

High Density Ratio Multi-fluid Simulation with Peridynamics

HAN YAN, TMCC, College of Computer Science, Nankai University, China
BO REN*, TMCC, College of Computer Science, Nankai University, China

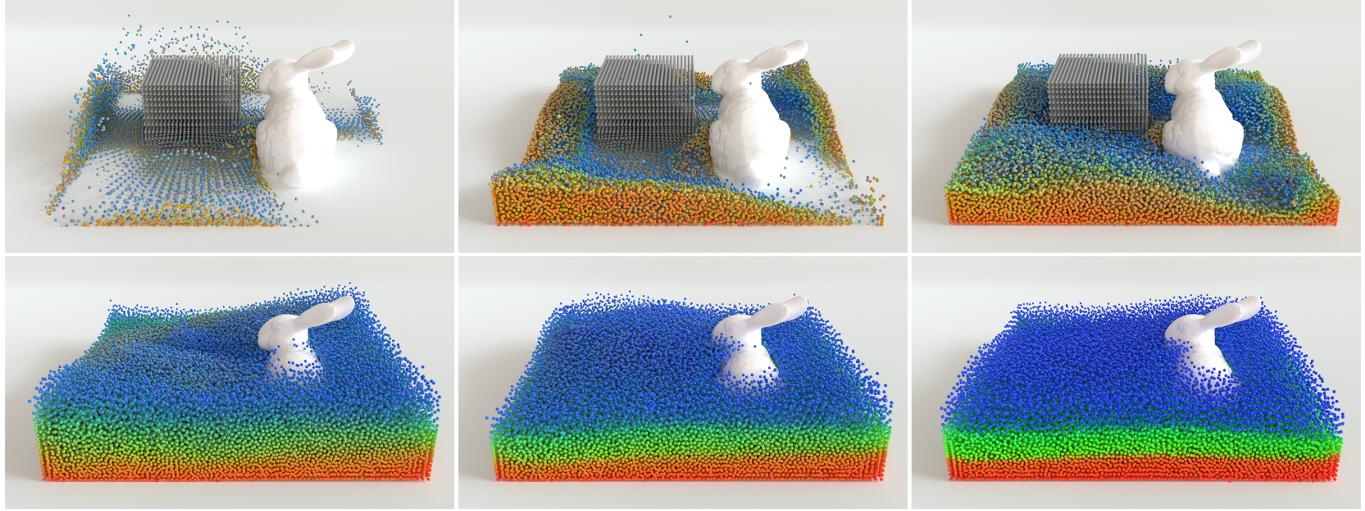


Fig. 1. Particle views of three immiscible phases unmixing in a rectangular container holding a static bunny obstacle, with three phases injected in a 1:1:1 mixed volume ratio. The density ratio is set to blue:green:red=1:500:1000. Our method keeps good phase mass conservation and volume ratio during fluid-solid interactions.

Multiple fluid simulation has raised wide research interest in recent years. Despite the impressive successes of current works, simulation of scenes containing mixing or unmixing of high-density-ratio phases using particle-based discretizations still remains a challenging task. In this paper, we propose a peridynamic mixture-model theory that stably handles high-density-ratio multi-fluid simulations. With assistance of novel scalar-valued volume flow states, a particle based discretization scheme is proposed to calculate all the terms in the multi-phase Navier-Stokes equations in an integral form. We also design a novel mass updating strategy for enhancing phase mass conservation and reducing particle volume variations under high density ratio settings in multi-fluid simulations. As a result, we achieve significantly stabler simulations in mixture-model multi-fluid simulations involving mixing and unmixing of high density ratio phases. Various experiments and comparisons demonstrate the effectiveness of our approach.

CCS Concepts: • Computing methodologies → Physical simulation.

*Corresponding author

Authors' addresses: Han Yan, TMCC, College of Computer Science, Nankai University, 38 Tongyan Rd, Tianjin, Tianjin, 300381, China, 1811205@mail.nankai.edu.cn; Bo Ren, TMCC, College of Computer Science, Nankai University, 38 Tongyan Rd, Tianjin, Tianjin, 300381, China, rb@nankai.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM. 0730-0301/2023/12-ART191 \$15.00 <https://doi.org/10.1145/3618347>

Additional Key Words and Phrases: multi-fluid, high-density-ratio, peridynamics, mass conservation

ACM Reference Format:

Han Yan and Bo Ren. 2023. High Density Ratio Multi-fluid Simulation with Peridynamics. *ACM Trans. Graph.* 42, 6, Article 191 (December 2023), 14 pages. <https://doi.org/10.1145/3618347>

1 INTRODUCTION

Simulations of multi-phase mixtures have raised interest in the computer graphics community in recent years. Commonly observed mixtures in the real world fall into a wide range of variety both physically and visually. Impenetrable liquid-liquid mixture (e.g. oil-water mixing), solid-liquid suspension (e.g. muddy soil flushed by the rain), and gas-liquid mixture (e.g. carbonated lemonade) are some of the examples among them. Such a wide range of various materials not only brings high complexities to physical models describing the mixture flow, but also often results in an intractable problem of very-high density ratios among phases in the simulation, especially for fully dispersed or miscible mixtures, and mixtures containing reactions constantly changing phase constitutions.

Particle-based [Band et al. 2018; Mercier et al. 2015], grid-based [Losasso et al. 2006] or hybrid [Stomakhin et al. 2013] methods have been popular for multi-phase simulations in graphics research. They have achieved impressive results for various multi-phase phenomena involving liquid and solid. However, simulations containing miscible mixtures with high-density ratios remains to be challenging tasks for algorithms adopting Lagrangian discretizations. Naively computing the gradients for either densities or phase velocities may

generate high errors and eventually lead to simulation failures under unpredictably dispersed distributions of high-density-ratio phases. Especially, this leads to difficulties in efficiently simulating unmixing effects, e.g. within highly-dispersed gas-liquid mixtures, using particle-based simulators. Another challenging factor is the huge difference in particle volume. The standard SPH framework assures mass conservation by keeping the mass of a particle unchanged during the simulation, but this causes a high volume ratio between different particles. For example, two particles having a high-density ratio of 1000:1 will have a volume ratio of 1000:1, which means their radius ratio is 10:1. Such a large radius ratio not only results in terrible visual effects but also brings about numerical errors. Moreover, current miscible multiple fluid solvers [Ren et al. 2021b] do not ensure phase mass conservation. This causes a more severe problem in high-density-ratio simulations.

Some recent works attempt to extend the Smoothed Particle Hydrodynamics(SPH) based multi-phase mixture model [Ren et al. 2014; Yan et al. 2016a] to two-phase flows with high density ratios, such as highly dispersed air-liquid mixture [Fonty et al. 2019, 2020]. However, these works only achieve stability under a strong assumption that only hydrostatic pressure is considered. The noisy inter-particle pressure field difference is completely ignored. Moreover, an interesting but key acknowledgment given by Fonty et al.[2020] is that:"as far as the pressure gradient is concerned, the multiple tests made with different formulations suggested by the literature, did not allow us to identify a completely satisfactory discrete expression", which strongly suggests that there is a fundamental difficulty to handle high-density-ratio multi-phase fluid simulations for the conventional SPH formulation using gradient calculations.

To overcome the above difficulty, we turn sight to solving the multi-phase Navier-Stokes equations with integral forms that can completely avoid calculations of gradients. Fortunately, a promising methodology candidate, namely the state-based peridynamics [Silling et al. 2007], proves helpful in such a task. The state-based peridynamic method has mainly received attentions in solid simulation works [Chen et al. 2018; He et al. 2017; Levine et al. 2014] with success in handling cracks or ruptures. However, there have also been some researches showing that it potentially generalizes to fluid simulations: Silling et al.[2007] demonstrate that a simple single-phase Navier Stokes equation can be solved with state-based peridynamics, and He et al.[2017] show their peridynamic model for elastoplastic materials can reproduce viscoelastic fluid motion with a carefully designed particle redistribution algorithm. Our work will in turn provide a compact solver for the multi-phase Navier-Stokes equations taking advantage of the integral-form mathematical nature of the state-based peridynamics.

In this work we propose a novel particle-based framework for multi-phase fluid simulation. We re-derive the previous smoothed particle hydrodynamics (SPH) mixture model [Ren et al. 2014] using the methodologies of peridynamics. To achieve this, we define volume flow scalar states that are key to deriving the governing equations into integral forms. As a result, the multi-phase Navier Stokes (N-S) equations are calculated in integral discretization forms in our approach, avoiding the highly erroneous direct computing of gradients under high-density ratios between phases. By allowing

particle mass to change, we further derive a novel mass updating strategy for enhancing phase mass conservation and reducing volume variation. As will be demonstrated in §6, our peridynamic mixture-model fluid (PMF) simulator can stably simulate miscible and immiscible mixture density ratios up to 1000 : 1 with moderate time step sizes, which is far better than current multi-phase simulators using conventional SPH formulations. As a result, our proposed general multi-phase simulation framework can recover challenging real-world phenomena featuring high phase density ratios, such as gas-liquid stratification within a pipe, etc. Our main contributions are:

- We derive a peridynamic mixture-model fluid (PMF) theory through introducing the volume flow scalar states for the multi-phase N-S equations.
- We propose a novel particle-based discretization scheme that calculates the multi-phase N-S equations in fully integral forms, avoiding direct calculation of gradients.
- We derive a novel mass updating strategy using variable particle masses for enhancing phase mass conservation and reducing particle volume variations in the simulation.
- We propose a multi-fluid simulation framework that is able to stably simulate high-density ratio mixing and unmixing effects, such as mixed gas-liquid stratification, with moderate computational cost.

2 RELATED WORK

2.1 Multi-phase Fluid Simulation

Multi-phase fluid simulations have been studied in computer graphics for a long time. Early works [Premžoe et al. 2003; Zhu et al. 2007, 2006] address simple mixing effects of immiscible/miscible flow. The concept of volume fraction is first introduced by Müller et al [2005] and later applied to grid-based solvers (e.g. [Kang et al. 2010]) for mixing effects.

In [Ren et al. 2014], an SPH-based mixture-model simulator for multi-phase fluid is proposed, which is able to model the phase-wise motions through analytically calculating a drift velocity between a phase and the aggregate mixture. This model is later extended in [Yan et al. 2016b] for including solid-liquid interactions concerning deformable bodies and granular materials. On the other hand, Yang et al.[2015] combine the multi-phase fluid simulation with the phase-field Cahn-Hilliard equations, simulating extraction and phase control using the Helmholtz free energy. Later, Tao et al.[2017] introduce a unified particle framework integrating the phase-field method with multi-material simulation, which can handle both liquids and solids, and the phase transitions between them. In [Ren et al. 2021b], a virtual-phase based approach is proposed to let the mixture model be able to handle multi-phase simulations including porous materials. The incompressibility of the mixture-model simulators is studied in several works. They solve a set of linear equations using deformation gradient [Ren et al. 2021a] or the divergence-free condition under an alternative mixture velocity definition [Jiang et al. 2020] that ensures the incompressibility in the simulation. Recently, Jiang and Lan[2021] extend [Ren et al. 2014] by directly calculating the phase velocity changes, achieving more dynamic mixing effects in the visual results.

The material point method (MPM) is also adopted for multi-phase simulations in recent years and achieved impressive results. An MPM solution for multi-phase solid-fluid interactions is provided in [Yan et al. 2018]. A two-layer grid structure is adopted in [Gao et al. 2018; Tampubolon et al. 2017] to capture solid-fluid interactions in granular multi-phase flows.

The above methods pays little attention to phase-mixing containing high-density ratios in general multi-phase flows. Despite on whether SPH, MPM or other discretizations their methods are based, all the method needs to directly compute the "Nabla" differential operator (i.e. for gradient, divergence, etc.) in the algorithms. Our proposed framework fully relies on integral-based calculations, and is able to stably simulate various phase-mixing effects with extremely high density ratios.

2.2 Density Ratio Handling

Multi-phase simulations containing relatively high-density ratios are typically studied in the graphics community in interfacial flows such as air-liquid flows. In this way, many methods are proposed to handle discontinuity problems in the interface regions. Hong and Kim[2005] use a ghost fluid method to accommodate jumps in physical values while keeping modifications to the linear system small . Solenthaler and Pajarola[2008] adjust the density calculations near the interfaces to successfully achieve high-density ratio immiscible multi-phase SPH simulations. Boyd and Bridson[2012] use the fluid implicit particle (FLIP) method to simulate the "glugging" effect of air and water. Their method reconstructs interfaces using Lagrangian particles and solves a combined divergence-free projection equation handling physical attribute jumps across interfaces on a grid following the ghost fluid method manner. For high-density ratio bubble-liquid simulations, Ihmsen et al.[2011] use the SPH method to simulate air and fluid phases with carefully designed drag-force-based two-way coupling at the fluid-air interface to avoid instabilities. Recently, Li et al.[2020] introduce a lattice Boltzmann method to handle the discretization of the gradient near the interface of two immiscible phases for air and liquid interactions. Contrary to the above methods, our approach can apply to miscible multi-phase simulations where no clear interface is present within the simulation.

2.3 Peridynamics

Peridynamics is first proposed in [Silling 2000] to simulate the fractures in elastic and plastic simulation. It is originally applied to model pairwise force groups in the solid theory. Silling and Askari[2005] use bond-based peridynamics to simulate the growth of the crack in elastic material. The above bond-based formulations are confined to certain brittle fracture models, and it is discussed that discretization of the bond-based peridynamics falls in the form of piecewise constant approximations of the true integrals in [Ganzenmüller et al. 2015], which leads to numerical errors. However, in [Silling et al. 2007], a much more general state-based peridynamic methodology is proposed, introducing a concept of "state" as a general description of local interactions. In the discretization of such state-based peridynamics, the inter-nodal interactions are no longer independent to each other, which leads to better simulation in a

much broader range of materials [Silling et al. 2007]. The graphics community also adopts the peridynamics for solid simulations in recent years. Levine et al.[2014] integrate peridynamics with a spring-mass system to simulate the fracture. He et al.[2017] derive a new method for simulating elastoplastic materials such as cloth. In addition to the simulation of solids, they also simulated viscoelastic fluids by combining PBF and peridynamics. However, their model still relies on the SPH kernel functions which need to directly calculate the derivatives and requires a carefully designed particle redistribution scheme to achieve feasible visual effects. Chen et al.[2018] also use state-based peridynamics to model elastoplasticity and anisotropy solid. The effectiveness of state-based peridynamics and its integral form calculation have been much studied in the computational physics literature in various simulation tasks. The priority of an integral-form calculation over the original SPH gradient has been studied by [Huang et al. 2015]. Van Le and Boabar[2018] and Wang et al.[2019] show that by substituting direct gradient calculation with integral forms when implemented with Lagrangian methods such as SPH, the state-based peridynamics performs better than similar integral approaches and agrees well with theoretical and experimental results. We extend the state-based peridynamic methodology to multi-phase fluid simulation, and derive a compact framework that fully avoids differential calculations in the particle-based discretization of the multi-phase N-S equations.

3 THEORETICAL BACKGROUNDS

In this section, we briefly recap the theories of the multi-phase mixture model and the state-based peridynamic foundations. Our derivations will follow in §4.1 and §4.2. Definitions of symbols are listed in Table 1.

3.1 Mixture Model

In computer graphics, the standard mixture model [Ren et al. 2014] discretizes multi-phase fluids with one set of SPH particles, in which each particle contains volume fractions and phase velocities to describe its different fluid phases. The relative velocity of each phase to the mixture is called the drift velocity. In mixture model, a single fluid particle shares properties including mass, rest density, pressure and mixture velocity, which is the centroid velocity of the particle with multiple components. Let $\rho_m = \sum_k \alpha_k \rho_k$ be the mixture density

and $v_m = \frac{\sum_k \alpha_k \rho_k v_k}{\rho_m}$ be the mixture velocity. The drift velocity is defined as

$$v_{mk} = v_k - v_m. \quad (1)$$

The volume fraction α represents the proportion of each component carried by a particle, and drift velocity is an important physical quantity that describes the volume fraction transfer between mixture particles. These mixture particles are updated in each timestep with their mixture velocity and the rate of change of volume fraction. The multi-phase governing equations are:

$$\frac{\partial \alpha_k}{\partial t} + (v_m \cdot \nabla) \alpha_k = -\alpha_k \nabla \cdot v_m - \nabla \cdot (\alpha_k v_{mk}), \quad (2)$$

$$\frac{\partial}{\partial t} v_m + (v_m \cdot \nabla) v_m = -\frac{\nabla p_m}{\rho_m} + g + \frac{\nabla \cdot T_m}{\rho_m} + \frac{\nabla \cdot T_{Dm}}{\rho_m}, \quad (3)$$

$$\begin{aligned} \mathbf{v}_{mk} = & \tau \left(\rho_k - \sum_{k'} c_{k'} \rho_{k'} \right) \mathbf{a} - \tau \left(\nabla p_k - \sum_{k'} c_{k'} \nabla p_{k'} \right) \\ & - \sigma \left(\frac{\nabla \alpha_k}{\alpha_k} - \sum_{k'} c_{k'} \frac{\nabla \alpha_{k'}}{\alpha_{k'}} \right). \end{aligned} \quad (4)$$

where $\mathbf{T}_{\mathbf{m}}$, $\mathbf{T}_{\mathbf{Dm}}$ and c_k represents the mixture's viscous stress tensor, convective momentum transfer between phases and mass fraction of phase k , respectively. The acceleration \mathbf{a} is

$$\mathbf{a} = \mathbf{g} - \frac{\partial}{\partial t} \mathbf{v}_m - (\mathbf{v}_m \cdot \nabla) \mathbf{v}_m, \quad (5)$$

which is the difference between the gravity acceleration \mathbf{g} and the substantial derivative of the mixture velocity [Ren et al. 2014]. What's more, τ and σ are constant coefficients that affect pressure effect and diffusion effect respectively. Generally speaking, larger τ makes pressure-related effects such as layering quicker, larger σ brings more diffusive appearance.

There are multiple gradient or divergence calculations in the above equations, which are directly computed in previous approaches. With presence of high density ratios between phases, the concentration and phase velocity field on the particles have high variances, leading to large errors and stability issues for such schemes.

3.2 State-Based Peridynamics

Proposed by Silling et al. [2007], the state-based peridynamic theory largely extends the formerly bond-based peridynamics to more reliably handle the simulations of a wide range of materials. A mathematical “state” object is introduced in their work, whose mathematical properties and operations lead to integral form derivations of the governing equations of various physical systems. Combining the state-based peridynamics and the Lagrangian discretization, the inter-particle force term between each particle pair will collectively depend on the all-particle distribution in the neighborhood, which is different from the case of traditional SPH formulations. These properties of the state-based peridynamics are promising for our purpose of stable high-density-ratio multi-phase simulations, for they theoretically both avoid the gradient calculations using integral equations and can alleviate the large inter-particle force variance due to unpredictable neighborhood distribution with large differences in physical quantities under high density ratios.

A state of order m , which is the core mathematical concept of the state-based peridynamic methodology, is defined by Silling et al. [2007] as a function that maps vectors within a spherical neighborhood into m -order tensors, where m is an integer. When $m = 1$, the state is called a vector state. For example, suppose that there is a 3×3 rotation matrix \mathbf{R} that can rotate a vector \mathbf{a} in \mathbb{R}^3 , and the rotation operation generates a vector $\mathbf{b} = \mathbf{R}\mathbf{a}$. Or we can say there is a state of order 1(or a vector state) $\underline{\mathbf{R}}$ that maps \mathbf{a} into \mathbf{b} (a tensor of order 1):

$$\underline{\mathbf{R}}(\mathbf{a}) = \mathbf{b}. \quad (6)$$

Similarly, a state of order 0 will map a vector into a scalar value, and is called a scalar state.

Silling et al. [2007] further introduces a set of mathematical operations, definitions and properties based on the state concept, which

Table 1. Definition of symbols

Symbol	Meaning
α_k	volume fraction of phase k
c_k	mass fraction of phase k
ρ_k	rest density of phase k
ρ_m	rest density of a mixture particle
$\bar{\rho}_m$	interpolated density of a mixture particle
$\mathbf{v}_k, \mathbf{v}_m$	velocity of phase k and the mixture
p_k, p_m	pressure acting on phase k and the mixture
\mathbf{v}_{mk}	drift velocity
\mathbf{g}	gravity
\mathbf{x}, \mathbf{x}'	position of particles in the reference state
\mathbf{y}	position of a particle in the current state
\mathbf{u}	displacement vector field
ξ	relative position of x and x'
\mathcal{H}	a spherical neighborhood of radius h centered at x
\mathcal{A}_m	the set of all tensors of order m
\mathcal{V}	the set of all vectors(tensors of order 1)
$\underline{\mathbf{T}}$	force vector state
$\underline{\mathbf{X}}$	reference position vector state
$\underline{\mathbf{Y}}$	deformation vector state
\underline{s}	volume flow scalar state
s_k	volume flow scalar state of phase k
$\underline{\mathbf{K}}$	shape tensor
\mathbf{F}	deformation gradient tensor
$W(\xi, h)$	kernel function

are too involved to be fully recapped in this section. Fortunately, our later derivation mainly relies on 4 basic definitions of the state-based peridynamic methodology. We quote them below from [Silling et al. 2007] within the quotation marks:

“Let h be a positive number, and let \mathcal{H} be a spherical neighborhood of radius h centered at the origin in \mathbb{R}^3 . Let \mathcal{A}_m denote the set of all tensors of order m (thus $\mathcal{A}_0 = \mathbb{R}$ and \mathcal{A}_1 is the vector space).

Definition 1 A state of order m is a function $\underline{\mathbf{A}}(\cdot) : \mathcal{H} \rightarrow \mathcal{A}_m$.

If components are used in a Cartesian coordinate system, an order m state has m components that are written as $\underline{A}_{i_1 i_2 \dots i_m}$. The vectors in \mathcal{H} on which a state operates are written in angle brackets $\langle \cdot \rangle$ to distinguish them from other quantities that the state itself may depend on.

Definition 2 The set of all vector states is denoted \mathcal{V} . Given a kernel function $\underline{\omega} = W(\xi, h)$ where ξ is the relative position of a particle and its neighbor, for any two vector states $\underline{\mathbf{A}} \in \mathcal{V}$ and $\underline{\mathbf{B}} \in \mathcal{V}$, $\underline{\mathbf{A}} * \underline{\mathbf{B}}$ is the second-order tensor defined by

$$\underline{\mathbf{A}} * \underline{\mathbf{B}} = \int_{\mathcal{H}} \underline{\omega}(\xi) \underline{\mathbf{A}}(\xi) \otimes \underline{\mathbf{B}}(\xi) dV_{\xi}, \quad (7)$$

where \otimes means dyadic product(In Cartesian coordinates, $\mathbf{C} = \mathbf{a} \otimes \mathbf{b}$ equals $C_{ij} = a_i b_j$.)

Definition 3 The reference position vector state $\underline{X} \in \mathcal{V}$ is defined by

$$\underline{X}(\xi) = \xi. \quad (8)$$

Definition 4 The shape tensor K is defined by

$$K = \underline{X} * \underline{X}, \quad (9)$$

where $\underline{X} \in \mathcal{V}$ is the reference position vector state.”

The state concept plays a key role in deriving the fluid governing equations into integral form and eliminating the pressure gradient calculations. A demonstration has been given in [Silling et al. 2007] for a simple single-phase Navier-Stokes momentum equation, whose integral form is derived as:

$$\rho \ddot{\mathbf{u}} = \int_{\mathcal{H}} \{ \underline{T}(\xi) - \underline{T}'(-\xi) \} dV'_x + \rho g, \quad (10)$$

where ρ is the fluid density, \mathbf{g} is the gravity, \mathbf{u} is the displacement vector field, ξ is the relative position of a particle and its neighbor, and \underline{T} and \underline{T}' is the force vector state of the particle and its neighbor, respectively. The force vector state is then given by Silling et al. [2007] as:

$$\underline{T}(\xi) = \underline{\omega}(\xi) \sigma K^{-1} \xi, \quad (11)$$

Where K is the shape tensor (a 3×3 symmetric matrix) defined in Definition 4, Equation (9), $\underline{\omega}(\xi) = W(\xi, h)$ is a scalar state that is similar to the kernel function used in SPH, and σ is the Piola-Kirchoff stress tensor. We refer the readers to [Silling et al. 2007] for a thorough and comprehensive understanding of the state-based peridynamics. In the next section, we will demonstrate how we can use this methodology to reshape both the multi-phase continuity equation and the multi-phase momentum equation into integral forms analogous to that of Equation (10).

4 PERIDYNAMIC MIXTURE-MODEL FLUID

For the multiphase governing equations described in §3.1, our goal is to rewrite Equations (2-4) into integral forms, which will rely on using the “state” object in the peridynamic methodology. In this way, our PMF governing equations will be able to avoid all the derivatives such as pressure gradient or velocity divergence, etc.

In this section, we describe how we can use state-based peridynamics to derive a fully integral-based set of multi-phase fluid governing equations. The completion of the terms in the derived governing equations are left to §5.1 within a particle-based discretization scheme.

4.1 Continuity Equation

We first study the continuity equation which is used to solve for the phase concentration (i.e. volume fraction) changes in multi-phase fluid motions. Given a fluid element with volume V , within which each phase has volume fraction α_k and phase density ρ_k , the change rate of mass of the k th phase is

$$\frac{d}{dt} (\alpha_k \rho_k V) = V \frac{d}{dt} (\alpha_k \rho_k) + \alpha_k \rho_k \frac{dV}{dt}, \quad (12)$$

since ρ_k stays constant in the simulation, by re-arranging the above terms, we get the change rate of the volume fraction:

$$\frac{d}{dt} \alpha_k = \alpha_k \left(-\frac{1}{V} \frac{d}{dt} V + \frac{1}{\alpha_k V} \frac{d}{dt} (\alpha_k V) \right). \quad (13)$$

Here and in the following derivations, we call the spatial volume V of the discussed fluid element a “meta volume”, which is a small volume at a macroscopic scale that is so small that its internal physical quantities can be considered near-constant. Such a meta volume conveniently corresponds to the spherical neighborhood \mathcal{H} in the peridynamic methodology. Given Equation (13), one can observe that it is essential to formulate $\frac{1}{V} \frac{d}{dt} V$ and $\frac{1}{\alpha_k V} \frac{d}{dt} (\alpha_k V)$ into integral forms. The forms of these two terms are similar. Since $\alpha_k V$ is exactly the local volume of phase k within a meta volume, the cyan term corresponds to the mixture volume change and the magenta term corresponds to the phase volume change. We thus assume that within a meta volume, they are describing the relative volume flow, i.e. the percentage of mixture/phase volume flowing through the meta-volume boundary calculated from the boundary velocities within unit time, as is demonstrated in Fig. 2, where the volume flow $\frac{d}{dt} V$ is defined by how much volume flows through the boundary of meta volume within unit time, which links Equation (13) to the velocity fields. We then propose to use two volume flow states, i.e. scalar states $\underline{s}(\xi) = \mathbf{v}_m \cdot \nabla J/J$ and $\underline{s}_k(\xi) = \alpha_k \mathbf{v}_{mk} \cdot \nabla J/J$, to represent each of the terms in the peridynamic form of Equation (13). Here J is the Jacobian of the deformation gradient tensor \mathbf{F} of local fluid distribution the same as that in [Ren et al. 2021a]:

$$J = \frac{V}{V_0} = \det \mathbf{F}, \quad (14)$$

where V_0 is particle’s volume in the reference state. For compactness, we directly give the integral form of Equation (13), and leave the detailed derivation to Appendix A.

$$\frac{d\alpha_k}{dt} = \alpha_k \int_{\mathcal{H}} (\underline{s}(\xi) - \underline{s}'(-\xi)) dV'_x - \int_{\mathcal{H}} (\underline{s}_k(\xi) - \underline{s}'_k(-\xi)) dV'_x, \quad (15)$$

The volume flow scalar states \underline{s} and \underline{s}_k reflect the relative change rate of the volumes at neighboring positions, whose spatial integrals, proven in Appendix A, equal to the cyan and magenta terms on the right-hand side of Equation (13).

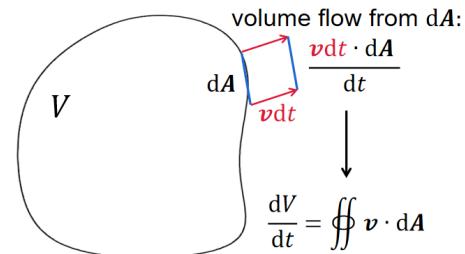


Fig. 2. Demonstration of the volume flow. The volume flow $\frac{d}{dt} V$ is defined by how much volume flows through the boundary of a meta volume within unit time, which links Equation (13) to the velocity fields.

Using the deformation gradient tensor \mathbf{F} , the volume flow scalar states can be further expressed in matrix forms without any nabla

operator (∇), we also directly give the final formulas and leave the derivations to Appendix A:

$$\underline{s} \langle \xi \rangle = \underline{\omega} \langle \xi \rangle v_m \cdot (\mathbf{F}^{-1} \mathbf{K}^{-1} \xi) \quad (16)$$

and

$$\underline{s}_k \langle \xi \rangle = \alpha_k \underline{\omega} \langle \xi \rangle v_{mk} \cdot (\mathbf{F}^{-1} \mathbf{K}^{-1} \xi). \quad (17)$$

where \mathbf{K} and \mathbf{F} are both reversible second-order tensors.

4.2 Momentum Equation

For simplicity, we give the PMF forms for the multi-phase momentum equation directly and include the derivation in Appendix B. Similar to Equation(10), the multi-phase momentum equation is

$$\frac{dv_m}{dt} = \frac{1}{\rho_m} \int_{\mathcal{H}} (\underline{T} \langle \xi \rangle - \underline{T}' \langle \xi \rangle) dV_\xi + \mathbf{g} + \sum_k \left(\frac{d\alpha_k}{dt} \frac{\rho_k v_{mk}}{\rho_m} \right), \quad (18)$$

where \underline{T} is the force vector state. As will be shown in the next section, we can analytically calculate the drift velocity terms similar to the integral-form force term, avoiding the gradient calculations in Equation (2,3,4). As a result, The last term can be computed in integral form using Equation (15), and Equation (18) is the integral-form momentum equation for a mixture particle.

5 IMPLEMENTATION

In this section, we first introduce the discretization for the governing equations (15,18) and the drift velocity using a particle-based framework, thus completing our theory in §5.1. Then we describe an enhanced mass updating strategy to handle calculation issues raised in high-density-ratio simulation situations in §5.2. We also derive the chemical reaction formulas in §5.3 and reformulate the boundary handling method [Akinci et al. 2012] in our peridynamic form in §5.4. Finally, we demonstrate the algorithm framework of our approach in §5.5.

5.1 Discretization

Continuity equation terms. We start from the shape tensor \mathbf{K} . Following [Silling et al. 2007],

$$\mathbf{K}_i = \sum_j \frac{m_j}{\bar{\rho}_{m,j}} W(\xi, h) \xi \otimes \xi, \quad (19)$$

where $W(\xi, h)$ is the cubic spline kernel from [Koschier et al. 2019], $\xi = \mathbf{x}_j - \mathbf{x}_i$, \otimes means the same as dyadic product in Equation (7), and i and j in the summation notation represents the index of a mixture particle and its neighbor particle, respectively.

For Equation (15), we first obtain the peridynamic definition of the deformation gradient tensor from [Silling et al. 2007]:

$$\mathbf{F} = (\underline{Y} * \underline{X}) \mathbf{K}^{-1}, \quad (20)$$

which can be calculated as

$$\mathbf{F}_i = \sum_j \frac{m_j}{\bar{\rho}_{m,j}} W(\xi, h) \{ [\xi + (v_{m,j} - v_{m,i}) dt] \otimes \xi \} \mathbf{K}_i^{-1}. \quad (21)$$

Although \mathbf{K} and \mathbf{F} are theoretically reversible (\mathbf{K} is proven by [Silling et al. 2007] and \mathbf{F} can be seen by the fact that its determinant represents the relative volume ratio before and after motion, thus never

equals to zero from a physical perspective), but in the implementation we need to handle irreversible cases in rare numerical cases. For irreversible \mathbf{K} , we recalculate it with:

$$\mathbf{K}_i = \frac{m_i}{\bar{\rho}_{m,i}} W(\mathbf{0}, h) \mathbf{I} r^2, \quad (22)$$

where \mathbf{I} is identity matrix, r is the particle radius. If \mathbf{F} becomes irreversible, it means that the particle may have extremely small volume due to compressing. In such a case we assume the small volume is uniformly deformed and

$$\mathbf{F}_i = \frac{1}{1 + N_i} \mathbf{I}, \quad (23)$$

where N_i is the neighborhood particle number of particle i .

The discretization of Equation (15) is

$$\frac{d\alpha_k}{dt} = S + S_k, \quad (24)$$

where S, S_k are in symmetric forms using the kernel function:

$$S_i = \sum_j \frac{m_j}{\rho_{m,j}} (\alpha_{k,j} - \alpha_{k,i}) (v_{m,i} - v_{m,j}) \cdot \mathbf{F}_i^{-1} \mathbf{K}_i^{-1} \xi W(\xi, h), \quad (25)$$

$$S_{k,i} = - \sum_j \frac{m_j}{\rho_{m,j}} (\alpha_{k,j} v_{mk,j} + \alpha_{k,i} v_{mk,i}) \cdot \mathbf{F}_i^{-1} \mathbf{K}_i^{-1} \xi W(\xi, h). \quad (26)$$

Momentum equation terms. Exploiting the similarity between the mixture-model and single-phase momentum equations, we directly replace the single-phase particle pressure and velocity in the discretization of the single-phase momentum equation in [Silling et al. 2007] with the mixture values to obtain the multi-phase discretization of the first term in Equation (18):

$$\frac{dv_m}{dt} = \frac{1}{\rho_m} (f_p + f_\mu) + \mathbf{g} + \sum_k \left(\frac{d\alpha_k}{dt} \frac{\rho_k v_{mk}}{\rho_m} \right), \quad (27)$$

where

$$f_{p,i} = - \sum_j \frac{m_j}{\bar{\rho}_{m,j}} (p_{m,j} + p_{m,i}) \mathbf{K}_i^{-1} \xi W(\xi, h) \quad (28)$$

and

$$f_{\mu,i} = \sum_j \frac{m_j}{\bar{\rho}_{m,j}} (\mu_j + \mu_i) \frac{(v_{m,j} - v_{m,i})(\xi \cdot \mathbf{K}_i^{-1} \xi) W(\xi, h)}{\|\xi\|^2} \quad (29)$$

are pressure force and viscous force acting on the mixture particle. The pressure p_m is calculated using the Tait equation of state in [Becker and Teschner 2007]. It is worth noting that, there are two major differences between Equations (28, 29) and the traditional SPH-formulation-based force calculation. First, the former no longer contain derivatives. Second, considering the dependency on ξ of the inter-particle force terms (i.e. each sum term within the summation), the existence of the shape tensor \mathbf{K} makes those of the former dependent to the all-particle position distributions in the neighborhood, while those of the latter falls into a piece wise constant manner, only depending on the related two particle's relative position. These two properties are directly inherited from the state-based peridynamic methodology. In our experiments in §6, the calculation using Equations (28, 29) leads to significantly better stability in the high-density-ratio miscible multi-fluid simulations.

Drift velocity terms. One can observe, by comparing Equation (28) with the standard SPH formulation of pressure gradient, that the calculation in Equation (28) effectively serves as an integral formulation for an original gradient term of a physical quantity (here it is the pressure gradient in the N-S equations). As a result, we are able to upgrade the drift velocity calculation in Equation (4) as follows:

$$\begin{aligned} v_{mk} = & \tau \left(\rho_k - \sum_{k'} c_{k'} \rho_{k'} \right) \mathbf{a} - \tau \left(T_{pk} - \sum_{k'} c_{k'} T'_{pk} \right) \\ & - \sigma \left(T_{\alpha_k} - \sum_{k'} c_{k'} T_{\alpha_{k'}} \right), \end{aligned} \quad (30)$$

where

$$T_{pk,i} = \sum_j \frac{m_j}{\bar{\rho}_{m,j}} (p_{k,j} - p_{k,i}) \mathbf{K}_i^{-1} \xi W(\xi, h), \quad (31)$$

$$T_{\alpha_k,i} = \sum_j \frac{m_j}{\bar{\rho}_{m,j}} \left(\frac{\alpha_{k,j}}{\alpha_{k,i}} - 1 \right) \mathbf{K}_i^{-1} \xi W(\xi, h), \quad (32)$$

and

$$\mathbf{a} = -\frac{1}{\rho_m} (f_p + f_\mu) - \sum_k \left(\frac{d\alpha_k}{dt} \frac{\rho_k v_{mk}}{\rho_m} \right). \quad (33)$$

As in [Ren et al. 2014], our approach deals with immiscible fluids by setting $p_k = p_m$ and with miscible fluids by setting $p_k = \alpha_k p_m$.

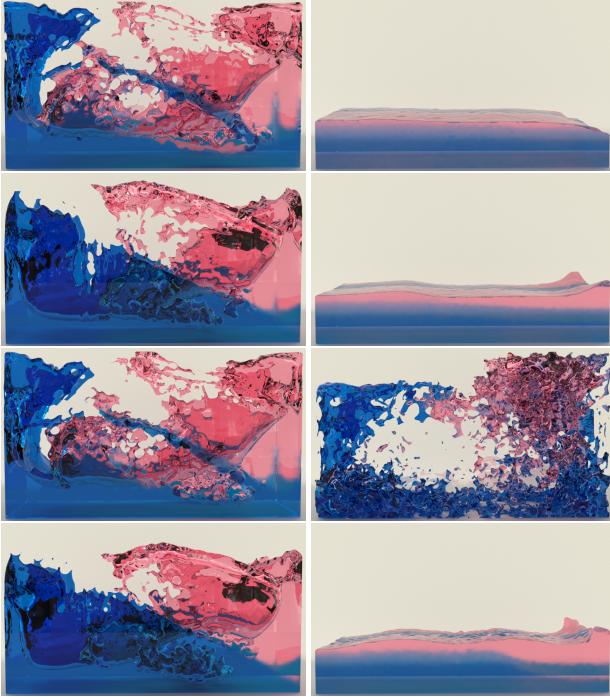


Fig. 3. Stability comparison with two miscible phases, part 1. From the top row to the bottom row: (1)Density ratio of blue to pink is 20:1, method in [Ren et al. 2014]. (2)Density ratio of blue to pink is 20:1, our method. (3)Density ratio of blue to pink is 50:1, method in [Ren et al. 2014]. (4)Density ratio of blue to pink is 50:1, our method.

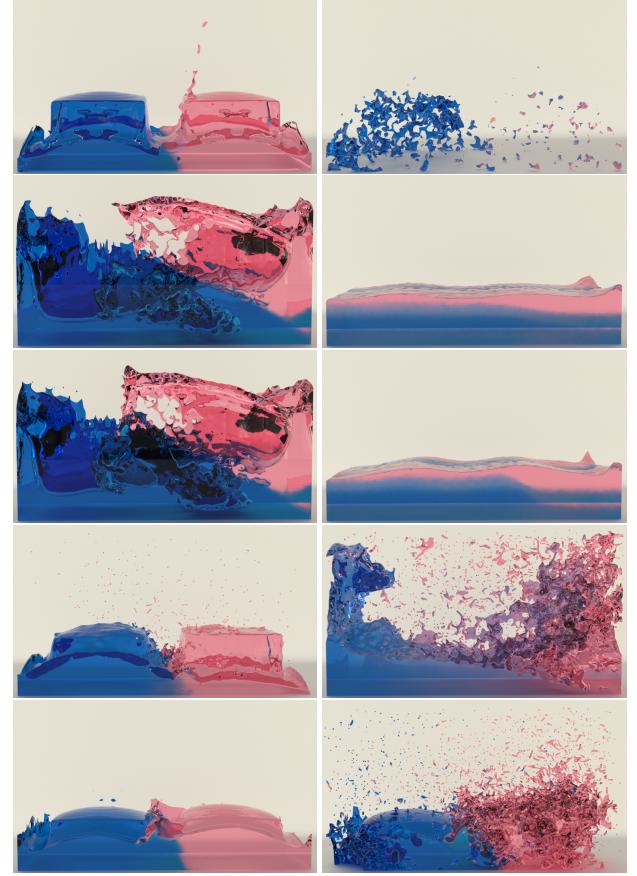


Fig. 4. Stability comparison with two miscible phases, part 2. From the top row to the bottom row: (1)Density ratio of blue to pink is 50:1, method in [Ren et al. 2014] using number density that in [Solenthaler and Pajarola 2008] (2)Density ratio of blue to pink is 50:1, our method using number density. (3)Density ratio of blue to pink is 200:1, our method. (4)Density ratio of blue to pink is 10:1, method in [Yang et al. 2015]. (5)Density ratio of blue to pink is 10:1, method in [Ren et al. 2021a]. Our method is stable while the previous method [Ren et al. 2014] using or without using number density all fail at a density ratio of 50:1. Method in [Yang et al. 2015] and method in [Ren et al. 2021a] all fail at a density ratio of 10:1.

5.2 Mass Updating for Enhanced Mass Conservation and Particle Volume Variation

Enhanced mass conservation. Equations (25,26) do not automatically ensure phase mass conservation during the simulation. One possible reason is that the $\mathbf{F}^{-1}\mathbf{K}^{-1}$ part in the calculations is still not symmetric. In our approach, Equations (25,26) are further adjusted to assure pair-wise phase-mass transfer equality between particles, which should obey:

$$\Delta m_k^{i \rightarrow j} + \Delta m_k^{j \rightarrow i} = 0, \quad (34)$$

or

$$\Delta \alpha_k^{i \rightarrow j} V_i + \Delta \alpha_k^{j \rightarrow i} V_j = 0, \quad (35)$$

where $V = m/\rho_m$ is the volume of a mixture particle, and i and j in the superscript are the index of a mixture particle, thus $\Delta m_k^{i \rightarrow j}$

is the k -th phase mass transmitted from particle i to particle j . The enhanced Equation (25) and Equation (26) are then rewritten as

$$S_i = \sum_j \frac{m_j}{\bar{\rho}_{m,j}} \frac{V_j}{V_i + V_j} (\alpha_{k,j} - \alpha_{k,i}) (v_{m,i} - v_{m,j}) \\ \cdot [F_i^{-1} K_i^{-1} + (F_j)^{-1} (K_j)^{-1}] \xi W(\xi, h), \quad (36)$$

$$S_{k,i} = \sum_j \frac{m_j}{\bar{\rho}_{m,j}} \frac{V_j}{V_i + V_j} (\alpha_{k,j} v_{mk,j} + \alpha_{k,i} v_{mk,i}) \\ \cdot [F_i^{-1} K_i^{-1} + (F_j)^{-1} (K_j)^{-1}] \xi W(\xi, h). \quad (37)$$

Enhanced particle volume variation. For particle-based simulations of miscible fluids with high-density ratios, one challenging problem is that the effective particle volumes may change violently given a constant particle mass. This problem will bring numerical errors during calculation, and cause severely bumpy surfaces or interfaces in the final rendering since effective particle resolution will be low for the lighter phases. In our experiments, we find that if we allow the particle mass to be changed, the max volume ratio in the simulation can be conspicuously reduced (e.g. from 1000:1 to 10:1), and the above issues can be alleviated. We then develop a discretization scheme with variable particle mass as follows.

Since we assume incompressibility of each phase, the rest density of phase k keeps the same in the whole simulation as ρ_k . On the other hand, at each time step, we can calculate the phase mass transfer of phase k as $\Delta m_k^i = -\sum_j \Delta m_k^{i \rightarrow j}$ for a particle i from Equations (35 - 37). This means we can calculate the new masses both for each phase and the total particle mass, which directly leads to new mass fractions c_k . We then update the particle rest mass and rest aggregate density. In case the phase mass is less than zero after the update, we adopt the iterative updating strategy in [Jiang and Lan 2021]. Afterward, the volume fractions of each phase within a particle are re-calculated using the relation

$$\alpha_k = \frac{c_k / \rho_k}{\sum_k c_k / \rho_k}. \quad (38)$$

Here \sum_k means the summation over all phases. The above mass updating strategies allow our algorithm to achieve significantly better phase mass conservation and reduce volume variance in the simulation. We demonstrate the effectiveness in §6.

5.3 Chemical reaction

We adopt a similar chemical reaction handling strategy from [Ren et al. 2014]. When liquid components A and B react to produce C, suppose the chemical reaction equation is written as



Where x, y and z are the stoichiometric coefficients. According to the conservation of mass, within one particle, the mass change in component C is

$$m\Delta c_C = \Delta m_C = -\Delta m_A - \Delta m_B = -m\Delta c_A - m\Delta c_B, \quad (40)$$

where m is the mass of the mixture particle and c_A, c_B, c_C are the mass fractions. In chemical reactions, the rate of reaction is proportional to the concentration of reactants and inversely proportional

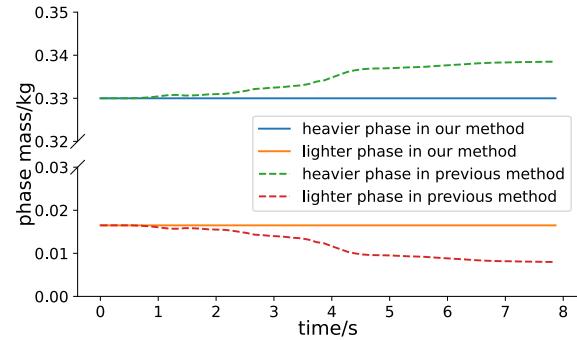


Fig. 5. Change of phase mass in the first and second rows of Fig. 3. The solid and dashed lines represent the change of phase mass in our method and the previous method [Ren et al. 2014] respectively. Our approach achieves better phase mass conservation in the simulation.

to the concentration of products. We express the change of mass fraction in the reaction using

$$\Delta c_A = \gamma \frac{x \alpha_A \alpha_B \rho_A \rho_B}{\alpha_c \rho_c + \lambda} \quad (41)$$

and

$$\Delta c_B = \gamma \frac{y \alpha_A \alpha_B \rho_A \rho_B}{\alpha_c \rho_c + \lambda}, \quad (42)$$

where γ and λ are constants that control reaction rate. After applying the change of mass fraction, the volume fraction of each component can be calculated with Equation (38). In our experiment, we set $\gamma = 1 \times 10^{-5}$ and $\lambda = 100 \text{ kg} \cdot \text{m}^{-3}$.

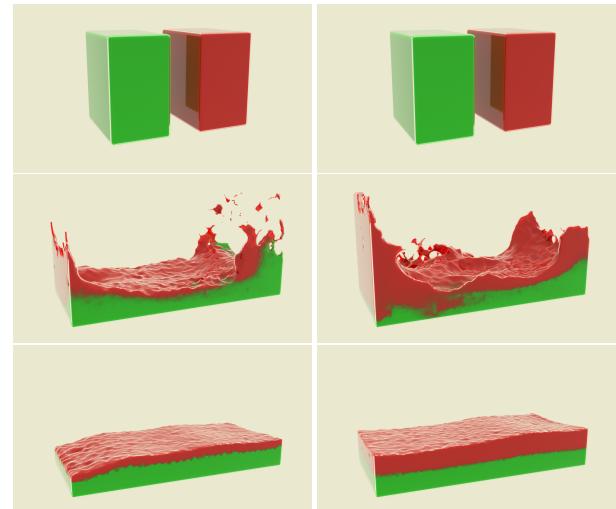


Fig. 6. Demonstration of phase mass conservation of our approach. The left column is peridynamic mixture-model fluid(PMF) without the enhanced mass updating strategy which is discussed in §5.2, and the right column is PMF that uses the enhanced mass updating strategy. The red phase and green phase is set to a density ratio of 1:50 and a volume ratio of 1:1 at the beginning. Our proposed scheme successfully conserves phase mass during the simulation.

5.4 Boundary Handling

The boundary handling in our approach is a rigid fluid coupling technique derived in [Akinci et al. 2012]. For each rigid particle, its effective mass is

$$\Psi_{b_i}(\rho_0) = \rho_0 \frac{m_{b_i}}{m_{b_i} \sum_j W(\xi, h)}, \quad (43)$$

where ρ_0 is the rest density of a particle interacting with the rigid particle. The pressure force and viscous force the rigid particle b_i applies to the fluid particle i are

$$f_{p,i}^b = - \sum_{b_i} \frac{\Psi_{b_i}(\rho_0)}{\rho_{m,i}} p_{m,i} \mathbf{K}_i^{-1} \xi W(\xi, h), \quad (44)$$

$$f_{\mu,i}^b = - \sum_{b_i} \frac{\Psi_{b_i}(\rho_0)}{\rho_{m,i}} \mu^b \frac{(\mathbf{v}_{m,i} - \mathbf{v}_{m,b_i})(\xi \cdot \mathbf{K}_i^{-1} \xi) W(\xi, h)}{\|\xi\|^2}, \quad (45)$$

where μ^b is the viscosity coefficient between rigid and fluid, and \mathbf{v}_{m,b_i} is the velocity of the rigid particle. We can simply apply the rigid force by changing the momentum function:

$$\frac{d\mathbf{v}_m}{dt} = \frac{1}{\rho_m} (f_p + f_\mu + f_p^b + f_\mu^b) + \mathbf{g} + \sum_k \left(\frac{d\alpha_k}{dt} \frac{\rho_k \mathbf{v}_{mk}}{\rho_m} \right). \quad (46)$$

5.5 Algorithm framework

The whole algorithm is in Algorithm 1, which is similar to the standard SPH mixture model. An explicit pressure solver is used to calculate the mixture pressure according to interpolated particle densities.

ALGORITHM 1: Implementation for the multi-fluid PD framework

```

repeat
    for each fluid particle  $i$  and each rigid particle  $b_i$  do
        compute the mixture density and pressure for fluid using standard
        SPH discretization, and use Equation (43) for bound mass
    end for
    for each fluid particle  $i$  do
        compute the drift velocity using Equation (30), then compute the
        shape tensor  $\mathbf{K}$  and the deformation gradient tensor  $\mathbf{F}$  using
        Equation (19) and Equation (21)
    end for
    for each fluid particle  $i$  do
        compute the change rate of  $\alpha$  using Equation (24), Equation (36) and
        Equation (37)
        update the mass and rest density of mixture particles
        re-balance  $\alpha$  with Equation (38)
    end for
    for each fluid particle  $i$  do
        compute the acceleration using Equation (46)
    end for
    for each fluid particle  $i$  do
        advance the particle
    end for
     $t \rightarrow t + \Delta t$ 
until end of simulation

```

6 RESULTS

We implemented the proposed approach on an Nvidia GeForce RTX 3080Ti GPU. We use both [Yu and Turk 2013] and marching cube technology in Houdini to reconstruct the surface. The time step is set to around 10^{-3} s; the performance of the simulations in this section is recorded in Table 2. We generally set initial particle mass to 1.6×10^{-5} kg, the typical rest density of the heaviest phases are around $1000\text{kg} \cdot \text{m}^{-3}$ and the other phase densities are set according to the density ratios reported in each example. The length scale of the scenes in our experiments is set to around 0.5m. We also show the typical values of parameters used in the experiments in Table 3.

Table 2. Performance

Example	#Phases	# Particles	Time(s/step)
Rigid Bunny	3	288k	0.289
Stability Comparison	2	198k	0.266
Mass Conservation	2	78k	0.016
Gas-liquid Flow	2	247k	0.411
Triple Dam Break	3	263k	0.153
Phase Separation	2	281k	0.154
"RT" Instability	2	154k	0.370
Timing Comparison	2	198k	0.266
Rainbow Cocktail	3-4	287-301k	0.227
"Depth Charge"	4	124k	0.094
Rising Bubbles	2	284k	0.532

Rigid Bunny. Three immiscible phases with density ratio blue:green:red = 1:500:1000 are injected into a rectangular container holding a static bunny obstacle in Fig.1. We provide particle views of this example. Our method is able to simulate multi-fluid behavior interacting with solid objects, and the result also shows good conservation of phase mass and volume ratio. We note that thanks to our enhanced particle volume variation reducing scheme, although the final spacings of the blue particles inevitably becomes larger than those of the higher-density particles, they are much smaller than the could-be 10-times original spacing if particle mass is kept constant.

Stability Comparison. In Fig.3 and Fig.4, we compare the stability of our method with [Ren et al. 2014] using a miscible two-phase fluid. Both methods are stable around the density ratio 20:1(1st and 2nd rows in Fig.3). However, the method in [Ren et al. 2014] becomes unstable without or with the number density scheme in [Solenthaler and Pajarola 2008] at the density ratio 50:1 (3rd row in Fig.3 and 1st row in Fig.4). On the contrary, our method stays stable at the density ratio 50:1 without or with the number density scheme(4th row in Fig.3 and 2nd row in Fig.4). At the density ratio 200:1, our method still keeps stable in the simulation(3rd row in Fig.4). We also test the position-based-fluid method in [Yang et al. 2015](4th row) and the implicit multi-fluid solver in [Ren et al. 2021a](5th row), and these two methods all fail at the density ratio 10:1. Note that at density ratio of 20:1, the method in [Ren et al. 2014] also shows some phase mass loss of the lighter phase. We demonstrate the quantitative phase mass changes using the 20:1 case in Fig.5. The solid and dashed lines represent our method and the previous method

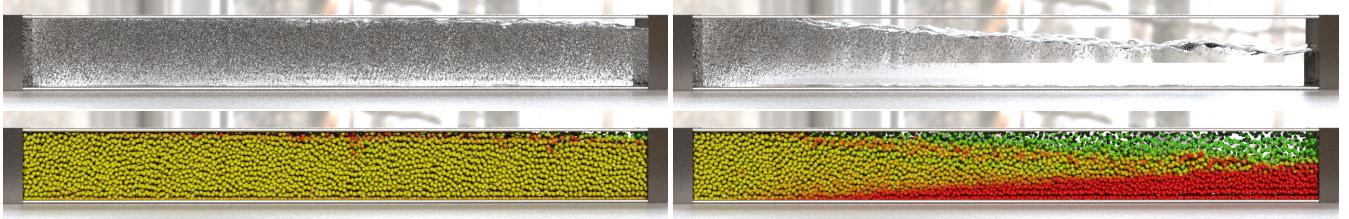


Fig. 7. A gas-liquid mixture with a density ratio of 1000:1 flows through a pipe from left to right. The injected mixture includes 50% gas and 50% liquid, and it is stratified during flow. We give the render views(1st row) and particle views(2nd row) in this example. As far as we know, our approach is the first particle-based approach that can achieve stable multi-fluid unmixing in such a high-density ratio benchmark.

[Ren et al. 2014], respectively. Our approach achieves significantly better phase mass conservation.

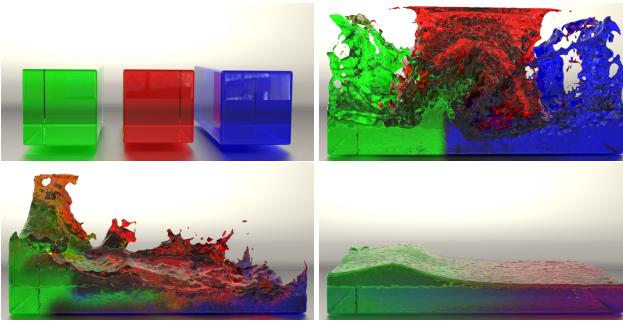


Fig. 8. Multiple fluid dam break with three miscible phases. The density ratio is red:green:blue = 5:600:1000. Our approach successfully achieves vigorous miscible mixing of high-density ratio phases.

Mass Conservation. In Fig.6, we demonstrate the advantage in phase-mass conservation of our novel mass updating strategy. The density ratio is set to be red:green=1:50. The two columns show the PMF method without/with the mass updating strategy discussed in §5.2. With our proposed scheme, the phase mass avoids severe loss of the lighter phase during the simulation and the phase volume ratio is also conserved.

Gas-liquid Flow. In Fig.7, a gas-liquid mixture with a density ratio of 1000:1 flows through a pipe from left to right. The pipeline is initially filled with a highly-dispersed gas-liquid mixture which is set to be mixed with a 1:1 volume fraction and flowing from left to right. After the simulation starts, the same initial mixture keeps flow into the pipe from the left, and the gas phase shall rise and the liquid phase shall go down, unmixing with each other before flowing out at the right. One shall expect a steady transition from fully mixed to a 1:1-volume stratification of the two phases in this benchmark. This setting is especially difficult for particle-based solvers to stably simulate. Our method achieves stable and nice stratification in this experiment. As far as we know, our approach is the first particle-based approach that can achieve stable multi-fluid unmixing in such a high-density ratio benchmark.

Triple Dam Break. A triple dam break simulation is demonstrated in Fig.8. The density ratio between the three miscible phases is red:green:blue = 5:600:1000. Our approach can stably simulate the vigorous mixing of multiple high-density-ratio miscible phases.



Fig. 9. Phase separation in a serpentine condenser with the density ratio cyan:orange=1000:1. Natural unmixing effects are obtained with gravity and centrifugal forces.

Table 3. Value of parameters

Parameter(Unit)	Value	Description
$\tau(\text{m}^3 \cdot \text{s} \cdot \text{kg}^{-1})$	$4.0 \times 10^{-6} - 2.0 \times 10^{-5}$	drift velocity parameter
$\sigma(\text{m}^2 \cdot \text{s}^{-1})$	$1.0 \times 10^{-4} - 3.0 \times 10^{-3}$	drift velocity parameter
$\gamma(1)$	1.0×10^{-5}	chemical reaction factor
$\lambda(\text{kg} \cdot \text{m}^{-3})$	1.0×10^2	chemical reaction factor
$\mu(\text{kg} \cdot \text{m}^{-1} \cdot \text{s}^{-1})$	6.0×10^0	viscosity factor

Phase Separation. In Fig.9, a liquid mixture consisting of two miscible phases travels down a serpentine condenser. The phase density ratio is set to cyan:orange=1000:1. Natural unmixing effects are obtained with gravity and centrifugal forces.

Rayleigh-Taylor Instability. In Fig. 10, we demonstrate Rayleigh-Taylor instabilities using both immiscible(left column) and miscible(right column) settings of two phases. Their density ratio is set to yellow:red = 100:1, which is bigger than those adopted in [De Rosis and Coreixas 2020; Losasso et al. 2006; Solenthaler and Pajarola 2008]. The Rayleigh-Taylor instability patterns are successfully reproduced by our approach.

Timing Comparison. In Fig.11, we test the total timing of the method in [Ren et al. 2014](previous method) and our method. The 1:50 density ratio case in Fig.3 is used and we only change the time step sizes. We use the bisection method to find the max time step that the previous method can handle. A very small time step (i.e. 5.5×10^{-5} s) is needed for the previous method to maintain stability under such a density ratio. Our method is stable at a time step of

1.5×10^{-3} s. Under these settings, the previous method [Ren et al. 2014] takes 25.8 times longer in total to complete the simulation.

Rainbow Cocktail. Multiple phases are poured into a cocktail glass in Fig.12. In the first row, we sequentially pour three miscible phases into the glass with density ratio red:yellow:cyan=1000:500:200. The liquid forms a stable rainbow cocktail appearance within the glass. In the second row, we first pour the same two phases into the glass, but replace the cyan phase with a green phase and allow the yellow and green phases react producing a new blue phase whose density is set to $\frac{1}{100}$ of the yellow phase. Vigorous mixing is observed in the upper region.

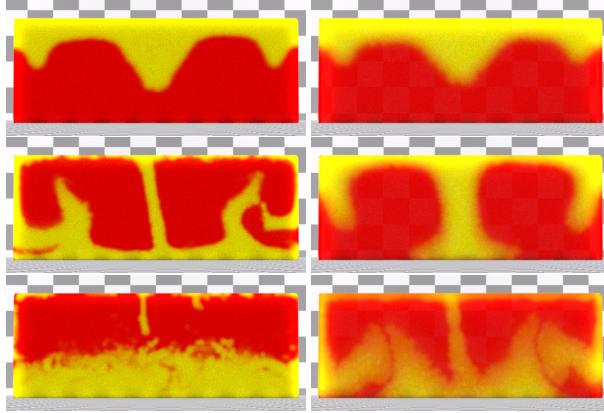


Fig. 10. Rayleigh-Taylor instability of miscible and immiscible fluid. The density ratio of red:yellow is 1:100. The multiphase fluid in the left-hand column is immiscible, while the one in the right-hand column is miscible.

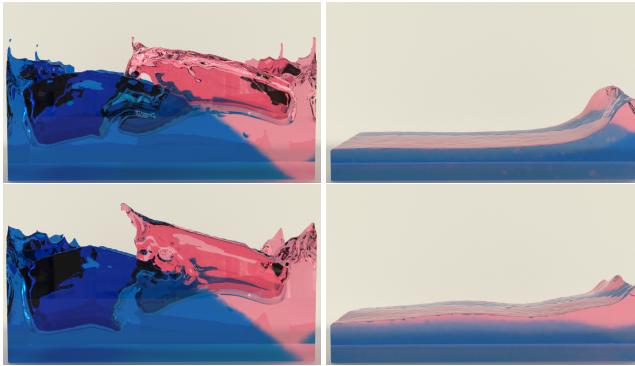


Fig. 11. Timing comparison of method in [Ren et al. 2014] and our method. The 1:50 density ratio case in Fig.3 is used and we only change the time step sizes. The first row is method in [Ren et al. 2014](previous method), and the second row is our method. We use the bisection method to find the max time step that the previous method can handle and test the total timing of the previous method and our method. The previous method takes 25.8 times longer in total to complete the simulation.

"Depth Charge". A very small piece of pink phase falls into a beer glass and undergoes a chemical reaction with the blue phase generating a red phase with small density. Since the decrease in



Fig. 12. Rainbow cocktail. The density ratio is set to red:yellow:cyan=1000:500:200 in the first row and red:yellow:green:blue=1000:500:200:5 in the second row. Without reaction, the first row shows stable layering of the three miscible phases forming a rainbow cocktail appearance. In the second row, after pouring the red and yellow phase, we inject another green phase and allow green and yellow phases to react. A newly produced blue phase rises and mixes with the green phase in the upper region.

the density of the product will result in a sudden large increase in volume, a spout like that by a depth charge is immediately formed, and the lighter phases keeps spilling out as the reaction continues. The density ratio in this example is red:orange:pink:blue = 1:10:50:100. High-density-ratio multi-phase simulations have such unique blasting-like effects, which can be stably reproduced by our algorithm.

Rising Bubbles. A pipe continuously injects gas into a rectangular water tank and produces bubbles. The density ratio of gas to water is 1:100. In this example, transparent gas-phase particles are set above the liquid surface. We color the uniform-density liquid phase with three different colors for better observations of flow motions. Realistic air-liquid coupling behaviors can be observed.

7 CONCLUSION

In this paper, we present a novel particle-based simulation framework. By deriving a peridynamic mixture-model fluid theory that solves the multi-fluid N-S equations in integral forms, our approach can simulate high-density ratio multi-fluid mixing, unmixing effects and chemical reaction. We also propose a novel mass updating strategy that enhances phase-wise mass conservation and reduces particle volume variations in the simulation.

Our current framework solves the fluid pressure in an explicit way, and combining the PMF model with recent incompressible multi-fluid solvers (e.g. [Ren et al. 2021a]) should bring further benefits to the visual quality. The explicit Tait equation of state for pressure calculations also limits the stability in high-speed impact situations and may cause slight disorder of particles at phase interface and sticking particles near the boundary, which are most obvious in the Rayleigh-Taylor case. Currently no significant benefit to stiffness choice in explicit solvers is observed due to peridynamic formulations. These limitations should be alleviated with an implicit solver. Due to the different particle spacing between different phases,



Fig. 13. "Depth charge". A very small piece of pink phase falls into a beer glass and undergoes a chemical reaction with the blue phase generating a red phase with small density. Due to the high density ratio of the blue phase to red phase, such a small piece of pink phase will produce a large volume of red phase in a short period of time, thus the red phase will flush away the orange phase located above it. The density ratio is red:orange:pink:blue = 1:10:50:100.

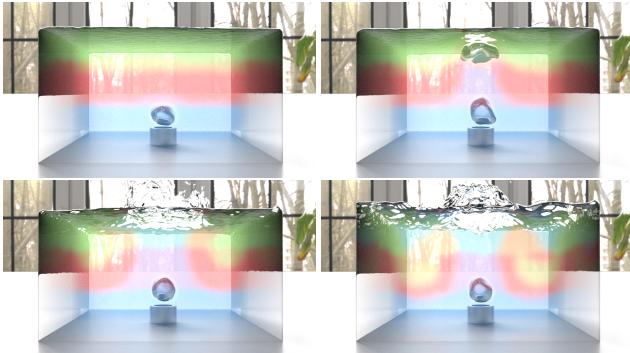


Fig. 14. A pipe continuously injects gas into a rectangular water tank and produces bubbles. The density ratio of gas to water is 1:100. The water is labeled with red, blue and green based on the position of particles in the initial state. Realistic air-liquid coupling behaviors can be observed.

the surface reconstruction in final rendering can still appear a bit bumpy for the lightest phase when the density ratio is very high, which may be enhanced by splitting the large-volume particles to smaller ones at a cost of computational complexity. Although not observed in existing engineering practices, the integral forms may potentially bring larger numerical kernel sizes in the calculation

and have a theoretical limitation of additional smoothing and worth investigation in later works. In addition, including energy-based effects such as distilling into the current model deserves investigation in the future.

ACKNOWLEDGMENTS

This work was supported by the Natural Science Foundation of China 62272245, the Fundamental Research Funds for the Central Universities (Nankai University) 63233080 and the Open Project Program of the State Key Lab of CAD&CG, Zhejiang University(Grant No. A2207). We thank the anonymous reviewers for suggestions.

REFERENCES

- N. Akinci, M. Ihmsen, G. Akinci, Barbara Solenthaler, and M. Teschner. 2012. Versatile Rigid-Fluid Coupling for Incompressible SPH. *AcM Transactions on Graphics* 31, 4CD (2012), 1–8.
- S. Band, C. Gissler, M. Ihmsen, J. Cornelis, A. Peer, and M. Teschner. 2018. Pressure Boundaries for Implicit Incompressible SPH. *ACM Transactions on Graphics (TOG)* 37, 2 (2018), 1–11.
- Markus Becker and Matthias Teschner. 2007. Weakly compressible SPH for free surface flows. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 209–217.
- Landon Boyd and Oberrt Bridson. 2012. MultiFLIP for energetic two-phase fluid simulation. *AcM Transactions on Graphics* 31, 2 (2012), 1–12.
- Wei Chen, Fei Zhu, Jing Zhao, Sheng Li, and Guoping Wang. 2018. Peridynamics-Based Fracture Animation for Elastoplastic Solids. 37, 1 (2018), 112–124.
- Alessandro De Rosia and Christophe Coreixas. 2020. Multiphysics flow simulations using D3Q19 lattice Boltzmann methods based on central moments. *Physics of Fluids* 32, 11 (2020).
- Thomas Fonty, Martin Ferrand, Agnès Leroy, Antoine Joly, and Damien Violeau. 2019. Mixture model for two-phase flows with high density ratios: A conservative and realizable SPH formulation. *International Journal of Multiphase Flow* 111 (2019), 158–174.
- Thomas Fonty, Martin Ferrand, Agnès Leroy, and Damien Violeau. 2020. Air entrainment modeling in the SPH method: a two-phase mixture formulation with open boundaries. *Flow, Turbulence and Combustion* 105, 4 (2020), 1149–1195.
- Georg C. Ganzenmüller, Stefan Hiermaier, and Michael May. 2015. Improvements to the Prototype Micro-brittle Model of Peridynamics. In *Meshfree Methods for Partial Differential Equations VII*, Michael Griebel and Marc Alexander Schweitzer (Eds.). Springer International Publishing, Cham, 163–183.
- Ming Gao, Andre Pradhan, 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. <https://doi.org/10.1145/3197517.3201309>
- Xiaowei He, Huamin Wang, and Enhua Wu. 2017. Projective peridynamics for modeling versatile elastoplastic materials. *IEEE transactions on visualization and computer graphics* 24, 9 (2017), 2589–2599.
- Jeong-Mo Hong and Chang-Hun Kim. 2005. Discontinuous fluids. *ACM Transactions on Graphics (TOG)* 24, 3 (2005), 915–920.
- C Huang, JM Lei, MB Liu, and XY Peng. 2015. A kernel gradient free (KGF) SPH method. *International Journal for Numerical Methods in Fluids* 78, 11 (2015), 691–707.
- Markus Ihmsen, Julian Bader, Gizem Akinci, and Matthias Teschner. 2011. Animation of air bubbles with SPH. In *International Conference on Computer Graphics Theory and Applications*, Vol. 2. SCITEPRESS, 225–234.
- Y. Jiang and Y. Lan. 2021. A Dynamic Mixture Model for Non-equilibrium Multiphase Fluids. *Computer Graphics Forum* 40 (10 2021), 85–95. <https://doi.org/10.1111/cgf.14403>
- 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 (2010), 685–694.
- Dan Koschier, Jan Bender, Barbara Solenthaler, Matthias Teschner, Michael Bronstein, Leonidas Guibas, Iasonas Kokkinos, Or Litany, Niloy Mitra, Federico Monti, et al. 2019. Smoothed Particle Hydrodynamics Techniques for the Physics Based Simulation of Fluids and Solids. *Eurographics 2019-Tutorials* (2019), 47–51.
- Joshua A Levine, Adam W Bargteil, Christopher Corsi, Jerry Tessendorf, and Robert Geist. 2014. A Peridynamic Perspective on Spring-Mass Fracture.. In *Symposium on Computer Animation*. 47–55.
- W. Li, D. Liu, M. Desbrun, J. Huang, and X. Liu. 2020. Kinetic-based Multiphase Flow Simulation. *IEEE Transactions on Visualization and Computer Graphics* 27, 7 (2020),

- 3318–3334.
- Frank Losasso, Tamar Shinar, Andrew Selle, and Ronald Fedkiw. 2006. Multiple Interacting Liquids. *ACM Trans. Graph.* 25, 3 (jul 2006), 812–819. <https://doi.org/10.1145/1141911.1141960>
- Olivier Mercier, Cynthia Beauchemin, Nils Thuerey, Theodore Kim, and Derek Nowrouzezahrai. 2015. Surface turbulence for particle-based liquid simulations. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–10.
- 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*, 237–244.
- Simon Premoze, Tolga Tasdizen, James Bigler, Aaron Lefohn, and Ross T Whitaker. 2003. Particle-based simulation of fluids. In *Computer Graphics Forum*, Vol. 22. Wiley Online Library, 401–410.
- B. Ren, W. He, C. Li, and X. Chen. 2021a. Incompressibility Enforcement for Multiple-fluid SPH Using Deformation Gradient. *IEEE transactions on visualization and computer graphics* PP (2021), 1–11.
- B. Ren, C. Li, X. Yan, M. C. Lin, J. Bonet, and S. M. Hu. 2014. Multiple-Fluid SPH Simulation Using a Mixture Model. *ACM Transactions on Graphics (TOG)* 33, 5 (2014), 1–11.
- Bo Ren, Ben Xu, and Chenzheng Li. 2021b. Unified Particle System for Multiple-Fluid Flow and Porous Material. *ACM Trans. Graph.* 40, 4, Article 118 (jul 2021), 14 pages. <https://doi.org/10.1145/3450626.3459764>
- S. A. Silling. 2000. Silling, S.: Reformulation of elasticity theory for discontinuities and long-range forces. *J. Mech. Phys. Solids* 48, 175–209. *Journal of the Mechanics and Physics of Solids* 48, 1 (2000), 175–209.
- S. A. Silling and E. Askari. 2005. A meshfree method based on the peridynamic model of solid mechanics. *Comput. & Structures* 83, 17/18 (2005), 1526–1535.
- S. A. Silling, M. Epton, O. Weckner, J. Xu, and E. Askari. 2007. Peridynamic States and Constitutive Modeling. *Journal of Elasticity* 88, 2 (2007), 151–184.
- 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.
- Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. 2013. A material point method for snow simulation. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–10.
- 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. <https://doi.org/10.1145/3072959.3073651>
- Y. Tao, C. Jian, M. C. Lin, R. R. Martin, and S. M. Hu. 2017. A unified particle system framework for multi-phase, multi-material visual simulations. *Acm Transactions on Graphics* 36, 6 (2017), 1–13.
- Quang Van Le and Florin Popovici. 2018. Objectivity of state-based peridynamic models for elasticity. *Journal of Elasticity* 131 (2018), 1–17.
- Han Wang, Yepeng Xu, and Dan Huang. 2019. A non-ordinary state-based peridynamic formulation for thermo-visco-plastic deformation and impact fracture. *International Journal of Mechanical Sciences* 159 (2019), 336–344.
- Xiao Yan, Yun-Tao Jiang, Chen-Feng Li, Ralph R. Martin, and Shi-Min Hu. 2016a. Multiphase SPH Simulation for Interactive Fluids and Solids. *ACM Trans. Graph.* 35, 4, Article 79 (jul 2016), 11 pages. <https://doi.org/10.1145/2897824.2925897>
- X. Yan, Y. T. Jiang, C. F. Li, R. R. Martin, and S. M. Hu. 2016b. Multiphase SPH simulation for interactive fluids and solids. *ACM Transactions on Graphics* 35, 4 (2016), 1–11.
- X. Yan, C.-F. Li, X.-S. Chen, and S.-M. Hu. 2018. MPM simulation of interacting fluids and solids. *Computer Graphics Forum* 37, 8 (2018), 183–193.
- T. Yang, J. Chang, B. Ren, M. C. Lin, J. J. Zhang, and Shi Min Hu. 2015. Fast Multiple-fluid Simulation Using Helmholtz Free Energy. *Acm Transactions on Graphics* 34, 6 (2015), 1–11.
- Jihui Yu and Greg Turk. 2013. Reconstructing Surfaces of Particle-Based Fluids Using Anisotropic Kernels. *ACM Trans. Graph.* 32, 1, Article 5 (feb 2013), 12 pages. <https://doi.org/10.1145/2421636.2421641>
- Hongbin Zhu, Kai Bao, Enhua Wu, and Xuehui Liu. 2007. Stable and efficient miscible liquid-liquid interactions. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, 55–64.
- H. Zhu, X. Liu, Y. Liu, and E. Wu. 2006. Simulation of miscible binary mixtures based on lattice Boltzmann method. *Computer Animation and Virtual Worlds* 17, 3-4 (2006), 403–410.

A DERIVATION OF EQUATION (15)

First, we supplement some additional definitions in peridynamics that are used to derive the volume flow state.

Definition 5 The point product of two states $\underline{A} \in \mathcal{A}_{m+p}$ and $\underline{B} \in \mathcal{A}_p$ is a state in \mathcal{A}_m defined by

$$(\underline{AB})_{i_1 \dots i_m} (\xi) = \underline{A}_{i_1 \dots i_m j_1 \dots j_p} (\xi) \underline{B}_{j_1 \dots j_p} (\xi) \quad \forall \xi \in \mathcal{H}. \quad (47)$$

The point product is commutative if the two states are of the same order.

Definition 6 The dot product of two states \underline{A} and \underline{B} is defined by

$$\underline{A} \bullet \underline{B} = \int_{\mathcal{H}} (\underline{AB}) (\xi) dV_{\xi} \quad (48)$$

Definition 7 Let Ψ be a function of a state, $\Psi(\cdot) : \mathcal{A}_m \rightarrow \mathcal{L}_n$. Suppose there exists a state-valued function denoted $\nabla \Psi \in \mathcal{A}_{m+n}$ such that for any $\underline{A} \in \mathcal{A}_m$ and any $\Delta \underline{A} \in \mathcal{A}_m$,

$$\Psi(\underline{A} + \Delta \underline{A}) = \Psi(\underline{A}) + \nabla \Psi(\underline{A}) \bullet \Delta \underline{A} + o(\|\Delta \underline{A}\|). \quad (49)$$

Then Ψ is said to be differentiable and $\nabla \Psi$ is called the Frechet derivative of Ψ .

Let $J = J(\underline{Y})$ be the ratio of the volume in the reference state to the one in the deformation state, which can be expressed as:

$$J = \frac{V}{V_0} = \det \mathbf{F}, \quad (50)$$

where \mathbf{F} is the deformation gradient tensor, and \underline{Y} is the deformation vector state defined by

$$\underline{Y}[x, t](\xi) = \mathbf{y}(x + \xi, t) - \mathbf{y}(x, t), \quad (51)$$

and

$$\mathbf{y}(x, t) = \mathbf{x} + \mathbf{u}(x, t) \quad (52)$$

is the position of a particle in the current state. let $\mathbf{y}' = \mathbf{y}(x + \xi, t)$, Using the Frechet derivative in Def. 7, we obtain

$$\Delta J = \nabla J(\underline{Y}) \bullet \Delta \underline{Y} = \int_{\mathcal{H}} [\nabla J(\underline{Y}) \Delta \underline{Y}] (\xi) dV_{\xi}. \quad (53)$$

Note that

$$\Delta \underline{Y}(\xi) = \Delta \mathbf{y}' - \Delta \mathbf{y} = \left(\frac{\partial \mathbf{y}'}{\partial \mathbf{x}} - \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \right) \Delta \mathbf{x} + \left(\frac{\partial \mathbf{y}'}{\partial t} - \frac{\partial \mathbf{y}}{\partial t} \right) \Delta t. \quad (54)$$

where

$$\Delta \mathbf{y} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \Delta \mathbf{x} + \frac{\partial \mathbf{y}}{\partial t} \Delta t. \quad (55)$$

The first term in Equation (55) corresponds to the deformation along a certain direction, whose difference in Equation (54) becomes a second-order derivative. In our experiments, we also find the second term in Equation (54) predominates, thus we approximate

$$\Delta \underline{Y}(\xi) = \left(\frac{\partial \mathbf{y}'}{\partial t} - \frac{\partial \mathbf{y}}{\partial t} \right) \Delta t = (\mathbf{v}_m' - \mathbf{v}_m) \Delta t. \quad (56)$$

Let $\underline{V}_m \subset \mathcal{V}$ be the velocity vector state so that

$$\underline{V}_m(\xi) = \mathbf{v}_m. \quad (57)$$

According to Def. 5 and Equation (53, 56), we can express $\Delta J/J$ in:

$$\frac{1}{J} \int_{\mathcal{H}} [\nabla J(\underline{Y}) \Delta \underline{Y}] (\xi) dV_{\xi} = \Delta t \int_{\mathcal{H}} \frac{1}{J} \left(\underline{V}_m' \nabla J(\xi) - \underline{V}_m \nabla J(\xi) \right) dV_{\xi}. \quad (58)$$

This falls in line with the first term on the right hand side of Equation (13), with

$$\frac{\Delta V}{V} = \frac{V_0}{V} \Delta \left(\frac{V}{V_0} \right) = \frac{\Delta J}{J} = \Delta t \int_{\mathcal{H}} \frac{1}{J} \left(\underline{V}_m' \nabla J(\xi) - \underline{V}_m \nabla J(\xi) \right) dV. \quad (59)$$

From the above deviations, we can obtain the definition of the volume flow state as with obtaining \underline{T} in [Silling et al. 2007]:

$$\underline{s} \langle \xi \rangle = \frac{1}{J} V_m \nabla J \langle \xi \rangle = \frac{\boldsymbol{v}_m \cdot \nabla J}{J}, \quad (60)$$

where

$$\boldsymbol{v}_m \cdot \nabla J = v_{m,i} \nabla J_i. \quad (61)$$

As for the volume flow state of phase k corresponding to the second term on the right hand side of Equation (13), we can simply replace Equation (50) by

$$J_k = \frac{V_k}{V_{k0}}, \quad (62)$$

and similarly define

$$\underline{s}_k \langle \xi \rangle = \frac{\alpha_k \boldsymbol{v}_k \cdot \nabla J_k}{J_k}, \quad (63)$$

from the second term on the right-hand side of Equation (13), where J_k is the volume ratio of the phase k , and $\boldsymbol{v}_m, \boldsymbol{v}_k$ is the mixture velocity and phase velocity, respectively. Equation (63) can be further transformed into a more calculation-friendly form. We consider that the state \underline{s}_k is defined within a meta volume (i.e. the meta volume serves as its \mathcal{H} , where \boldsymbol{v}_m and α keep constant spatially). Within \mathcal{H} , we can evaluate $\frac{dV_k}{dt}$ from the absolute reference system where

$$\frac{dV_k}{dt} = \frac{\partial V_k}{\partial \xi} \cdot \frac{d\xi}{dt} + \frac{\partial V_k}{\partial t} \quad (64)$$

and evaluate it from the reference system of the relative center of mass where

$$\frac{dV_k}{dt} = \frac{\partial V_k}{\partial \xi} \cdot \frac{\partial \xi}{\partial \xi'} \cdot \frac{d\xi'}{dt} + \frac{\partial V_k}{\partial t}. \quad (65)$$

Since $\xi = \xi' + \boldsymbol{v}_m t$, $\frac{d\xi}{dt} = \boldsymbol{v}_k$ and $\frac{d\xi'}{dt} = \boldsymbol{v}_{mk}$, we have

$$\frac{\partial \xi}{\partial \xi'} = 1 + \frac{\partial \boldsymbol{v}_m t}{\partial \xi'} = 1 \quad (66)$$

and

$$\boldsymbol{v}_k \cdot \nabla V_k = \boldsymbol{v}_{mk} \cdot \nabla' (\alpha_k V), \quad (67)$$

Where $\nabla' = \frac{\partial}{\partial \xi'} = \nabla \cdot \frac{\partial \xi}{\partial \xi'}$ is the Nabla operator in the reference system of the relative center of mass. As is discussed above, α_k stays constant in a meta volume, so the Equation (67) is rewritten as

$$\boldsymbol{v}_k \cdot \nabla V_k = \alpha_k \boldsymbol{v}_{mk} \cdot \nabla' V. \quad (68)$$

Since $\frac{\nabla V}{V} = \frac{\nabla J}{J}$ and $V_k = \alpha_k V$, we get

$$\boldsymbol{v}_k \cdot \frac{\nabla J_k}{J_k} = \boldsymbol{v}_{mk} \cdot \frac{\nabla J}{J}. \quad (69)$$

As a result, the phase volume state \underline{s}_k can be rewritten as

$$\underline{s}_k \langle \xi \rangle = \frac{\alpha_k \boldsymbol{v}_{mk} \cdot \nabla J}{J}. \quad (70)$$

Similar to the derivations leading to the force vector state calculation (Equation (11)) in [Silling et al. 2007], the volume flow states in Equation (15) can be computed with the following scheme. First, the Frechet derivative of J can be written as

$$\nabla_{\underline{Y}} J(\underline{Y}) = \nabla_{\underline{Y}} [\det \mathbf{F}(\underline{Y})] = \left(\nabla_{\underline{Y}} \mathbf{F} \right) \nabla_{\mathbf{F}} \det \mathbf{F}. \quad (71)$$

According to matrix differentiation,

$$\nabla_{\mathbf{F}} \det \mathbf{F} = \det \mathbf{F} (\mathbf{F}^{-1})^T \quad (72)$$

is a second order tensor. In [Silling et al. 2007], the state $\nabla_{\underline{Y}} \mathbf{F}$ is written as

$$\nabla_{\mathbf{F}} f_{ijq} \langle \xi \rangle == \delta_{iq} \omega \langle \xi \rangle K_{jw}^{-1} \xi_w, \quad (73)$$

where δ is the Kronecker delta function. Using Equation(60), Equation(71-73) and Equation(50), we get the components of the volume flow state (the computable form), which is written as

$$\underline{s} = \frac{1}{J} v_i \nabla J_i = v_i \left(\nabla_{\mathbf{F}} f_{ijq} \right) F_{qj}^{-1} = \underline{\omega} v_i \delta_{iq} F_{qj}^{-1} K_{jw}^{-1} \xi_w = \underline{\omega} v_i F_{ij}^{-1} K_{jw}^{-1} \xi_w, \quad (74)$$

where the subscript i, j, q and w denote the component of the tensor. This directly corresponds to the matrix forms of the volume states as in Equation (16) and Equation (17).

B DERIVATION OF EQUATION (18)

We start from the phase-wise momentum equation of the multi-phase N-S equations. Similar to Equation (10) in §3.2, the momentum equation for a single phase k can be written as

$$\alpha_k \rho_k \frac{d\boldsymbol{v}_k}{dt} = \int_{\mathcal{H}} f_k dV_{\xi} + \alpha_k \rho_k \mathbf{g} + \Gamma_k, \quad (75)$$

where f_k is the force density on the phase k , and Γ_k is the interaction between phases. Note that f_k includes pressure force, viscous force, surface tension and so on. The term Γ_k accounts for the interactions between phases, such as drag and frictional forces. According to the mixture model [Ren et al. 2014], by summing all momentum equations over phases, Γ_k can be cancelled and we get

$$\sum_k \alpha_k \rho_k \frac{d\boldsymbol{v}_k}{dt} = \sum_k \int_{\mathcal{H}} f_k dV_{\xi} + \rho_m \mathbf{g}. \quad (76)$$

In order to get the left-hand side, firstly we use the chain rule:

$$\sum_k \frac{d(\alpha_k \rho_k \boldsymbol{v}_k)}{dt} = \frac{d(\rho_m \boldsymbol{v}_m)}{dt} = \frac{d\rho_m}{dt} \boldsymbol{v}_m + \rho_m \frac{d\boldsymbol{v}_m}{dt}. \quad (77)$$

Since ρ_k stays the same during the simulation, then we can rearrange Equation (77) into

$$\sum_k \alpha_k \rho_k \frac{d\boldsymbol{v}_k}{dt} = \frac{d\rho_m}{dt} \boldsymbol{v}_m + \rho_m \frac{d\boldsymbol{v}_m}{dt} - \sum_k \frac{d\alpha_k}{dt} \rho_k \boldsymbol{v}_k. \quad (78)$$

By substituting Equation (76) into Equation (78), we get

$$\frac{d\boldsymbol{v}_m}{dt} = \sum_k \int_{\mathcal{H}} f_k dV_{\xi} + \mathbf{g} + \sum_k \left(\frac{d\alpha_k}{dt} \frac{\rho_k \boldsymbol{v}_k}{\rho_m} \right) - \frac{1}{\rho_m} \frac{d\rho_m}{dt} \boldsymbol{v}_m. \quad (79)$$

Note that for the last term in Equation (79),

$$\frac{1}{\rho_m} \frac{d\rho_m}{dt} \boldsymbol{v}_m = \sum_k \left(\frac{d\alpha_k}{dt} \frac{\rho_k \boldsymbol{v}_m}{\rho_m} \right) = \sum_k \left[\frac{d\alpha_k}{dt} \frac{\rho_k (\boldsymbol{v}_k - \boldsymbol{v}_{mk})}{\rho_m} \right]. \quad (80)$$

Equation (79) becomes:

$$\frac{d\boldsymbol{v}_m}{dt} = \sum_k \int_{\mathcal{H}} f_k dV_{\xi} + \mathbf{g} + \sum_k \left(\frac{d\alpha_k}{dt} \frac{\rho_k \boldsymbol{v}_{mk}}{\rho_m} \right). \quad (81)$$

As is done to the single-phase force integral in [Silling et al. 2007], we can replace the first term using the same force vector state form as in Equations (10,11):

$$\frac{d\boldsymbol{v}_m}{dt} = \frac{1}{\rho_m} \int_{\mathcal{H}} (\underline{T} \langle \xi \rangle - \underline{T}' \langle \xi \rangle) dV_{\xi} + \mathbf{g} + \sum_k \left(\frac{d\alpha_k}{dt} \frac{\rho_k \boldsymbol{v}_{mk}}{\rho_m} \right). \quad (82)$$