et al. 2014] at lower computational cost.

Mixture models deal with the coexistence of multiple phases within a singular discretization unit. A distinctive drift velocity is leveraged to illustrate the relative motion between the mixture and individual phases. This movement stems from the multiphase governing equations, which, in turn, find their roots in the single-phase Navier-Stokes equations. A significant challenge in this domain

[*] Corresponding authors

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0315-7/23/12…$15.00
https://doi.org/10.1145/3610548.3618215

arises from the fact that many fluid simulation techniques are tailored primarily for single-phase fluids. Designing a suitable method that can integrate additional sub-discretization unit dynamics computation is far from trivial.

A common workaround to this complexity has been the adoption of the *Local Equilibrium Assumption (LEA)* [Jiang et al. 2020; Ren et al. 2022, 2014]. While the LEA simplifies simulations by forcing drift states back to non-drift states, enabling compatibility with existing simulation models, it is not without limitations. Such an approach can often introduce discontinuities in phase velocities, compromising the transfer between phases during time integration. Consequently, this diminishes the dynamism stemming from multiphase interactions.

Alternatively, energy-based multiphase models have been explored [Tampubolon et al. 2017; Yang et al. 2015]. Although they can bypass the challenges of LEA and drift velocity computations associated with phase-phase interactions, these models often make concessions in numerical accuracy to align with existing simulation structures.

Numerical instability is another challenge for multiphase flow simulation. The standard Smoothed Particle Hydrodynamics (SPH) strategy [Bender and Koschier 2017] struggles to conserve mass and momentum using the volume fraction scheme. Jiang *et al.* [2021] abandoned LEA for improved dynamic performance but sacrificed mixture incompressibility and mass conservation.

We propose an implicit mixture model to address dynamic and stability issues in multiphase flow simulation. We construct interphase momentum sources using no-slip and free-slip conditions among phases, calculate the change rate of phase velocity fields, and conserve divergence-free conditions for mixture flow and numerical consistency for phase velocities. To avoid instability caused by excessive drift velocities, we introduce a mixture viscosity model to smoothly limit drift velocity while maintaining conservation. The main contributions of our implicit mixture model are:

- An *analytic form* of interphase momentum sources describing how the mixture flow affects the movement of individual fluid phases;
- A phase-mixture *momentum mapping mechanism* that preserves the incompressibility of the mixture and ensures numerical consistency and accuracy of the phases;
- A mixture *viscosity model* to prevent instability arising from overly dynamic interphase movement.

2 RELATED WORK

Fluid simulation is well-researched in computer graphics. For an overview, we refer to Bridson's book [Bridson 2015] and the report by Koschier *et al.* [2019]. We next discuss related work on multiphase flow simulation focusing on (im)miscible fluids [Ren et al. 2018].

Immiscible fluids. Research on immiscible fluids focuses on discontinuities at interfaces [Boyd and Bridson 2012; Hong and Kim 2005; Kim 2010; Li et al. 2021; Losasso et al. 2006]. Simulations range from simple particle collision computations [Mao and Yang 2006] to more advanced approaches, like distinct phase labeling and attribute assignment in MPS [Premžoe et al. 2003] and SPH [Müller et al. 2005] methods. The number density concept [Solenthaler and Pajarola 2008] improved capturing interface density discrepancies to model solid and multiphase fluid interactions [Akinci et al. 2012; Band et al. 2018]. Extensions include position-based fluid methods [Alduán et al. 2017] and hybrid simulations like FLIP [Ando et al. 2015; Boyd and Bridson 2012] and LBM [Guo et al. 2017].

Mixture models. A pivotal aspect of fluid simulations, especially when interacting with intricate interfaces, is the precise characterization of the mixture state. The *Mixture model*, leveraging the *volume fraction* approach, has become a mainstay. It is widely applied into a variety of coupling scenarios for fluid simulation. Notably, Daviet and Bertails-Descoubes [2017] incorporated mixture theory for granular materials, negating the need for fluid-solid coupling at granular levels, thereby achieving computational economy. Analogous challenges emerge in fluid-fabric dynamics. Fei *et al.* [2018] devised a model that adeptly captures the intrinsic physical attributes of fabric, both structurally and texturally, especially under saturation conditions. In the realm of underwater bubble simulations, Wretborn *et al.* [2022] employed mixture models to cohesively integrate bubble and water phases using continuum mechanics equations, obviating intricate parameter adjustments and delivering authentic whitewater visualizations.

For coupling between fluids, Müller *et al.* [2005] introduced volume fraction for miscible multi-fluid systems, later adopted by grid-based [Bao et al. 2010; Kang et al. 2010] and SPH solvers [Liu et al. 2011]. However, these models overlooked fluid mixing/separation due to flow motion and force distribution. Drift-velocity-based models [Ren et al. 2014; Yan et al. 2016] addressed these and enhanced mixing and unmixing simulations. Yet, limitations remained, such as incompatibility with incompressible solvers [Ren et al. 2014; Yan et al. 2016]. Jiang *et al.* proposed further refinements with divergence-free [Jiang et al. 2020] an dynamic mixture models [Jiang and Lan 2021]. However, the latter model still struggles with maintaining incompressibility and with numerical error reduction [Jiang and Lan 2021; Jiang et al. 2020].

Energy-based multi-fluid simulations, such as Yang *et al.* [2015], use the Helmholtz free energy and the Cahn-Hilliard equation for complex fluid interactions. However, these methods lack high-order accuracy for drift velocity and suffer from slow convergence [Yang et al. 2017, 2015]. Hybrid methods like the Material Point Method (MPM) offer advantages for simulating sand-water and sediment mixtures [Gao et al. 2018; Tampubolon et al. 2017].

Previous mixture models relying on LEA or ignoring relative motion struggle to capture behaviors in dynamic scenarios [Jiang and Lan 2021; Ren et al. 2014]. Jiang *et al.* [2021] proposed a dynamic mixture model, but faced challenges with mass and volume conservation and solver integration.

We present an implicit mixture model for mass conservation and enhanced numerical accuracy. Drawing inspiration from the force balance model [Manninen et al. 1996], we unify the motion of individual phases and the mixture for improved interphase effects. Our method integrates seamlessly with existing implicit Lagrangian pressure solvers [Band et al. 2018; Solenthaler and Pajarola 2008] with only minimal modifications.

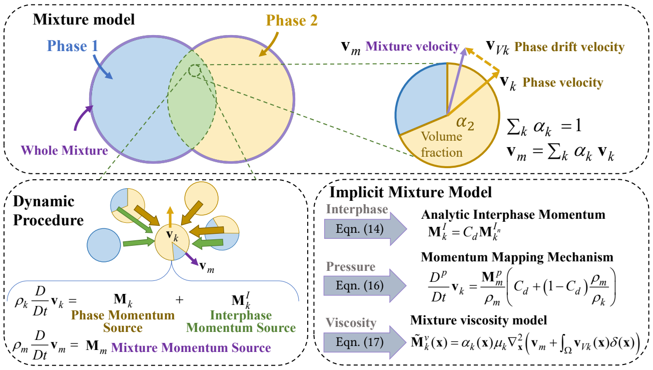

Mixture model
Phase 1
Phase 2
Whole Mixture
$\mathbf{v}_m$ Mixture velocity
$\mathbf{v}_{Vk}$ Phase drift velocity
$\mathbf{v}_k$ Phase velocity
α_2
Volume fraction
$\sum_k \alpha_k = 1$
$\mathbf{v}_m = \sum_k \alpha_k \mathbf{v}_k$
Dynamic Procedure
$\mathbf{v}_k$
$\mathbf{v}_m$
$\rho_k \frac{D}{Dt}\mathbf{v}_k = \mathbf{M}_k + \mathbf{M}_k^I$
Phase Momentum Source
Interphase Momentum Source
$\rho_m \frac{D}{Dt}\mathbf{v}_m = \mathbf{M}_m$ Mixture Momentum Source
Implicit Mixture Model
Interphase
Eqn. (14)
Analytic Interphase Momentum
$\mathbf{M}_k^I = C_d \mathbf{M}_k^{I_n}$
Pressure
Eqn. (16)
Momentum Mapping Mechanism
$\frac{D^p}{Dt}\mathbf{v}_k = \frac{\mathbf{M}_m^p}{\rho_m}\left(C_d + (1-C_d)\frac{\rho_m}{\rho_k}\right)$
Viscosity
Eqn. (17)
Mixture viscosity model
$\tilde{\mathbf{M}}_k^v(\mathbf{x}) = \alpha_k(\mathbf{x})\mu_k\nabla_{\mathbf{x}}^2\left(\mathbf{v}_m + \int_\Omega \mathbf{v}_{Vk}(\mathbf{x})\delta(\mathbf{x})\right)$

Fig. 2. Schematic diagram of our proposed implicit mixture model. We use a volume fraction-based mixture model to illustrate the coexistent state of multiphase flows (upper diagram, see Sec. 3). Unlike traditional approaches which use a unified mixture field to compute motion for all phases, we employ a phase-specific interphase momentum mechanism (lower-left diagram, Sec. 4.1) to allow individual phase motion computation. We derive each momentum source for every fluid phase implicitly taking into account both no-slip and free-slip conditions (lower-right diagram, Secs. 4.2 and 4.3, respectively).

3 MIXTURE MODEL

3.1 Volume Fraction Scheme

The volume fraction scheme [Manninen et al. 1996] models the coexistence state of phases within a single fluid parcel (see the upper part of Fig. 2). While the term *particle* refers to the fluid elements in the discretized SPH approach, we use the term *parcel* to denote a small, finite portion of fluid in the continuum hypothesis, representing the fluid in physical equations.

Volume is conserved for all k fluid phases. That is, given the fractionalized volume $\alpha_k \in [0, 1]$ of a fluid parcel at location $\mathbf{x}$, we have that

$$\sum_k \alpha_k(\mathbf{x}) = 1. \tag{1}$$

We next generally omit the dependence on $\mathbf{x}$ to simplify exposition.

Let $\mathbf{v}_k$ denote the velocity field of phase k. The mixture velocity giving the velocity of the volume center of a fluid parcel is

$$\mathbf{v}_m = \sum_k \alpha_k \mathbf{v}_k. \tag{2}$$

The mixture velocity conserves $\nabla \cdot \mathbf{v}_m = 0$ as it represents the mixture's incompressibility [Jiang et al. 2020].

The drift velocity of phase k given by

$$\mathbf{v}_{Vk} = \mathbf{v}_k - \mathbf{v}_m \tag{3}$$

measures the difference between the phase and mixture velocities. According to Eqn. (2), the drift velocity should follow $\sum_k \alpha_k \mathbf{v}_{Vk} = 0$. Similarly, the mixture density can be computed using the rest density ρ_k of each phase as

$$\rho_m = \sum_k \alpha_k \rho_k. \tag{4}$$

3.2 Governing Equations

A mixture model defines the continuity and momentum equations for each phase. Following the original physical model [Manninen et al. 1996], the *continuity equation* reads

$$\frac{\partial}{\partial t}(\alpha_k \rho_k) + \nabla \cdot (\alpha_k \rho_k \mathbf{v}_k) = 0. \tag{5}$$

This equation ensures mass conservation for all phases. By substituting the phase velocity in Eqn. (5) with the drift velocity from Eqn. (3), the volume fraction change rate can be written as

$$\frac{D}{Dt}\alpha_k = -\nabla \cdot (\alpha_k \mathbf{v}_{Vk}), \tag{6}$$

where $\frac{D}{Dt}$ is the material derivative that is subject to the mixture velocity. For any physical attribute A, its material derivative with respect to the mixture velocity $\mathbf{v}_m$ is defined as $\frac{DA}{Dt} = \frac{\partial A}{\partial t} + (\mathbf{v}_m \cdot \nabla)A$. The *momentum equation* for phase k reads

$$\frac{\partial}{\partial t}(\alpha_k \rho_k \mathbf{v}_k) + \nabla \cdot (\alpha_k \rho_k \mathbf{v}_k \mathbf{v}_k) = -\alpha_k \nabla p_k + \nabla \cdot (\alpha_k \boldsymbol{\tau}_k) + \alpha_k \rho_k \mathbf{g} + \mathbf{M}_k^I, \tag{7}$$

where p_k, $\boldsymbol{\tau}_k$, $\mathbf{g}$, and $\mathbf{M}_k^I$ are the pressure, viscous stress tensor, gravity, and interphase momentum source, respectively. Eqn. (7) ensures momentum conservation for all phases. The interphase momentum source $\mathbf{M}_k^I$ models how the motion of a single fluid phase is affected by the entire mixture flow. This obeys a rigorous force balance model [Manninen et al. 1996] complying with $\sum_k \mathbf{M}_k^I = 0$.

Since the standard mixture model does not explicitly define the interphase momentum source $\mathbf{M}_k^I$, various forms of it have been introduced in previous works. Ren *et al.* [2014] proposed a comprehensive term to cover both inertia-induced drag and diffusion effects. Jiang *et al.* [2020] simplified it to model only the drag effect. However, these two terms make it challenging to maintain *stable* simulation systems. The LEA must ensure that the drift velocity returns to zero at the start of each time step. Jiang *et al.* [2021] further presented a more artificial, yet stabler, approach to enable the continuity of drift velocity. In our work, we derive the interphase momentum source directly from Eqn. (5) and Eqn. (7), which can overcome all above difficulties without the need for LEA.

4 IMPLICIT MIXTURE MODEL FOR MULTIPHASE INTERACTIONS

We now introduce our implicit mixture model (see Fig. 2 bottom). To ease reading, Tab. 1 lists all momentum sources used throughout our exposition. Note that the advection momentum source in Tab. 1 is the advection caused by drift velocity, not the term in the Navier-Stokes equation caused by fluid velocity; the latter is the mixture velocity in this paper. We also adopt the notation $p_k = p_m$ for all mixture phases [Manninen et al. 1996; Ren et al. 2014].

4.1 Interphase Momentum

Considering the conservation of mass terms inherited from Eqn. (5), the left-hand side of Eqn. (7) is subject to a transformation. This revised representation, encompassing the predefined momentum sources, is given by

$$\frac{\partial}{\partial t}(\alpha_k \rho_k \mathbf{v}_k) + \nabla \cdot (\alpha_k \rho_k \mathbf{v}_k \mathbf{v}_k) = \alpha_k \rho_k \frac{D}{Dt}\mathbf{v}_k - \mathbf{M}_k^a. \tag{8}$$

Then, Eqn. (7) can be rewritten as

$$\alpha_k \rho_k \frac{D}{Dt}\mathbf{v}_k = \mathbf{M}_k^p + \mathbf{M}_k^v + \mathbf{M}_k^g + \mathbf{M}_k^I + \mathbf{M}_k^a. \tag{9}$$

Table 1. Notations of momentum sources for a mixture of k phases (Sec. 4).

Phase momentum sources $\mathbf{M}_k$			**Corresponding mixture momentum sources $\mathbf{M}_m$**	**Corresponding interphase momentum sources $\mathbf{M}_k^I$**
Component	**Notation**	**Values**	**Notation & Values**	**Notation & Values**
pressure	$\mathbf{M}_k^p$	$-\alpha_k \nabla p_k$	$\mathbf{M}_m^p = \sum_k \mathbf{M}_k^p = -\nabla p_m$	$\mathbf{M}_k^{I_p}$
viscosity	$\mathbf{M}_k^v$	$\nabla \cdot (\alpha_k \boldsymbol{\tau}_k)$	$\mathbf{M}_m^v = \sum_k \mathbf{M}_k^v = \nabla \cdot \boldsymbol{\tau}_m$	$\mathbf{M}_k^{I_v}$
gravity	$\mathbf{M}_k^g$	$\alpha_k \rho_k \mathbf{g}$	$\mathbf{M}_m^g = \sum_k \mathbf{M}_k^g = \rho_m \mathbf{g}$	–
advection	$\mathbf{M}_k^a$	$-\alpha_k \rho_k \mathbf{v}_{Vk} \cdot \nabla \mathbf{v}_k$	$\mathbf{M}_m^a = \sum_k \mathbf{M}_k^a$	$\mathbf{M}_k^{I_a}$
drift	$\mathbf{M}_k^d$	$-\alpha_k \rho_k \frac{D}{Dt} \mathbf{v}_{Vk}$	$\mathbf{M}_m^d = \sum_k \mathbf{M}_k^d$	–

Here, $\alpha_k \rho_k \frac{D}{Dt} \mathbf{v}_k$ can be further split as $\alpha_k \rho_k \frac{D}{Dt} \mathbf{v}_m - \mathbf{M}_k^d$ according to Eqn. (3). By summing up Eqn. (9) for all phases under the force balance model [Manninen et al. 1996] condition $\sum_k \mathbf{M}_k^I = 0$, the mixture's momentum is

$$\rho_m \frac{D}{Dt} \mathbf{v}_m = \mathbf{M}_m^p + \mathbf{M}_m^v + \mathbf{M}_m^g + \mathbf{M}_m^d + \mathbf{M}_m^a. \tag{10}$$

Subtracting Eqn. (10) from Eqn. (9) yields the change rate of drift velocity

$$\frac{D}{Dt} \mathbf{v}_{Vk} = \frac{D}{Dt} \mathbf{v}_k - \frac{D}{Dt} \mathbf{v}_m = \frac{\mathbf{M}_k^{\{p,v,g,a,I\}}}{\alpha_k \rho_k} - \frac{\mathbf{M}_m^{\{p,v,g,d,a\}}}{\rho_m}, \tag{11}$$

where $\{\}$ denotes the sum of the specified momentum sources.

To investigate how interphase momentum affects phase-phase interactions, we consider two extreme analysis scenarios.

No-drift scenario. In this scenario, all phases are tightly coupled, indicating strong interphase momentum. This prevents the mixture from separating, resulting in $\frac{D}{Dt} \mathbf{v}_{Vk} = 0$ and $\mathbf{M}_k^d = 0$ being always true. We denote next the interphase momentum as $\mathbf{M}_k^{I_n}$ and the drift momentum as $\mathbf{M}_k^{d_0}$. Following Eqn. (11), the interphase momentum is given by

$$\begin{aligned} \mathbf{M}_k^{I_n} &= \frac{\alpha_k \rho_k}{\rho_m} \mathbf{M}_m^{\{p,v,g,d_0,a\}} - \mathbf{M}_k^{\{p,v,g,a\}} \\ &= \alpha_k (\rho_k - \rho_m) \left(\frac{D}{Dt} \mathbf{v}_m - \mathbf{g} \right) + \left(\alpha_k \mathbf{M}_m^{\{p,v,a\}} - \mathbf{M}_k^{\{p,v,a\}} \right). \end{aligned} \tag{12}$$

The no-drift interphase momentum can be interpreted as follows. The first term on the right-hand side of Eqn. (12) is the offset created to reduce inertia-related drift, such as sand in water separated by gravity. This term counteracts the acceleration causing sand and water to stick together. The second term addresses the differences in momentum sources between phase-level and mixture-level, caused by pressure, viscosity, and advection.

Free-drift scenario. In this scenario, all phases move independently without influencing each other, so the interphase momentum is always zero. The free-drift momentum source can be derived from Eqn. (11) as

$$\begin{aligned} \mathbf{M}_k^{d_f} &= \mathbf{M}_k^{\{p,v,g,a\}} - \frac{\alpha_k \rho_k}{\rho_m} \mathbf{M}_m^{\{p,v,g,d_f,a\}} \\ &= \alpha_k (\rho_m - \rho_k) \left(\frac{D}{Dt} \mathbf{v}_m - \mathbf{g} \right) - \left(\alpha_k \mathbf{M}_m^{\{p,v,a\}} - \mathbf{M}_k^{\{p,v,a\}} \right). \end{aligned} \tag{13}$$

Analytic interphase momentum expression. An interesting relation between the no-drift and free-drift scenarios emerges from Equations 12 and 13 as $\mathbf{M}_k^{d_f} = -\mathbf{M}_k^{I_n}$. This tells that the stronger the interphase momentum source, the harder it is to generate drift velocity. Based on this observation, we further assume that $\mathbf{M}_k^I = -\mathbf{M}_k^d$, which perfectly satisfies Eqn. (11). We next analytically express the interphase momentum source using Eqn. (12) as

$$\mathbf{M}_k^I = C_d \mathbf{M}_k^{I_n}, \tag{14}$$

where $C_d \in [0, 1]$ models the drift amount: $C_d = 0$ yields free drift and $C_d = 1$ yields no drift. Similar forms of interphase momentum sources have been earlier used to compute the change rate of drift velocity [Jiang et al. 2020; Ren et al. 2014]. Yet, while crucial for producing a divergence-free mixture velocity field, a concise expression without *any* approximation during derivation has been absent. In contrast, we provide phase-level operations that show how drift conditions can be altered using an interphase momentum source without violating mixture-level incompressibility.

4.2 Phase-mixture Momentum Mapping

We next compute each phase's motion changes by each momentum source via Eqn. (14). Four main momentum sources influence $\mathbf{M}_k^I$ according to Eqn. (12), while $\frac{D}{Dt} \mathbf{v}_m$ can be affected by all five (see Eqn. (10)).

When computing $\mathbf{M}_k^I$, we ignore advection and drift momentum sources. This not only eliminates the computational demand for $\nabla \mathbf{v}_k^a$ and the time derivative of $\mathbf{v}_{Vk}$ but also crucially allows Eqn. (10) to be treated as a single-phase fluid. This strategy enables the use of an implicit pressure solver [Bender and Koschier 2017] throughout the simulation, facilitating larger time steps compared to the explicit solver-based method [Jiang and Lan 2021], and yielding a speedup of roughly 2–3 times in total. We next show how viscosity and pressure momentum sources influence each phase's motion.

Since viscosity affects each phase, the change rate of phase velocity generated by viscosity can be derived from Eqn. (9) as

$$\begin{aligned} \frac{D^v}{Dt} \mathbf{v}_k &= \frac{1}{\alpha_k \rho_k} \left(\mathbf{M}_k^v + \mathbf{M}_k^{I_v} \right) \\ &= \frac{C_d \alpha_k (\rho_k - \rho_m) \frac{D^v}{Dt} \mathbf{v}_m + C_d \alpha_k \mathbf{M}_m^v + (1 - C_d) \mathbf{M}_k^v}{\alpha_k \rho_k} \\ &= C_d \frac{\mathbf{M}_m^v}{\rho_m} + (1 - C_d) \frac{\mathbf{M}_k^v}{\alpha_k \rho_k}, \end{aligned} \tag{15}$$

where $\mathbf{M}_k^{I_v} = C_d \alpha_k (\rho_k - \rho_m) \frac{D^v}{Dt} \mathbf{v}_m + C_d \alpha_k \mathbf{M}_m^v$ is the viscosity-related part of the interphase momentum source according to Eqn. (14), and $\frac{D^v}{Dt} \mathbf{v}_m = \mathbf{M}_m^v / \rho_m$ is the viscosity-induced change rate of mixture velocity from Eqn. (10).

Similarly, the change rate of phase velocity produced by pressure is

$$\frac{D^p}{Dt}\mathbf{v}_k = C_d \frac{\mathbf{M}_m^p}{\rho_m} + (1-C_d)\,\frac{\mathbf{M}_k^p}{\alpha_k\rho_k} = \frac{\mathbf{M}_m^p}{\rho_m}\left(C_d + (1-C_d)\,\frac{\rho_m}{\rho_k}\right), \quad (16)$$

where $\mathbf{M}_k^p = -\alpha_k \nabla p_k$. Again, we follow $p_k = p_m$ for all phases [Manninen et al. 1996]. Eqn. (16) is our proposed phase-mixture momentum mapping mechanism which calculates the movement of phase flow using mixture momentum sources.

Previous mixture models [Jiang et al. 2020; Ren et al. 2022, 2014] focus on the solvable momentum equation, considering interphase momentum as a consequence. This creates a *causal* relation between momentum, mixture velocity change rate, and drift velocity (Eqns. (14) and (12)). The change rate of mixture velocity, $\frac{D}{Dt}\mathbf{v}_m$, must occur first in a time step to 'generate' the interphase momentum source that next alters drift velocity. This approach violates the relation between phase velocity and mixture velocity in Eqn. (2), making it impossible to obtain a stable mixture-level fluid description without LEA. Although Jiang *et al.* [2021] proposed to compute mixture-level fluid motion from phase-level to abandon LEA, the introduced artificial momentum source term could result in weak compressibility for the mixture flow; also, the volume fraction of fluid particles needs to be re-normalized at each time step.

In contrast, our method *simultaneously* solves the velocity change rate for all fluid phases and the mixture flow. According to Eqn. (16), the phase velocity change rate can be implicitly expressed through the pressure component of the mixture momentum source. The term $-\frac{1}{\rho_m}\nabla p_m$ is solvable using any standard fluid simulation approach. This substitution in Eqn. (15) and (16) unifies phase, drift, and mixture velocities, ensuring natural mass conservation and consistent volume fraction summation for all particles.

4.3 Mixture Viscosity Model

The standard mixture model does not specify how phases interact with each other. Since the mixture model uses the LEA, viscosity is assumed to bind phases together, reverting to the mixture velocity. Jiang*et al.* [2021] designed their artificial interphase momentum source with this idea in mind. In contrast, we modify the viscous momentum source from a single-phase viscosity computation procedure to a viscosity computation between the phase and the mixture. For the entire computational domain Ω, this is given by

$$\widetilde{\mathbf{M}}_k^v(\mathbf{x}) = \alpha_k(\mathbf{x})\,\mu_k\nabla_\mathbf{x}^2\left(\mathbf{v}_m + \int_\Omega \mathbf{v}_{Vk}(\mathbf{x})\,\delta(\mathbf{x})\right), \quad (17)$$

where μ is the dynamic viscosity coefficient and δ denotes the Dirac delta function. This derivation is consistent with the standard mixture model, telling that a single fluid phase interacts equally with all other phases. Moreover, this form conserves momentum for each fluid parcel while maintaining $\sum_k \widetilde{\mathbf{M}}_k^v = \mathbf{M}_m^v$ if the dynamic viscosities of all phases are the same.

5 IMPLEMENTATION

We integrate our implicit mixture model (Sec. 4) into the DFSPH approach [Bender and Koschier 2017] as detailed next (see also Alg. 1). We refer to SPH overviews [Ihmsen et al. 2014; Koschier et al. 2019] for more details about smoothed particle hydrodynamics approaches and to Ren et al. [2014] for the volume fraction implementations.

```
Algorithm 1: Mixture model with DFSPH
A. Preparation
    1. Calculate mixture velocity $\mathbf{v}_m$                          Eqn. (2)
    2. Calculate drift velocity $\mathbf{v}_{Vk}$                          Eqn. (3)
    3. Update particle rest density $\rho_m$                           Eqn. (4)
B. DFSPH divergence-free solver
    4. Calculate $\mathbf{M}_m^{p_{\text{div}}}$ using DFSPH
    5. Update phase velocity (from $\mathbf{M}_k^p$) $\mathbf{v}_k \mathrel{+}= \Delta t \frac{D^{p_{\text{div}}}}{Dt}\mathbf{v}_k$
  Eqn. (16)
    6. Repeat Step A.1 to update $\mathbf{v}_m$
C. Advection
    7. Update phase velocity (from $\mathbf{M}_k^v$)
        $\mathbf{v}_k \mathrel{+}= \Delta t \frac{D^v}{Dt}\mathbf{v}_k$ with $\mathbf{M}_k^v$ as $\tilde{\mathbf{M}}_k^v$            Eqns. (15), (17)
    8. Update phase velocity (from $\mathbf{M}_k^g$) $\mathbf{v}_k \mathrel{+}= \Delta t \mathbf{g}$
    9. Repeat Step A.1 to update $\mathbf{v}_m$
D. DFSPH incompressible solver
    10. Calculate $\mathbf{M}_m^{p_{\text{inc}}}$ using DFSPH
    11. Update phase velocity (from $\mathbf{M}_k^p$)
        $\mathbf{v}_k \mathrel{+}= \Delta t \frac{D^{p_{\text{inc}}}}{Dt}\mathbf{v}_k$                               Eqn. (16)
    12. Repeat Steps A.1 and A.2 to update $\mathbf{v}_m$ and $\mathbf{v}_{Vk}$
    13. Update particle position $\mathbf{x} \mathrel{+}= \Delta t \mathbf{v}_m$
E. Phase transfer
    14. Update volume fraction
        $\alpha_k \mathrel{+}= \Delta t \frac{D}{Dt}\alpha_k + \Delta t \nabla^2 (D_m \alpha_k)$          Eqns. (6), (20)
```

5.1 Volume Incompressible SPH

We replace using density to constrain the incompressible state with the concept of volume incompressibility [Band et al. 2018] to adapt SPH solvers for fluids with nonuniform density fields via

$$\psi_i = \sum_j V_j^0 W_{ij}, \quad (18)$$

where ψ represents the compression ratio of particle i and j are all its neighbors within a supporting radius h (we use the same i, j notation in all our implementation next); V_j^0 is the rest volume; and for the normalized smoothing kernel $W_{ij} = W(\|\mathbf{x}_i - \mathbf{x}_j\|, h)$, we used a cubic spline as outlined in [Ihmsen et al. 2014]. Pressure solvers are employed to maintain $\psi < 1$.

5.2 Advection-projection Procedure

Following implicit SPH, we divide a simulation time-step into two stages: advection and projection, as follows.

Advection. During advection, gravity and viscosity forces are initially used to alter the velocity of each fluid particle, yielding a compressible fluid field. Our advection for each phase (Alg. 1, step C) computes our refined viscosity momentum source (Eqn. 17) using an artificial SPH Laplacian approximation [Ihmsen et al. 2014] as

$$\widetilde{\mathbf{M}}_k^v(\mathbf{x}_i) = 2(d+2)\alpha_{k_i}\mu_k \sum_j V_j^0 \frac{\left(\mathbf{v}_{k_i} - \mathbf{v}_{m_j}\right)\cdot \mathbf{x}_{ij}}{\|\mathbf{x}_i - \mathbf{x}_j\|^2 + 0.01h^2}\nabla W_{ij}, \quad (19)$$

where d denotes the dimension ($d = 2$ or $d = 3$) of the scenario and h is a regularization factor.

Projection. During projection, pressure forces are implicitly derived to restore the fluid to an incompressible state. Following DFSPH [Bender and Koschier 2017], we do two projection steps, before and after advection (Alg. 1, steps B and D). Step B uses pressure forces to make the velocity field divergence-free. We denote the induced mixture-level momentum source and change rate of phase velocity as $\sum_k \mathbf{M}_k^{p_{\text{div}}}$ and $\frac{D^{p_{\text{div}}}}{Dt}\mathbf{v}_k$, respectively. Step D predicts the position of the fluid particle at the next time step to make the fluid incompressible. We denote the momentum and change rate of phase velocity in step D as $\sum_k \mathbf{M}_k^{p_{\text{inc}}}$ and $\frac{D^{p_{\text{inc}}}}{Dt}\mathbf{v}_k$, respectively.

5.3 Phase Transfer

Two factors cause changes in the particles' volume fraction (Alg. 1, step E). The first one is drift velocity. We use the SPH approximation [Jiang et al. 2020] to compute Eqn. (6) as

$$\frac{D\alpha_{k_i}}{Dt} = -\sum_j V_j^0 \left(\alpha_{k_i}\mathbf{v}_{Vk_i} + \alpha_{k_j}\mathbf{v}_{Vk_j}\right)\cdot \nabla W_{ij}. \tag{20}$$

The second factor is the diffusion term. Similar to the previous divergence-free mixture model [Jiang et al. 2020], we use a diffusion coefficient D_m to control the diffusion effect as

$$\nabla^2\left(D_m\alpha_{k_i}\right) = D_m\sum_j V_j^0\left(\alpha_{k_i}-\alpha_{k_j}\right)\frac{(\mathbf{x}_i-\mathbf{x}_j)\cdot\nabla W_{ij}}{\|\mathbf{x}_i-\mathbf{x}_j\|^2+0.01h^2}. \tag{21}$$

6 EXPERIMENTS

We evaluate our method under different scenarios both analytically and visually (see Tab. 2) and compare it with a state-of-the-art dynamic mixture model [Jiang and Lan 2021]. We implemented our physics simulation framework using Taichi [Hu et al. 2019] and rendered results with Blender's Cycles engine. We ran our experiments on an NVIDIA 3090 Tensor Core GPU for both simulation and rendering. Our implementation is available as open source for replicability [Xu 2023].

Table 2. Experiment configurations.

Scene	Figure(s)	Particle count	Particle size [m]	Drag coefficient	Diffusion coefficient
Tea	Fig. 1	499K	0.03	0.5	0.05
Collide	Fig. 3	474K	0.0055	C_d: 0, 0.3, 0.7, 1 k_d: 0, 3, 7, 10	0
Hourglass	Fig. 4	350K	0.03	C_d: 0.29; k_d: 5	0.05
Rotate	Fig. 5	473K	0.0075	C_d: 0, 0.3, 0.7, 1 k_d: 0, 3, 7, 10	0
Ink drop	Fig. 6	509K	0.05	C_d: 1, 0.6; k_d: 10	0
Phase separation	Fig. 7, Fig. 8	49K	0.045	C_d: 0.61, 0.29, 0 k_d: 10, 5, 2.53	0
Cocktail 2	Fig. 9	1.09M	0.05	0.94	0.6
Propeller	Fig. 10	555K	0.035	C_d: 0.95	0.5
Cocktail 1	Fig. 11	1.09M	0.05	0.6	0.1

6.1 Performance Analysis

6.1.1 Momentum Conservation. We executed two experiments to evaluate the conservation of linear and angular momentum. In each case, fluid particles were initialized with an equal mixture of two fluid phases, shown in red and blue, maintaining a density ratio of $1 : 2$ (see Figs. 3 and 5). We excluded any external momentum sources, *e.g.*, gravity. Our method was tested using drift coefficients $C_d = 0.0$ (free-drift), $0.3, 0.7$, and 1.0 (no drift, DFSPH-only). We also assessed the mixture model proposed by Jiang *et al.* using drift coefficients $k_d = 0.0, 3.0, 7.0$, and 10.0. Although [2021] does not mention free-drift, we use $k_d = 0.0$ to denote the most pronounced effects, which taper off with increasing values, in tandem with WCSPH-only. We excluded viscosity in these tests, barring two sets in the Collide experiment. Detailed results can be also seen in the accompanying video.

Collide: The first experiment aimed to test the conservation of linear momentum. A smaller fluid cube, initially having a velocity of $1m/s$ moving to the right, was set to collide with a larger, stationary fluid cube (Fig. 3). As observed during the collision, the denser blue phase began to separate from the contact surface, a result of its heightened inertia. In contrast, the lighter red phase, with its heightened sensitivity to the collision dynamics, quickly congregated in the spherical region of the fluid cubes. Over time, the phase separation between the red and blue fluids became increasingly distinct.

The chart in Fig. 3(c) shows that, in the absence of a multiphase solver (negating drift effects), both DFSPH and WCSPH adeptly conserve momentum. When our implicit mixture model is activated at $C_d < 1$, the phase separation intensifies, yet linear momentum remains largely unaltered over time. In contrast, Jiang *et al.*'s model introduces momentum discrepancies, as most trials using this approach exhibited deviations from the benchmark, particularly over extended simulation durations.

Rotate: Our second experiment focused on the conservation of angular momentum and was set in a round pool containing a swirling fluid ring (Fig. 5). The fluid ring initially displayed an irrational counterclockwise velocity field. Here, each particle's velocity was tangential to the center, with its magnitude being inversely proportional to its distance to the center. As the experiment progressed, the centrifugal effect manifested itself. The denser blue fluid phase gradually settled in the peripheral regions, while the center became dominated by the pure red phase.

Fig. 5(c) shows that DFSPH and our method both experience similar angular momentum reduction over time, reflecting the simulation's inherent dissipative nature without external energy. WCSPH dissipates faster than DFSPH and could destabilize without viscosity. The dynamic mixture model of [2021] intensifies this, hindering optimal conservation. In our model, as C_d nears zero, angular momentum attenuation slightly surpasses the DFSPH-only case. We believe this is due to updating the volume fraction at step E of Alg. 1, causing minor errors during phase exchanges among particles, especially with notable phase separation. However, this does not adversely impact visuals or stability.

6.1.2 Simulation Efficiency. We assessed the efficiency of our method against the existing approach using performance statistics

Table 3. Time statistics for the collide experiment.

		time consumed for one time step [s]	time consumed for one second of animation [s]
Jiang *et al.*	WCSPH	1.41×10^{-3}	51.41
	Neighbor search	2.18×10^{-3}	79.12
	Multiphase	2.49×10^{-2}	905.33
Our method	DFSPH	1.75×10^{-2}	63.65
	Neighbor search	2.67×10^{-3}	9.70
	Multiphase	2.98×10^{-2}	108.51

from the collide experiment in Fig. 3. For this experiment, time consumption was measured from two vantage points: The average time consumed per segment of a single simulation loop (referred to as the "time step"), and The average time needed to generate 1 second of simulation animation (which involves summing the time cost of each time step up to a second). This distinction is pivotal, given that DFSPH, an implicit fluid solver, permits a considerably larger time step than the explicit WCSPH solver. As a result, DFSPH requires significantly fewer simulation loops than WCSPH to produce an equivalent duration of simulation. Notably, although each DFSPH loop might individually take longer than a WCSPH loop, fewer loops overall lead to time savings. Tab. 3 shows the average time taken to generate 1 loop/second of animation for our method and Jiang *et al.*’s model. Our approach consistently required significantly less time. This efficiency arises because DFSPH supports a considerably larger time step per simulation loop compared to WCSPH. With a particle size of $0.0055m$, our method’s maximum time step is approximately $2.75 \times 10^{-4}s$ and about $2.75 \times 10^{-5}s$ for WCSPH. Our multiphase solver requires a marginal 2% more time per loop than Jiang *et al.*’s but our substantial speedup due to the time step difference makes this extra cost negligible.

6.1.3 Mass Conservation. **Two Phase Separation**: In this experiment, a fluid bulk with two thoroughly mixed phases is left to separate under gravity (Fig. 8(a)). The volume fraction of the two phases is 1 : 1 and the density ratio is 1 : 2 (red : blue). This experiment tests the numerical performance of our method, the implicit mixture model [Jiang and Lan 2021], and ours with LEA for two-phase separation under gravity. The experiment also maps the relation between the drag k_d and C_d.

All three methods are capable of simulating this scenario. Yet, our method with LEA separates very slowly. Only a thin separation layer is noticeable after 30 seconds, suggesting that the mixture model suboptimally handles this scenario (see also the accompanying video). Our method and the method of Jiang *et al.* separate at similar speeds, with $C_d = 0$, $k_d = 2.53$ being the quickest; followed by $C_d = 0.29$, $k_d = 5$; and $C_d = 0.61$, $k_d = 10$ being the slowest. We compare these three coefficient sets because $k_d = 10$ is recommended in the dynamic mixture model [Jiang and Lan 2021] and $C_d = 0$ models free drift in our method.

Figure 8(c)–(e) shows the phase velocity on the y-axis ($v_{k,y}$) with $C_d = 0$, $k_d = 2.53$ under the status of Fig. 8(b). The method of Jiang *et al.* retains some phase velocity at the far ends of the y-axis, where the two phases are separated and no phase velocity should occur. Our method avoids this issue and has the *largest* phase velocity near the separated and mixed regions, where the gradient of volume fraction is the greatest.

Fig. 7 shows the variation of the incompressible state of the mixture (Fig. 7(a)), volume fraction of each phase (Fig. 7(b)), and phase-level kinetic energy measured through the phase velocity (Fig. 7(c)) over time. The dynamic mixture model [Jiang and Lan 2021] conserves neither incompressibility nor mass of each phase as rigorously as our method since they use a weakly-compressible pressure solver and regularize the volume fraction at each time step. Moreover, the kinetic energy of our method always manages to reach the same peak regardless of the speed of the separation process. This means the transformation of energy from gravitational potential to kinetic is unconditionally conserved, which is not the case for [Jiang and Lan 2021].

6.2 Comparisons of Effects

Ink drop: A drop of red ink (70% ink phase, 30% water phase) follows a parabolic trajectory into the water (Fig. 6). The diffusion coefficient is set to 0. Results show that our method’s drag coefficient can produce more substantial and consistent turbulence effects due to phase interactions. In contrast, in the dynamic mixture model [Jiang and Lan 2021], the drag force reduces this effect.

Hourglass: A blue phase (10 times heavier than the transparent one) flows through a narrow neck under the influence of both diffusion and interphase momenta. As Fig. 4 shows, our method better captures the convection vortex. Also, the momentum does not dissipate during convection, as shown by the lively bubbles in the third column of Fig. 4.

6.3 Effectiveness under Complex Scenarios

Propeller: We simulate the foam generated by a propeller spinning under water (Fig. 10). The foam phase has half the density of the water. Moreover, a strong interphase momentum is applied to make the two phases challenging to separate. As Fig. 10 shows, the foam becomes more robust and more agitated as the propeller spins faster. The centrifugal effect can also be seen with the foam being dragged by the blades, demonstrating our method’s versatility.

Cocktail - phase separation: Five phases of a cocktail with a density ratio of 1 : 2 : 4 : 8 : 16 are initially evenly mixed in a glass (Fig. 11). Then, the drag coefficient is applied to ease the separation of the five phases. The significant density ratio makes the separation quite dramatic. Our method can handle this process with ease.

Cocktail - drag dominant: Building upon the previous settings, we show in Fig. 9 an extra strong coupling force and diffusion effect between phases. Upon stirring, the mixture starts to separate as the significant density variation creates a high drag force causing the phases to separate from one another (see also the accompanying video for detailed differences between phase separation and drag-dominant experiments).

Tea bag brewing - diffusion dominant: In contrast to stirring-induced separation, Fig. 1 presents an experiment of brewing tea using a tea bag. The tea diffusion into water is obtained by assigning volume fractions to the solid and enabling diffusion between solid and liquid. After stirring with a rod, tea and water become evenly mixed. The density ratio is small enough (1.05) to make the diffusion effect more dominant than the separation.

7 CONCLUSION

We have presented an implicit mixture model for incompressible fluid simulation. Our model offers a unified physical description between phase-level and mixture-level fluid fields by combining an interphase momentum source and a phase-momentum mapping mechanism with a mixture viscosity model. This enables mass conservation for each phase and volume conservation for the mixture. Our unified model allows for easy integration with existing fluid solvers with minimal coding efforts.

Several aspects warrant further exploration to achieve more accurate simulations. The phase transfer occurring at each time step also introduces momentum from one fluid parcel to another. This effect could produce intriguing visual results when the drift velocity is substantial. In future work, we plan to delve deeper into studying this mechanism and integrate it into our framework.

ACKNOWLEDGMENTS

This research was funded by National Natural Science Foundation of China (No.62376025), National Key Research and Development Program of China (No.2022ZD0118001), Guangdong Basic and Applied Basic Research Foundation (No.2023A1515030177), and the Fundamental Research Funds for the Central Universities (No.QNXM20220043).

REFERENCES

Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. 2012. Versatile Rigid-Fluid Coupling for Incompressible SPH. *ACM Trans. Graph.* 31, 4, Article 62 (jul 2012), 8 pages.

Iván Alduán, Angel Tena, and Miguel A. Otaduy. 2017. DYVERSO: A Versatile Multi-Phase Position-Based Fluids Solution for VFX. *Computer Graphics Forum* 36, 8 (2017), 32–44.

Ryoichi Ando, Nils Thuerey, and Chris Wojtan. 2015. A Stream Function Solver for Liquid Simulations. *ACM Trans. Graph.* 34, 4, Article 53 (jul 2015), 9 pages.

Stefan Band, Christoph Gissler, Markus Ihmsen, Jens Cornelis, Andreas Peer, and Matthias Teschner. 2018. Pressure Boundaries for Implicit Incompressible SPH. *ACM Trans. Graph.* 37, 2, Article 14 (feb 2018), 11 pages.

Kai Bao, Xiaolong Wu, Hui Zhang, and Enhua Wu. 2010. Volume fraction based miscible and immiscible fluid animation. *Computer Animation and Virtual Worlds* 21, 3–4 (2010), 401–410.

Jan Bender and Dan Koschier. 2017. Divergence-Free SPH for Incompressible and Viscous Fluids. *IEEE Transactions on Visualization and Computer Graphics* 23, 3 (2017), 1193–1206.

Landon Boyd and Robert Bridson. 2012. MultiFLIP for energetic two-phase fluid simulation. *ACM Transactions on Graphics* 31, 2 (2012), 1–12.

Christopher E Brennen. 2005. *Fundamentals of multiphase flow.* Cambridge University Press.

Robert Bridson. 2015. *Fluid simulation for computer graphics.* CRC Press.

Simon Clavet, Philippe Beaudoin, and Pierre Poulin. 2005. Particle-Based Viscoelastic Fluid Simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '05)*. Association for Computing Machinery, New York, NY, USA, 219–228.

Gilles Daviet and Florence Bertails-Descoubes. 2017. Simulation of Drucker–Prager granular flows inside Newtonian fluids. (Feb. 2017). https://inria.hal.science/hal-01458951 working paper or preprint.

Yun (Raymond) Fei, Christopher Batty, Eitan Grinspun, and Changxi Zheng. 2018. A Multi-Scale Model for Simulating Liquid-Fabric Interactions. *ACM Trans. Graph.* 37, 4, Article 51 (jul 2018), 16 pages. https://doi.org/10.1145/3197517.3201392

Ming Gao, Andre Pradhana, Xuchen Han, Qi Guo, Grant Kot, Eftychios Sifakis, and Chenfanfu Jiang. 2018. Animating Fluid Sediment Mixture in Particle-Laden Flows. *ACM Trans. Graph.* 37, 4, Article 149 (jul 2018), 11 pages.

Yulong Guo, Xiaopei Liu, and Xuemiao Xu. 2017. A Unified Detail-Preserving Liquid Simulation by Two-Phase Lattice Boltzmann Modeling. *IEEE Transactions on Visualization and Computer Graphics* 23, 5 (2017), 1479–1491.

Jeong-Mo Hong and Chang-Hun Kim. 2005. Discontinuous Fluids. *ACM Trans. Graph.* 24, 3 (jul 2005), 915–920.

Yuanming Hu, Tzu-Mao Li, Luke Anderson, Jonathan Ragan-Kelley, and Frédo Durand. 2019. Taichi: A Language for High-Performance Computation on Spatially Sparse Data Structures. 38, 6, Article 201 (nov 2019), 16 pages.

Markus Ihmsen, Jens Orthmann, Barbara Solenthaler, Andreas Kolb, and Matthias Teschner. 2014. SPH Fluids in Computer Graphics. *Eurographics 2014: State of the art reports* (2014).

Y. Jiang and Y. Lan. 2021. A Dynamic Mixture Model for Non-equilibrium Multiphase Fluids. *Computer Graphics Forum* 40, 7 (2021), 85–95.

Y. Jiang, C. Li, S. Deng, and S. M. Hu. 2020. A Divergence-free Mixture Model for Multiphase Fluids. *Computer Graphics Forum* 39, 8 (2020), 69–77.

Nahyup Kang, Jinho Park, Junyong Noh, and Sung Yong Shin. 2010. A Hybrid Approach to Multiple Fluid Simulation using Volume Fractions. *Computer Graphics Forum* 29, 2, 685–694.

Byungmoon Kim. 2010. Multi-Phase Fluid Simulations Using Regional Level Sets. *ACM Trans. Graph.* 29, 6, Article 175 (dec 2010), 8 pages.

Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. 2019. Smoothed Particle Hydrodynamics Techniques for the Physics Based Simulation of Fluids and Solids. *Eurographics 2019 - Tutorials.*

Wei Li, Daoming Liu, Mathieu Desbrun, Jin Huang, and Xiaopei Liu. 2021. Kinetic-Based Multiphase Flow Simulation. *IEEE Transactions on Visualization and Computer Graphics* 27, 7 (2021), 3318–3334.

Shiguang Liu, Qiguang Liu, and Qunsheng Peng. 2011. Realistic Simulation of Mixing Fluids. *Vis. Comput.* 27, 3 (mar 2011), 241–248.

Frank Losasso, Tamar Shinar, Andrew Selle, and Ronald Fedkiw. 2006. Multiple Interacting Liquids. *ACM Trans. Graph.* 25, 3 (jul 2006), 812–819.

Mikko Manninen, Veikko Taivassalo, and Sirpa Kallio. 1996. On the mixture model for multiphase flow. *VTT Publications* 288 (1996), 3–67.

Hai Mao and Yee-Hong Yang. 2006. Particle-Based Immiscible Fluid-Fluid Collision. In *Proceedings of Graphics Interface 2006* (Quebec, Canada) *(GI '06)*. Canadian Information Processing Society, CAN, 49–55.

Matthias Müller, Barbara Solenthaler, Richard Keiser, and Markus Gross. 2005. Particle-Based Fluid-Fluid Interaction. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '05)*. Association for Computing Machinery, New York, NY, USA, 237–244.

Simon Premžoe, Tolga Tasdizen, James Bigler, Aaron Lefohn, and Ross T. Whitaker. 2003. Particle-Based Simulation of Fluids. *Computer Graphics Forum* 22, 3 (2003), 401–410.

Bo Ren, Wei He, Chen-Feng Li, and Xu Chen. 2022. Incompressibility Enforcement for Multiple-Fluid SPH Using Deformation Gradient. *IEEE Transactions on Visualization and Computer Graphics* 28, 10 (2022), 3417–3427.

Bo Ren, Chen-Feng Li, Xiao Yan, Ming Lin, Javier Bonet, and Shi-Min Hu. 2014. Multiple-Fluid SPH Simulation Using a Mixture Model. *ACM Transactions on Graphics* 33 (09 2014), 1–11.

Bo Ren, Xu-Yun Yang, Ming C Lin, Nils Thuerey, Matthias Teschner, and Chenfeng Li. 2018. Visual simulation of multiple fluids in computer graphics: A state-of-the-art report. *Journal of Computer Science and Technology* 33 (2018), 431–451.

B. Solenthaler and R. Pajarola. 2008. Density Contrast SPH Interfaces. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Dublin, Ireland) *(SCA '08)*. Eurographics Association, Goslar, DEU, 211–218.

Andre Pradhana Tampubolon, Theodore Gast, Gergely Klár, Chuyuan Fu, Joseph Teran, Chenfanfu Jiang, and Ken Museth. 2017. Multi-Species Simulation of Porous Sand and Water Mixtures. *ACM Trans. Graph.* 36, 4, Article 105 (jul 2017), 11 pages.

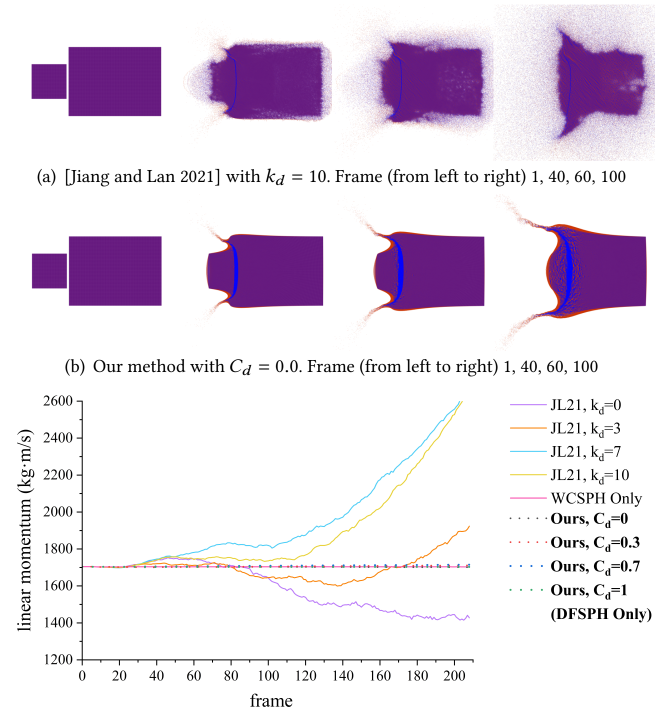
Joel Wretborn, Sean Flynn, and Alexey Stomakhin. 2022. Guided Bubbles and Wet Foam for Realistic Whitewater Simulation. *ACM Trans. Graph.* 41, 4, Article 117 (jul 2022), 16 pages. https://doi.org/10.1145/3528223.3530059

Yanrui Xu. 2023. Implementation of multiphase flow simulation. https://github.com/sakamotoyan/TiSPH_multiphase.

Xiao Yan, Yun-Tao Jiang, Chen-Feng Li, Ralph R. Martin, and Shi-Min Hu. 2016. Multiphase SPH Simulation for Interactive Fluids and Solids. *ACM Trans. Graph.* 35, 4, Article 79 (jul 2016), 11 pages.

Tao Yang, Jian Chang, Ming C. Lin, Ralph R. Martin, Jian J. Zhang, and Shi-Min Hu. 2017. A Unified Particle System Framework for Multi-Phase, Multi-Material Visual Simulations. *ACM Trans. Graph.* 36, 6, Article 224 (nov 2017), 13 pages.

Tao Yang, Jian Chang, Bo Ren, Ming C. Lin, Jian Jun Zhang, and Shi-Min Hu. 2015. Fast Multiple-Fluid Simulation Using Helmholtz Free Energy. *ACM Trans. Graph.* 34, 6, Article 201 (nov 2015), 11 pages.

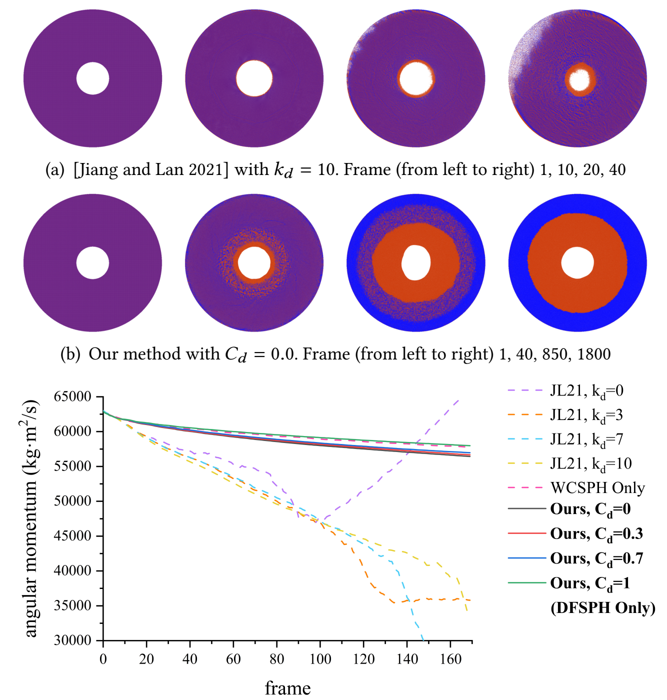


(a) [Jiang and Lan 2021] with $k_d = 10$. Frame (from left to right) 1, 40, 60, 100

(b) Our method with $C_d = 0.0$. Frame (from left to right) 1, 40, 60, 100

(c) Comparison of linear momentum in the two blocks collision experiment between our method and the dynamics mixture model using various C_d and k_d coefficient values.

Fig. 3. A collision experiment shows the pressure-induced phase separation abilities of Jiang *et al.*'s method (a) and our method (b). Our approach exhibits superior stability, especially in the absence of gravity and viscosity, by consistently maintaining linear momentum over time (c). See Secs. 6.1.1 and 6.1.2.

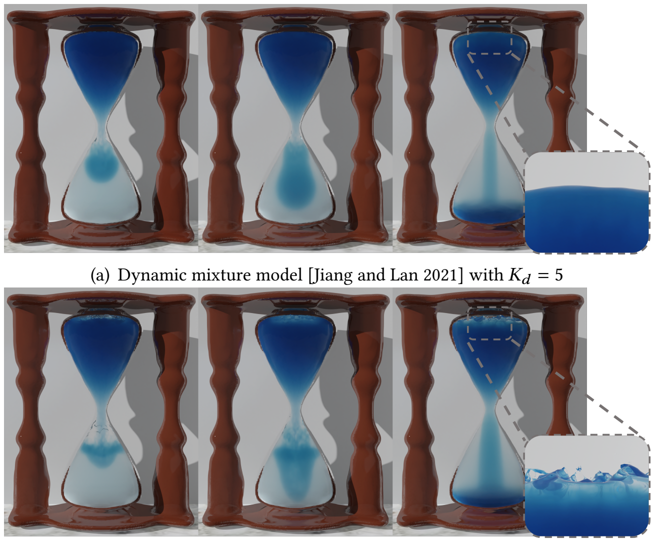


(a) [Jiang and Lan 2021] with $k_d = 10$. Frame (from left to right) 1, 10, 20, 40

(b) Our method with $C_d = 0.0$. Frame (from left to right) 1, 40, 850, 1800

(c) Comparison of angular momentum in the fluid rotation experiment between our method and the dynamics mixture model using various C_d and k_d coefficient values.

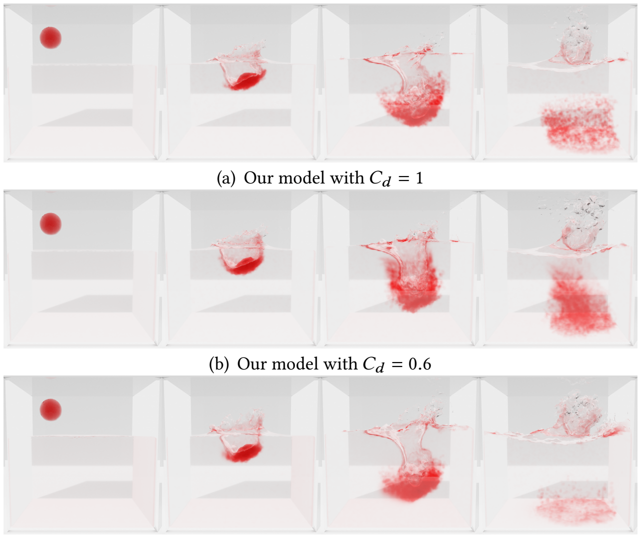
Fig. 5. Under the centrifugal effect, the rotated fluid separates into two distinct phases. The dynamic mixture model of [Jiang and Lan 2021] struggles to maintain stability without the constraints of gravity and viscosity, leading to unreasonable phase velocities (a). Conversely, our method (b) ensures a stable conservation of angular momentum throughout the simulation (c). See Sec. 6.1.1.



(a) Dynamic mixture model [Jiang and Lan 2021] with $K_d = 5$

(b) Our method with $C_d = 0.29$

Fig. 4. The blue phase in the hourglass is convected with the transparent one causing vortexes and bubbling effects. See Sec. 6.2.

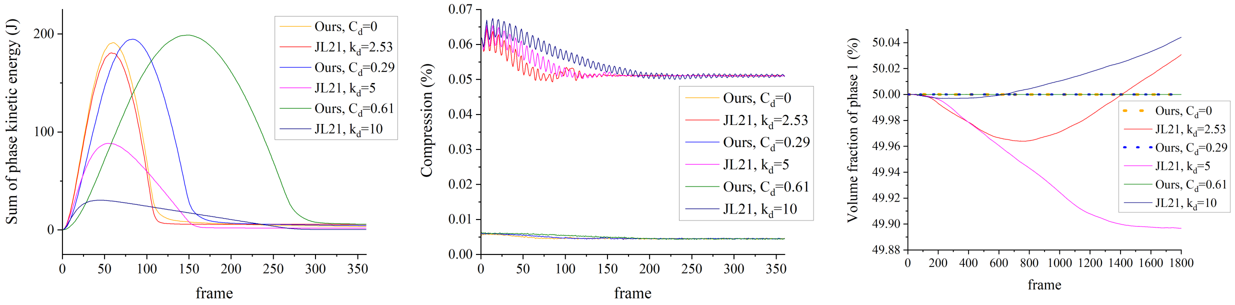


(a) Our model with $C_d = 1$

(b) Our model with $C_d = 0.6$

(c) [Jiang and Lan 2021] with $k_d = 10$

Fig. 6. Red ink drop falls into water along a parabolic trajectory. See Sec. 6.2.

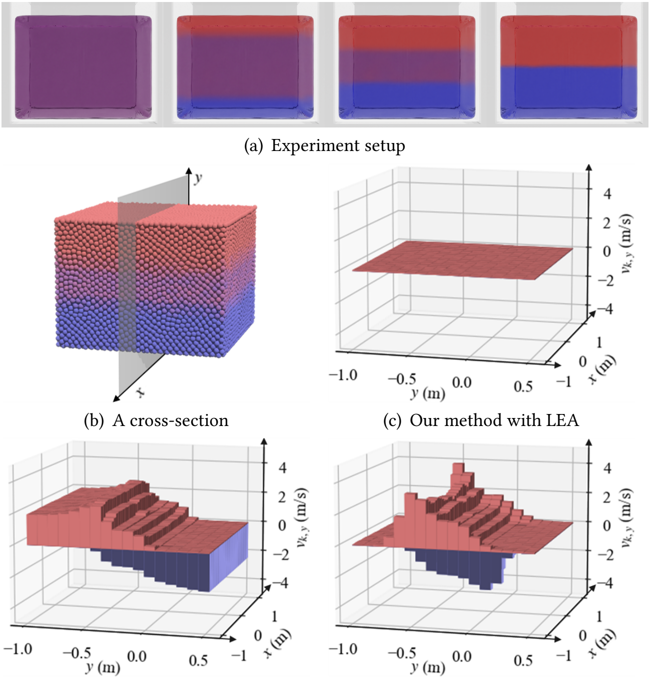


(a) Phase-level kinetic energy

(b) Compression

(c) Volume fraction

Fig. 7. Comparison of phase kinetic energy, compression, and volume fraction in the two-phase separation experiment (Fig. 8) between our method and the dynamic mixture model of [Jiang and Lan 2021] using various C_d and k_d coefficient values.



(a) Experiment setup

(b) A cross-section

(c) Our method with LEA

(d) Dynamic mixture model [2021]

(e) Our method

Fig. 8. Our two-phase separation experiment. (a) shows the experiment setup, (b) shows a visualization of phase velocity (c)–(e) using different methods. See Sec. 6.1.3.

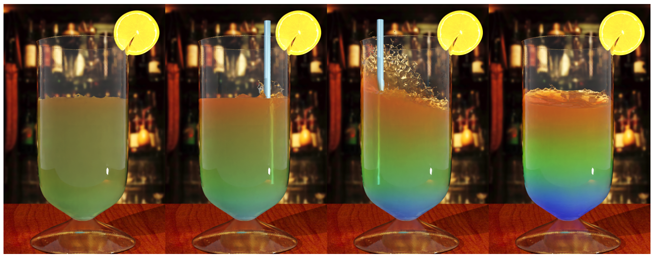


Fig. 9. A tightly coupled cocktail gets separated after a stir. See Sec. 6.3

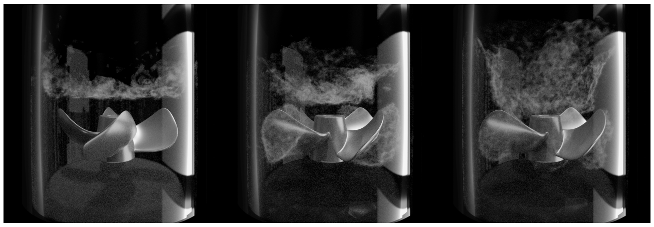


Fig. 10. Propeller spinning in water. Foams are formed, especially at higher propeller speeds. See Sec. 6.3.

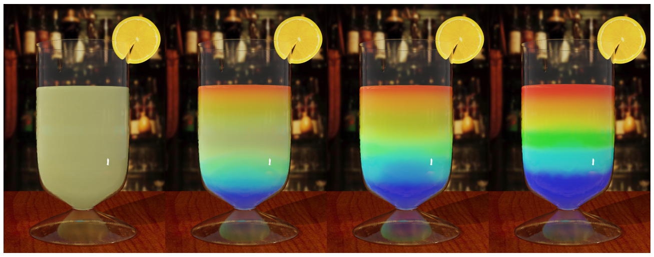


Fig. 11. A fully mixed cocktail with phase separation due to the lack of interphase momentum. See Sec. 6.3.