DOI: 10.1111/cgf.14403
Pacific Graphics 2021
E. Eisemann, K. Singh, and F.-L Zhang
(Guest Editors)

Volume 40 (2021), Number 7

A Dynamic Mixture Model for Non-equilibrium Multiphase Fluids

Y. Jiang[1] and Y. Lan[2]

[1]Tsinghua University, China
[2]Peking University, China

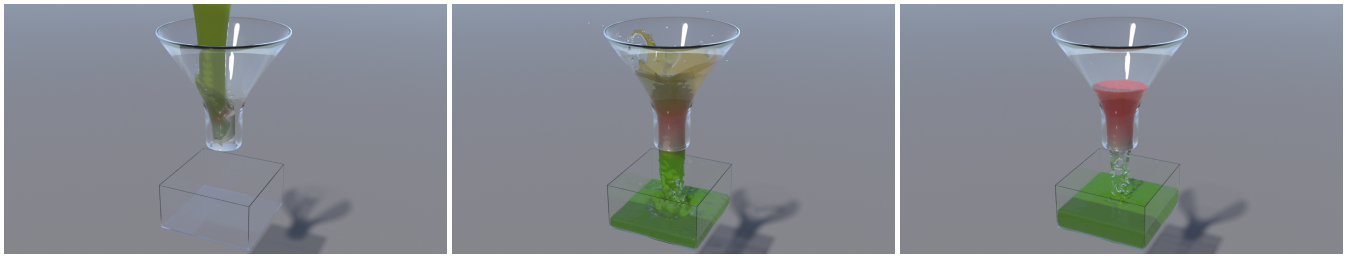

Figure 1: *Water filtration. A mixture of two fluids is poured into a funnel, where only the green phase is allowed to pass through the filter. The red phase stays in the funnel, separating from the green phase.*

Abstract

We present a dynamic mixture model for simulating multiphase fluids with highly dynamic relative motions. The previous mixture models assume that the multiphase fluids are under a local equilibrium condition such that the drift velocity and the phase transport can be computed analytically. By doing so, it avoids solving multiple sets of Navier-Stokes equations and improves the simulation efficiency and stability. However, due to the local equilibrium assumption, these approaches can only deal with tightly coupled multiphase systems, where the relative speed between phases are assumed stable. In this work we abandon the local equilibrium assumption, and redesign the computation workflow of the mixture model to explicitly track and decouple the velocities of all phases. The phases still share the same pressure, with which we enforce the incompressibility for the mixture. The phase transport is calculated with drift velocities, and we propose a novel correction scheme to handle the transport at fluid boundaries to ensure mass conservation. Compared with previous mixture models, the proposed approach enables the simulation of much more dynamic scenarios with negligible extra overheads. In addition, it allows fluid control techniques to be applied to individual phases to generate locally dynamic and visually interesting effects.

CCS Concepts

• ***Computing methodologies*** → ***Physical simulation;***

1. Introduction

Multiphase fluids are mixtures composed by multiple fluids that are miscible or immiscible with each other, and are widely encountered in both daily life and industries. Among others, examples include making a cup of latte with coffee and milk, and mixing paints of different colors in a palette to create more colors. Due to the important applications in industries, especially for energy, manufacturing and chemical engineering sectors, multiphase fluids have been extensively studied by using both experiments and numerical simulations. However, due to the wide range of materials that can be involved in multiphase systems and the associated complex microstructure, the simulation of multiphase fluids still poses great challenges for researchers.

Multiphase fluids can be modelled in different ways, which can be roughly divided into two categories. The first group are the multiphase models, where each phase is modeled with an individual representation, for example a grid or a set of particles. The continuity and momentum equations of each phase are solved separately, and the interactions between the phases are modelled explicitly as forces or boundary conditions. Classic approaches in this group include the two-fluid model (or the Eulerian-Eulerian model), where both phases are treated as separate fluids described

by its own Navier-Stokes equations, and the Eulerian-Lagrangian model, where the continuous phase is treated as a fluid described by Navier-Stokes equations and the dispersed phase is treated as discrete particles described by Newton's law.

The second group of methods to simulate multiphase fluids are the mixture models, where the mixture is modeled as a single continuum that contains many component phases. The motion of the mixture is characterized by a mixture velocity, while the velocity of each phase is described by a relative velocity to the mixture, which is called drift velocity. The transport of individual phases inside the mixture can then be computed by combining the drift and mixture velocities. This approach simplifies the modeling by using a single representation, and also saves the computation by solving only one set of momentum equations. These methods have also been successfully applied in graphics, and have shown their advantages in the stability and efficiency.

However, the previous mixture models have limitations in handling mixtures with dynamic motions between the phases. The reason lies in the constitutive model of the drift velocity [RLY*14], which assumes the phases of the mixture are under a local equilibrium state. In such state, the drift velocity is their terminal relative velocity, and can be determined by the drag force and computed analytically. This assumption works well for those strongly-coupled multiphase systems, such as fluid mixtures or stable suspensions. However, in loosely-coupled systems, the phases may never reach an equilibrium state, and the inertia-dominated relative motion could not be captured by this model. The following works [YJL*16, JLDH20] use the same technique for computing the drift velocity and therefore share the same limitation. It is noted that the model in [YCR*15] uses a chemical potential instead of the drift velocity for the phase transport, which completely ignores the relative motion between different phases.

To solve this problem, we propose a novel dynamic non-equilibrium mixture model that does not require the local equilibrium assumption. In this model, the velocity of each phase is explicitly tracked, and the mixture velocity is computed as a volume-weighted average of the phase velocities. Like the original mixture models, different phases share the same pressure, from which the incompressibility condition can be enforced. The drift velocity of each phase is calculated directly by its definition, and effectively captures the dynamic relative motion of the phases. The main contributions of this work include:

- A dynamic non-equilibrium mixture model that can capture a much wider range of relative motions between the component phases, and therefore much more dynamic visual effects.
- A robust phase transport scheme that corrects the transport at the fluid boundaries to ensure mass conservation, controlling the error to below 0.01%.
- A simple and flexible way to edit multiphase flows through control particles and force fields, enabling separate manipulation on individual component phases.

2. Related work

2.1. Multiphase fluids

Multiphase fluids have been topical in computer graphics in recent years, and some most relevant studies are briefly recapped in this section to lay the proposed dynamic non-equilibrium mixture model in perspective. For a detailed overview of multiple-fluid simulation we refer to [RYL*18].

For immiscible fluids, the focus of the simulation is handling the discontinuity of physical properties at the interfaces between the fluid phases. These including the early grid-based works [HK05, LSSF06], the more recently hybrid methods such as FLIP [BB12], and the lattice Boltzmann method [LLD*20] etc. For miscible fluids, the phase transport in the mixture includes both mixing and separation processes. The mixing process is relatively easy to simulate, and was initially treated as diffusion in the early works [KPNS10, LLP11]. For multiphase systems involving fluids and suspending phases, two-fluid approaches have been used to simulate foam and water spray [NO13, MMS09], particle-laden flows in sand-water mixtures and fluid-solid coupling systems [TGK*17, GPH*18, FQL*20].

To fully capture the behavior of multiphase fluids, [RLY*14] introduced the mixture model with a SPH implementation, which successfully simulates the mixing and unmixing with the drift velocity. Alternatively, [YCR*15] proposed a Helmholtz energy based framework to guide the phase transport, providing direct control of target mass fractions and capturing complex phenomena such as fluid extraction. [YJL*16] extended the mixture model in [RLY*14] to simulate fluid-solid coupling, where the solid diffusion is treated uniformly with the phase diffusion. [YCL*17] later introduced phase-fields to the model in [YCR*15] to capture the phase transition. Recently [JLDH20] proposed a divergence-free mixture model based on the model in [RLY*14], enabling the use of incompressible solvers in simulating multiphase fluids.

As stated above, the previous mixture models either depend on the local equilibrium assumption or ignore the relative motion completely, failing to capture dynamic motions between phases. The dynamic non-equilibrium mixture model in this work differs from these models by computing the drift velocity in a completely different way, and fully represents the motions of phases that may or may not be in an equilibrium condition.

2.2. Fluid control

Fluid control is a trending topic that is partially related to this work. Again, only the most relevant works are briefly summarised here, while we refer to recent works such as [LCY*19] for a more detailed overview on this topic. The previous works on fluid control mainly focus on single phase fluids, while multiphase fluids are potentially more challenging to apply control techniques.

A common approach in fluid control is using control particles. To control the shape of fluids, [Mad, TKPR09] exert forces to fluids with control particles to attract the fluid to target shapes. [LCY*19] also uses a control particle system to guide the shape deformation of fluids. [HK04] uses a geometrical potential to coerce the fluids

into target shapes. [YCR*15] modifies the chemical potential with a spatial pattern, producing colorful images with multiphase fluids.

In this work, we combine the control particles and force fields to manipulate the target shape of multiphase fluids, where the force fields are computed based on geometrical potential fields.

3. Mixture Model

In this section we briefly recap the standard mixture model, which serves as the foundation for the proposed dynamic non-equilibrium mixture model. Like the previous works [RLY*14, YJL*16, JLDH20], we explain the mixture model with SPH implementation. It is noted that the mixture model itself is however compatible with other numerical methods such as grid-based or hybrid methods.

In the mixture model, multiple fluids are discretized with one set of SPH particles, where each particle contains multiple component phases, described by their volume fractions and phase velocities, as shown in Fig. 3. The relative velocity of each phase to the mixture is called the drift velocity. Apart from these phase specific properties, the particles share properties with single phase fluid particles, including mass, rest density, pressure and velocity, where the particle velocity corresponds to the mixture velocity and describes the motion of the whole mixture.

Mixture models enforce the incompressibility of the mixture by solving for the pressure on the particles. The mixture velocity in [RLY*14] is not inherently divergence-free, and the pressure is computed with the WCSPH method [BT07]. In the recent divergence-free mixture model [JLDH20], incompressible SPH solvers [BK15, ICS*14] are used instead. Compared to multiphase models, where each phase is solved with a separate set of equations, mixture models can significantly save the computation.

The previous mixture models are based on the local equilibrium assumption, under which the drift velocities are the terminal velocities of the phases relative to the mixture. By using a linear drag force formulation, the drift velocities of the component phases can be estimated directly from the acceleration of the mixture. This approach has proved effective in handling strongly coupled multiphase systems, where the interaction between the phases dominates the behavior of the mixture. Such systems include many liquid-particle mixtures and liquid-liquid mixtures. A common example of these mixtures is the ink, where the ink pigments are tiny particles suspending in water, whose behavior is dominated by the drag force rather than the inertia effect. However, such approach could not capture the behavior of more dynamic scenarios, where the phases are far from reaching their terminal velocity and have much more vigorous relative motion. The previous drift velocity model ignores the inertia effects of the component phases, and fails to describe such movements.

In this work, we solve this problem with a different mixture model that abandons the local equilibrium assumption and focuses on the inertia effects. The workflow of the proposed mixture model is shown in Figure 2, where we also give the comparison between our approach and the previous approaches. The drift velocity definition and the phase transport scheme are retained in the proposed dynamic non-equilibrium mixture model, but the drift and mixture velocities are both computed differently. Specifically, the computation process is redesigned to explicitly track the phase velocities, which allows capturing the inertia effects of phases with negligible overhead. Due to the absence of the local equilibrium, we also need to explicitly compute the drag force between the phases, which directly affects the speed of the mixing and unmixing of different phases. Our method can also be applied to those strongly coupled systems by using a large drag force coefficient, and therefore provides a flexible solution to different multiphase scenarios. We can further apply external forces to manipulate the individual phase velocities, and generate artist-controlled animations of multiphase fluids. The technical details of the new model is explained in the next section.

4. Dynamic Mixture Model

We assume all component phases are incompressible, and the mixing and unmixing of phases do not introduce extra volume change. Under this assumption, we adopt the mixture velocity of the divergence-free mixture model [JLDH20], which is defined as

$$\mathbf{u}_m = \sum_k \alpha_k \mathbf{u}_k, \tag{1}$$

where $\mathbf{u}_m$ is mixture velocity, and α_k and $\mathbf{u}_k$ are the volume fraction and the velocity of phase k, respectively. This mixture velocity is divergence-free by definition, and incompressible solvers are still applicable for our model. In this work we use WCSPH to compute fluid pressure for its simplicity, but it is straightforward to implement an incompressible solver as described in [JLDH20]. The derivation of the momentum equations and the phase transport are explained in detail in § 4.1 and § 5.4, respectively.

4.1. Momentum equations

Following [MTK96], the momentum equation of phase k is given by

$$\alpha_k \rho_k \left(\frac{\partial \mathbf{u}_k}{\partial t} + \mathbf{u}_k \nabla \mathbf{u}_k\right) = -\alpha_k \nabla p_k + \alpha_k \nabla \cdot \boldsymbol{\tau}_k + \alpha_k \rho_k \mathbf{g} + \mathbf{M}_k, \tag{2}$$

where p_k is the fluid pressure, $\boldsymbol{\tau}_k$ the viscosity tensor, $\mathbf{g}$ the gravity, and $\mathbf{M}_k$ the momentum exchange with other phases. The main source of $\mathbf{M}_k$ is the drag force between phases, and it should satisfy the momentum conservation, that is $\sum_k \mathbf{M}_k = \mathbf{0}$. Using a linear drag force model similar to [JLDH20], we define the $\mathbf{M}_k$ as

$$\mathbf{M}_k = -k_d \rho_m \alpha_k (\mathbf{u}_k - \mathbf{u}_m), \tag{3}$$

where k_d is the drag coefficient, and $\rho_m = \sum_k \alpha_k \rho_k$ the mixture density.

The material derivative $\frac{D\mathbf{u}_k}{Dt}$ at a fluid particle is defined using the mixture velocity, which is $\frac{D\mathbf{u}_k}{Dt} = \frac{\partial \mathbf{u}_k}{\partial t} + \mathbf{u}_m \cdot \nabla \mathbf{u}_k$. Here we use an approximation $\mathbf{u}_k \cdot \nabla \mathbf{u}_k \approx \mathbf{u}_m \cdot \nabla \mathbf{u}_k$, indicating that the advection caused by drift velocity is insignificant compared to advection caused by the whole motion. With this assumption we rewrite the left hand side of Eq. (2) with the material derivative and get:

$$\frac{D\mathbf{u}_k}{Dt} \approx -\frac{1}{\rho_k}\nabla p_k + \frac{1}{\rho_k}\nabla \cdot \boldsymbol{\tau}_k + \mathbf{g} + \frac{1}{\alpha_k \rho_k}\mathbf{M}_k. \tag{4}$$

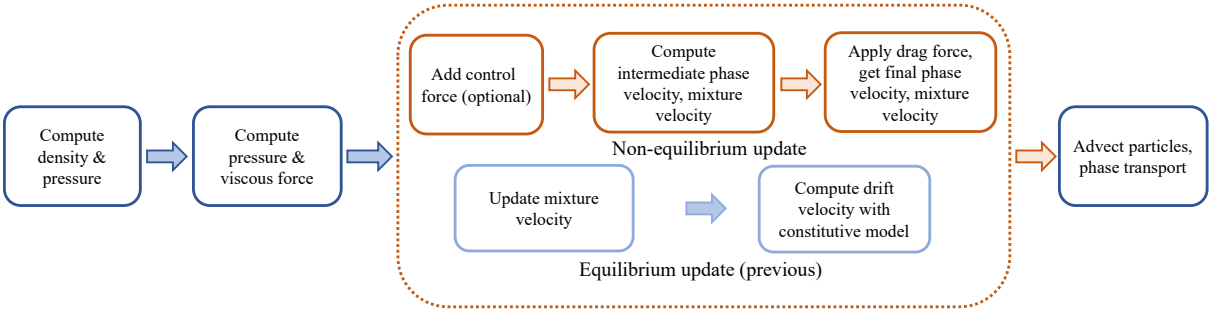

Figure 2: *The pipeline of our non-equilibrium mixture model. The non-equilibrium part is highlighted with orange color, and we also provide the previous equilibrium version for comparison.*

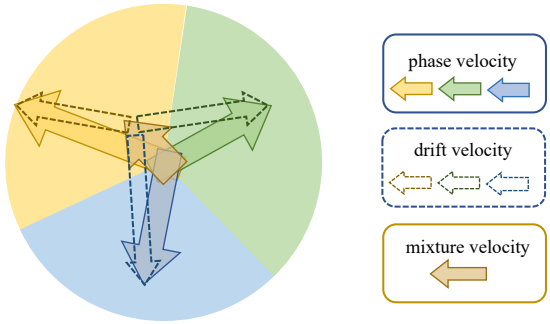

Figure 3: *The particles in a mixture model carries the volume fractions of the component phases, where each phase has its own velocity. The drift velocity is defined as their relative velocities to the mixture.*

With Eq. (4) the phase velocity $\mathbf{u}_k$ can be directly updated. To achieve this, we partly follow the previous mixture models by assuming all phases share the same fluid pressure p and viscosity stress $\boldsymbol{\tau}$, namely $p_k = p$, $\boldsymbol{\tau}_k = \boldsymbol{\tau}$. With these assumptions, and using the mixture velocity definition, we can obtain the following relations:

$$\frac{D\mathbf{u}_k}{Dt} = -\frac{1}{\rho_k}(\nabla p - \nabla\cdot\boldsymbol{\tau}) + \mathbf{g} + \frac{1}{\alpha_k\rho_k}\mathbf{M}_k,$$
$$\frac{D\mathbf{u}_m}{Dt} = -\sum_k \frac{\alpha_k}{\rho_k}(\nabla p - \nabla\cdot\boldsymbol{\tau} - \frac{1}{\alpha_k}\mathbf{M}_k) + \mathbf{g} + \mathbf{M}_t, \quad (5)$$

where $\mathbf{M}_t = \sum_k \mathbf{u}_k \frac{D\alpha_k}{Dt}$ is the velocity change induced by phase transport. Unlike the previous methods, we do not directly update $\mathbf{u}_m$, but update the phase velocities $\mathbf{u}_k$ instead. The mixture velocity is computed with Eq. (1) when needed, as explained in § 5.3.

The velocity updating is split into two parts, namely the predicting part and the pressure part. Since the WCSPH scheme is used for computing pressure in this work, these two parts are computed simultaneously. Note that if an incompressible solver is chosen instead, we can first predict $\mathbf{u}_k$ and $\mathbf{u}_m$, and then compute the pressure p to ensure $\nabla\cdot\mathbf{u}_m = 0$. The implementation detail is given in § 5.3.

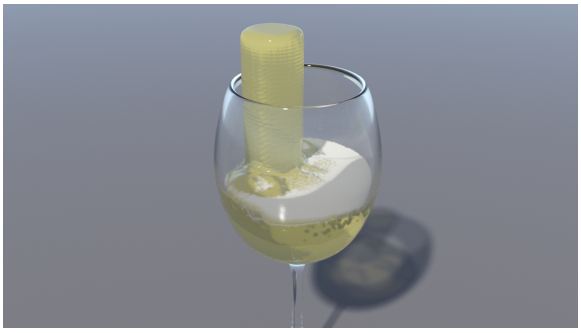

Figure 4: *Pouring bubbly wine. The fluid being injected contains a fluid phase and a lighter gaseous phase. The gaseous phase separates and rises rapidly, forming a foam like layer.*

4.2. Phase transport

After updating the phase velocity $\mathbf{u}_k$ and mixture velocity $\mathbf{u}_m$, we compute the drift velocity $\mathbf{u}_{mk}$ and the phase transport. As in [JLDH20], the drift velocity is defined as

$$\mathbf{u}_{mk} = \mathbf{u}_k - \mathbf{u}_m. \quad (6)$$

Then, the volume fraction of phase k is updated as follows:

$$\frac{D\alpha_k}{Dt} = -\nabla\cdot\alpha_k\mathbf{u}_{mk}. \quad (7)$$

This formulation only captures the transport due to the relative motion of the phases, while in reality the diffusion due to the multiphase turbulence and Brownian motion can also make a substantial contribution to the phase transport. Therefore, we introduce an extra diffusion term to Eq. (7), and obtain

$$\frac{D\alpha_k}{Dt} = -\nabla\cdot\alpha_k\mathbf{u}_{mk} - C_d\nabla^2\alpha_k, \quad (8)$$

where C_d is the diffusion coefficient.

However, using Eq. (8) to update the volume fractions of the particles directly gives undesirable result due to the particle discretization. The phase transport flux between particles, estimated as the weighted divergence of drift velocities, can cause negative values at fluid boundaries. A simplified one dimensional case is shown in

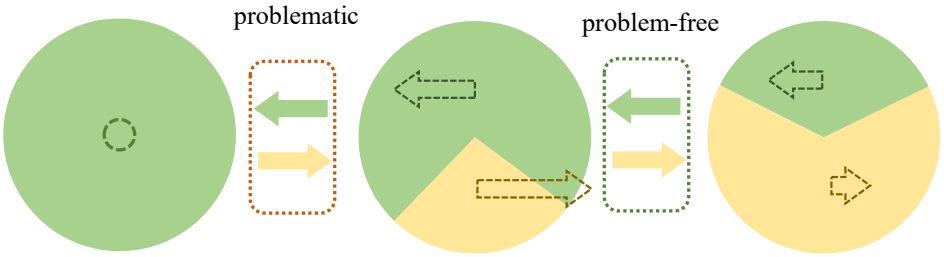

Figure 5: *The phase transport is computed as fluxes between particles. Due to the particle discretization, the fluxes estimated with drift velocities at fluid boundaries can cause negative values, which are marked as problematic in the figure.*

Fig. 5 to demonstrate this problem. In this figure, the leftmost particle does not contain yellow phase, but the flux between the leftmost particle and the middle one can be non-zero. This flux causes the leftmost particle to have a negative volume fraction of yellow phase and a volume fraction of green phase that exceeds one. Clamping these erroneous values back to $[0, 1]$ restores the stability, but breaks the law of mass conservation. These cases are common near boundaries or interfaces between different phases. We propose an iterative correcting scheme that effectively addresses this problem, while keeping the relative error below 0.01%. The details are explained in § 5.4.

5. Implementation

The proposed dynamic non-equilibrium mixture model is implemented with multiphase SPH framework as in previous works, where the main difference between our new model and the previous works lies in the computation of the drift velocity and the mixture velocity. The overall workflow is basically unchanged, making it easy to adapt a previous multiphase framework to use our novel mixture model. The explicit tracking of the phase velocity requires no extra memory consumption and negligible additional computational effort compared to the previous methods, as discussed below.

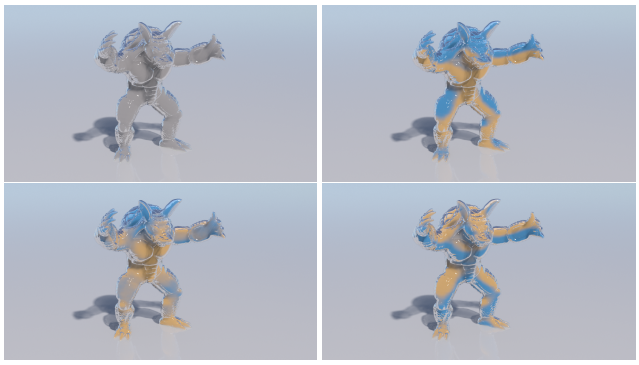

Figure 6: *We simulate the phase transport driven by artificial phase velocity fields in a static armadillo-shaped geometry. By changing the direction of the velocity fields, the transport can be freely modified.*

ALGORITHM 1: Simulation Workflow for Dynamic Non-equilibrium Mixture Model with WCSPH

```
sort particles;
forall particles i do
    estimate density with Eq. (11);
    compute pressure with Eq. (12);
end
forall particles i do
    compute pressure force with Eq. (13);
    compute viscous force with Eq. (14);
end
forall particles i do
    compute intermediate phase velocities with Eq. (15);
    compute mixture velocities;
    apply drag force with Eq. (16);
    compute mixture velocities again;
    advect particles;
end
compute phase transport (Algorithm 2);
```

5.1. Workflow

The simulation workflow of our model contains the following steps:

- Sorting particles with a uniform grid, which is later used for searching neighboring particles.
- Computing particle density with Eq. (11). The pressure is also estimated if using WCSPH.
- Computing acceleration with Eq. (15) and get intermediate phase velocities. The drag force is then applied to get the final phase velocities, as in Eq. (16).
- Updating particle velocities $\mathbf{u}_i$ as the mixture velocity with Eq. (1), and updating particle positions.
- Computing phase transport with an iterative flux correction, as explained in § 5.4.

The simulation workflow is also summarized in Alg. 1 to provide a quick reference for implementation.

5.2. SPH discretization

The basic SPH formulation is briefly recapped below, which is helpful to explain the discretization of momentum equations and phase transport. In SPH method, a property A and its spatial gradi-

ent at position $\mathbf{x}_i$ is estimated by

$$A(\mathbf{x}_i) = \sum_j V_j A_j W(\mathbf{x}_i - \mathbf{x}_j), \tag{9}$$

$$\nabla A(\mathbf{x}_i) = \sum_j V_j A_j \nabla W(\mathbf{x}_i - \mathbf{x}_j), \tag{10}$$

where A_j is the property carried by particle j, V_j denotes the volume of particle j, and W is the SPH kernel function. Note that in the multiphase SPH method, the density of particle i is estimated as

$$\rho_i = m_i \sum_j W_{ij}, \tag{11}$$

where m_i is the mass of particle i. This formulation is proposed by [SP08], and adopted by the previous mixture models.

Using the WCSPH method, the pressure is computed by an equation of state:

$$p_i = \kappa \rho_{m,i} ((\frac{\rho_i}{\rho_{m,i}})^7 - 1), \tag{12}$$

where κ is the stiffness coefficient, and $\rho_{m,i} = \sum_k \alpha_{k,i}\rho_k$ the rest mixture density of particle i. If the estimated pressure is negative, we clamp it to 0 to eliminate the undesired cohesion, as in previous SPH methods.

5.3. Velocity update

The velocity update is performed in a two-step procedure, where in the first step the pressure and viscous force are applied to get an intermediate velocity, and in the second step the drag force is applied to obtain the final velocity. After computing the pressure, we compute the pressure force as

$$\mathbf{F}_i^p = \sum_j V_i V_j (p_i + p_j) \nabla W_{ij}, \tag{13}$$

where the volume of particle i can be estimated with $V_i = \frac{m_i}{\rho_i}$. The viscous force is computed with the artificial viscosity proposed in [BT07]:

$$\mathbf{F}_i^v = -\sum_j m_i m_j \Pi_{ij} \nabla W_{ij},$$
$$\Pi_{ij} = -\frac{2\eta}{\rho_i + \rho_j} \frac{\min(0, \mathbf{u}_{ij} \cdot \mathbf{x}_{ij})}{\mathbf{x}_{ij} \cdot \mathbf{x}_{ij} + \varepsilon h^2}, \tag{14}$$

where $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$ is velocity difference, η the viscosity coefficient, ε a small constant to prevent singularity, and h the SPH smooth radius.

According to Eq. (5), an intermediate phase velocity $\mathbf{u}_k^*$ is first computed as

$$\mathbf{u}_{k,i}^* = \mathbf{u}_{k,i}^n + \left(\frac{1}{\rho_k V_i} (\mathbf{F}_i^p + \mathbf{F}_i^v) + \mathbf{g} \right) \Delta t, \tag{15}$$

where the superscript n denotes the nth time step. The intermediate mixture velocity is then calculated as $\mathbf{u}_i^* = \sum_k \alpha_{k,i}^n \mathbf{u}_{k,i}^*$.

We then apply the drag force to the intermediate velocity to get the phase velocity at $n+1$th time step as

$$\mathbf{u}_{k,i}^{n+1} = \mathbf{u}_{k,i}^* - k_d \Delta t \frac{\rho_{m,i}}{\rho_k} (\mathbf{u}_{k,i}^* - \mathbf{u}_i^*). \tag{16}$$

Finally, the mixture velocity is calculated as $\mathbf{u}_i^{n+1} = \sum_k \alpha_{k,i}^n \mathbf{u}_{k,i}^{n+1}$.

We use an explicit integration of drag force here, as it is normally small in loosely-coupled multiphase systems. For handling arbitrarily large drag coefficients, a backward-Euler integration could be used instead. The time step in our model is mainly restricted by the velocity update. As discussed in [BT07], the time step is related to the speed of sound c_s, and therefore related to the stiffness κ. We found the CFL condition in the original WCSPH can be too strict, and we use a reference value as $\Delta t = c_{CFL} h / c_s$, where $c_s = 88.5\text{m/s}$, and the factor c_{CFL} is set in range $[0.5, 2]$.

5.4. Phase transport

The volume fraction update is discretized as

$$\frac{D\alpha_{k,i}}{Dt} = -(\nabla \cdot \alpha_k \mathbf{u}_{mk})_i - c_d (\nabla^2 \alpha_k)_i \tag{17}$$

where

$$(\nabla \cdot \alpha_k \mathbf{u}_{mk})_i = \sum_j (\alpha_{k,i} \mathbf{u}_{mk,i} + \alpha_{k,j} \mathbf{u}_{mk,j}) V_0 \nabla W_{ij},$$
$$(\nabla^2 \alpha_k)_i = 2 \sum_j (\alpha_{k,i} - \alpha_{k,j}) V_0 \frac{\mathbf{x}_{ij} \cdot \nabla W_{ij}}{\mathbf{x}_{ij} \cdot \mathbf{x}_{ij} + \varepsilon h^2}. \tag{18}$$

The rest volume V_0 is used to ensure the symmetry of the flux between particles, which helps to reduce the influence of fluid compression to the phase transport. These formulations are very similar to those in [JLDH20], and the phase transport is exactly discretized as the volume exchange between the particles.

To solve the aforementioned problem at fluid boundaries, the transport is corrected by assigning a scaling factor λ_i to each particle, and the inter-particle fluxes are adjusted with this factor. The adjusted phase transport is given by

$$\frac{D\alpha_{k,i}}{Dt} = \sum_j \lambda_{ij} (T_{m,ij} + T_{d,ij}),$$
$$T_{m,ij} = -(\alpha_{k,i} \mathbf{u}_{mk,i} + \alpha_{k,j} \mathbf{u}_{mk,j}) V_0 \nabla W_{ij},$$
$$T_{d,ij} = -2 c_d (\alpha_{k,i} - \alpha_{k,j}) V_0 \frac{\mathbf{x}_{ij} \cdot \nabla W_{ij}}{\mathbf{x}_{ij} \cdot \mathbf{x}_{ij} + \varepsilon h^2}, \tag{19}$$

where $\lambda_{ij} = \min(\lambda_i, \lambda_j)$, $T_{m,ij}$ and $T_{d,ij}$ denote the fluxes due to the relative motion and the phase diffusion, respectively.

The maximum flux based on the current volume fraction and the flow state around particle i is controlled by λ_i. For example, in the situation depicted in Fig. 5, we would expect $\lambda = 0$ for the leftmost particle. By picking the minimum value between λ_i and λ_j, we ensure the flux is valid for both particles.

An iterative approach is adopted to compute λ. Before the iteration starts, we set $\lambda_i = 1$ for each particle. During each iteration, we predict the volume fraction as

$$\tilde{\alpha}_{k,i}^l = \alpha_{k,i}^n + \Delta t (\sum_j \lambda_{ij}^l (T_{m,ij}^n + T_{d,ij}^n)), \tag{20}$$

where the superscript l denotes the lth iteration step.

If no negative volume fraction occurs, the iteration ends. Otherwise, if for particle i, the volume fraction of certain phases are

negative after diffusion, we update λ_i as

$$\lambda_i^{l+1} = \lambda_i^l \min_{k \in S}\left(\frac{\alpha_{k,i}^n}{-\Delta\alpha_{k,i}^l}\right),$$
$$S = \{k | \tilde{\alpha}_{k,i}^l < 0\} \quad (21)$$

where $\Delta\alpha_{k,i}^l$ is the volume fraction change calculated in Eq. (20).

After each iteration, we can estimate the error by summing all negative values in $\alpha_{k,i}^*$. If the relative error is lower than a threshold, we can terminate the iteration process. In all our experiments, we find fixing the maximum iteration number to $l = 2$ is sufficient to keep the relative error below 0.01%.

When the iteration ends, we set $\alpha_{k,i}^{n+1} = \tilde{\alpha}_{k,i}$, where $\tilde{\alpha}_{k,i}$ is the predicted value in the last iteration. We need to normalize $\alpha_{k,i}^{n+1}$ to eliminate the accumulation of the numerical error, which otherwise would lead to the sum of volume fractions to drift away from 1. Finally we update the mass and rest density of particle i according to the volume fractions.

ALGORITHM 2: Phase Transport

```
let max_iter = 2;
forall particles i do
    λ_i = 1;
end
let l = 1;
while l < max_iter do
    forall particles i do
        predict phase transport with Eq. (20);
        update λ_i with Eq. (21)
    end
    l = l + 1;
end
forall particles i do
    update α_{k,i} with α̃_{k,i} in the last iteration;
    normalize volume fractions;
    update ρ_{m,i} and m_i ;
end
```

6. Experiments & Results

The proposed dynamic non-equilibrium mixture model is implemented with the Nvidia CUDA toolkit, and the experiments presented in this paper are all conducted on a PC platform with an Nvidia Geforce GTX1080 graphic card and an 8 core Intel i7-6700K CPU. The performance data are summarised in Table 1.

6.1. Dynamic phase motion

The phase separation due to the density difference is first examined. We use a two-phase mixture, where the densities of the two phases are 500 kg/m^3 and 1000 kg/m^3, respectively. The drag coefficient is set as $k_d = 0.01$, which produces very weak coupling between the phases and therefore rapid separation. In the following experiments, a volume ratio of 1:1 is used to initialize the mixtures.

Pouring wine. In Fig. 4 the mixture is poured into a wine glass,

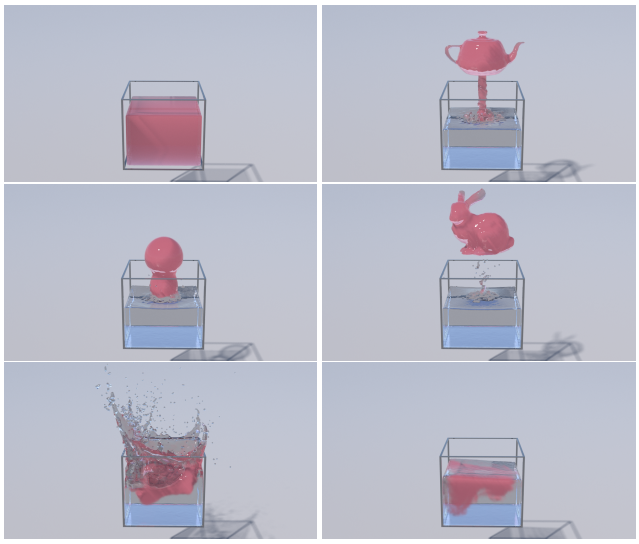

Figure 7: *The external force fields can be applied to individual phases to achieve multiphase fluids control. Here the red phase is forced to separate from the other phase and form different shapes. When being released, the red phase drops back to the water tank.*

and the two phases separate from each other immediately when reaching the bottom of the glass, forming two layers. Bubbling scenes have been well studied with discrete particle method such as [PPK], while here we achieve similar phenomena with the continuum-based approach. The diffusion term is not enabled in this case, and we can observe the bubbles of the lighter phase rising inside the water body, producing interesting visual effects. The lighter phase is rendered as a gaseous foam-like fluid, and the mixture mimics a sparkling wine pouring into a glass.

Rapid mixing. In Fig. 8 we demonstrate that our model also captures the fast motion-induced mixing. In this experiment we use a rather small diffusion coefficient $C_d = 0.0005$, and set the densities of the two phases to 800 kg/m^3 and 1000 kg/m^3 to slightly weaken the phase separation effect. We observe that in this scenario, the relative motion of the two phases and the overall mixture movements significantly increase the mixing process.

Model comparison. A comparison between the previous mixture model [JLDH20] and the proposed dynamic non-equilibrium mixture model is conducted. As shown in Fig. 9, a uniform mixture cube is released into a glass tank. The diffusion coefficient is set as $C_d = 0.001$ for both models, the drag is $k_d = 10$, and the separation coefficient is 0.05 for the previous approach, which is close to the maximum value permitted for stable simulation. The fluids are rendered as particles to better illustrate the phase distribution. It can be observed that the previous method fails to capture the phase separation, and the lighter phase is partially trapped inside the denser phase. This is due to the heavy damping effect of the local equilibrium assumption. On the contrary, the proposed method produces a neat separation effect for the two phases.

It is worth noting that when dealing with high density ratio mixtures, as achieved in [JLDH20], a smaller drag coefficient k_d is re-

Table 1: *Performance data.*

case	N	spacing(m)	Δt(ms)	momentum(ms)	transport(ms)	ms/time step	sec/frame
wine	91.0K	0.005	0.1	5.76	10.76	20.16	8.06
transport	65.6K	0.005	0.1	5.50	10.26	19.02	7.61
→ no fix	-	-	-	5.51	4.59	13.03	5.21
→ old fix	-	-	-	5.47	4.61	13.10	5.24
unmixing	99.2K	0.005	0.1	6.01	14.11	23.27	9.31
→ old unmixing	-	-	-	5.40	11.26	19.71	7.88
fast mixing	111.0K	0.005	0.1	6.71	17.57	27.82	11.13
armadillo	122.2K	0.005	0.1	-	14.78	14.78	5.91
shape control	222.0K	0.006	0.1	13.04	26.18	43.90	17.56
pattern control	117.3K	0.007	0.1	5.95	14.46	23.46	9.38
filtration	147.0K	0.005	0.1	8.03	19.87	33.23	13.29

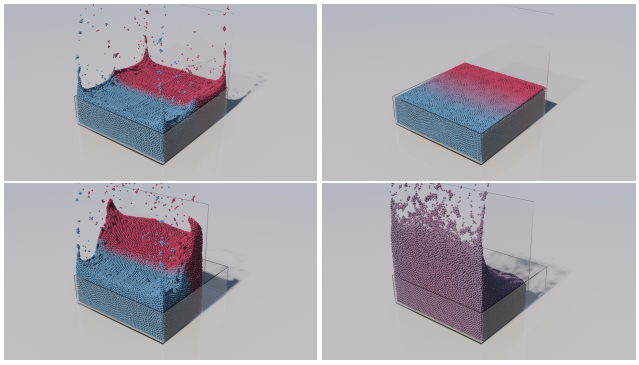

Figure 8: *Rapid mixing. In this experiment, two fluid blocks are released in a tank without being stirred (the first row), and the mixing process is rather slow. In the other case we add a moving glass wall to help the mixing (the second row).*

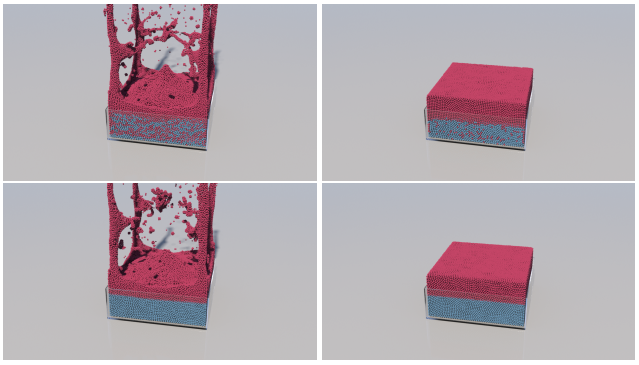

Figure 9: *We compare our method with the previous one in [JLDH20]. The previous method (the first row) does not fully capture the rapid unmixing effect, while our method (the second row) creates a neat separation.*

quired to ensure convergence. We observe that for two phases with density 10 and 1000 kg/m^3, k_d less than 10 is desirable. While for cases with density ratio less than 10, k_d can be arbitrary value.

Verification of transport correction. The effect of our transport correction scheme is examined in this experiment. A simple correction scheme was proposed in [JLDH20], where the flux between particles is directly cancelled in the case of negative volume fractions. When using this simple treatment in the proposed dynamic mixture model, it cannot fully prevent the negative volume fraction values, as the flux is much more vigorous and complicated in our method. We find that even if the flux between particles does not lead to negative values itself, the accumulated flux may still lead to negative values. Our correction scheme handles this case well, as shown in Fig. 10. We compare the three cases: dam break with or without our correction, and the third one with the previous correction scheme. The errors of all three cases are plotted in Fig. 11. Severe mass loss can be observed without correction, our method controls the error around 0.01%, while the previous method only controls the error to around 1%.

6.2. Multiphase fluid control

In previous models only the mixture velocity is tracked and updated, and therefore the external force can only be applied to the whole mixture rather than individual component phases. The previous drift velocity formulation does not consider such control forces either, making it impossible to control the mixing and unmixing effects. In our model, since the phase velocities are tracked and updated by separate momentum equations, we can add control force or manipulate the velocity field of the phases individually for artist controlled effects. In the following experiments, the densities of both phases are set to 1000 kg/m^3 such that the mixture does not separate due to the density difference.

Transport in static geometry. By designing proper velocity fields of the phases, the mixture model can be used to simulate the transport of physical properties in static geometry objects. In Fig. 6, we define the following velocity fields:

$$\mathbf{u}_1 = (1-(\alpha_1-\alpha_2))\mathbf{u},$$
$$\mathbf{u}_2 = -(1-(\alpha_1-\alpha 2))\mathbf{u},$$

where the two phases are denoted with the subscripts $\{1,2\}$. The

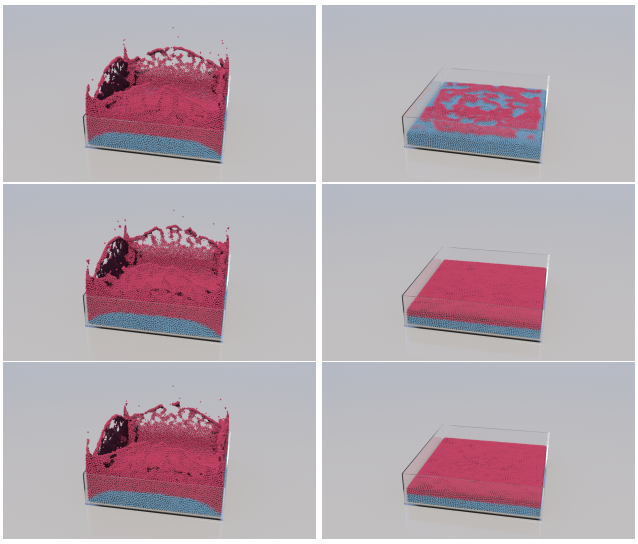

Figure 10: *Transport comparison. The first row is without correction scheme, and the second row is with our correction scheme. The third row is with the correction scheme in [JLDH20]. We observe severe mass loss in the first case, while the other two give visually plausible results.*

mixture velocity is $\mathbf{u}_m = \alpha_1\mathbf{u}_1 + \alpha_2\mathbf{u}_2 = \mathbf{0}$, corresponding to the fixed position of the geometry. As shown in Fig. 6, the transport of phases inside the armadillo is controlled by adjusting the direction of $\mathbf{u}$. An interesting trial that deserves further research is to transport other physical fields such as heat or electricity using this mechanism supported by the dynamic mixture model.

Shape control. In Fig. 7 we demonstrate a shape control technique for multiphase fluids. This is achieved by combining a force field with control particles, where the former acts as a far field attraction, and the latter is the near field effect. Denoting the phase being attracted with the subscript k, the force field is defined as $\mathbf{F}_f(\mathbf{x}_i) = -k_1 m_{k,i} \frac{\mathbf{x}_i - \mathbf{x}_0}{|\mathbf{x}_i - \mathbf{x}_0|^2}$, where $\mathbf{x}_0$ is the attraction center, k_1 is the far field attraction factor, and $m_{k,i}$ is the mass of the fluid phase being attracted. The near field force on fluid particle i is computed by summing up the force from control particle j in its neighborhood, given by

$$\mathbf{F}_n(\mathbf{x}_i) = -m_{k,i} \sum_j k_2(\rho_j + k_3)^{-1} \cos(\frac{\pi}{4}\frac{x_{ij}}{h}) \frac{\mathbf{x}_{ij}}{x_{ij}},$$

where k_2 is the near field attraction factor, $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$, and x_{ij} is the distance. ρ_j is a fake density of the control particle j, estimated as $\rho_j = V_0 \sum_i \alpha_{i,k} W_{ji}$, where the summation includes all fluid particles in its neighborhood. We use this fake density to weaken the attraction of control particles that have been surrounded by fluid particles, as done in [TKPR09]. The constant k_3 is set to 0.1 to prevent from singularity. With these two control forces, we achieve controlled separation and shape deformation for multiphase fluids in Fig. 7, where the red phase is pulled up by the force fields and deforms into different shapes driven by the control particles.

Pattern control. In this experiment, we apply a spatial force field defined by an image patterns to control the color of multiphase fluids, as shown in Fig. 12. The force field is defined as

$$\mathbf{F}_c(\mathbf{x}_i) = -m_{k,i} k_4 \nabla c(\mathbf{x}_i),$$

where $c(\mathbf{x})$ is the single-valued color field, acting as the energy potential, and k_4 is a coefficient adjusting the strength of the control effect. In this case we only control one of the two phases, which is adequate for generating clear image patterns.

Water filtration. In the last experiment we pour the mixture into a funnel with a filter fitted to its bottom neck. When the mixture reaches the filter, we apply the near field attraction and extra drag force to the red phase. As a result, only the green phase can pass through the filter, and the two phases are separated in this filtration process. This experiment shows the capability of our framework to create a wide range of multiphase phenomena by adding proper controls to the individual phases.

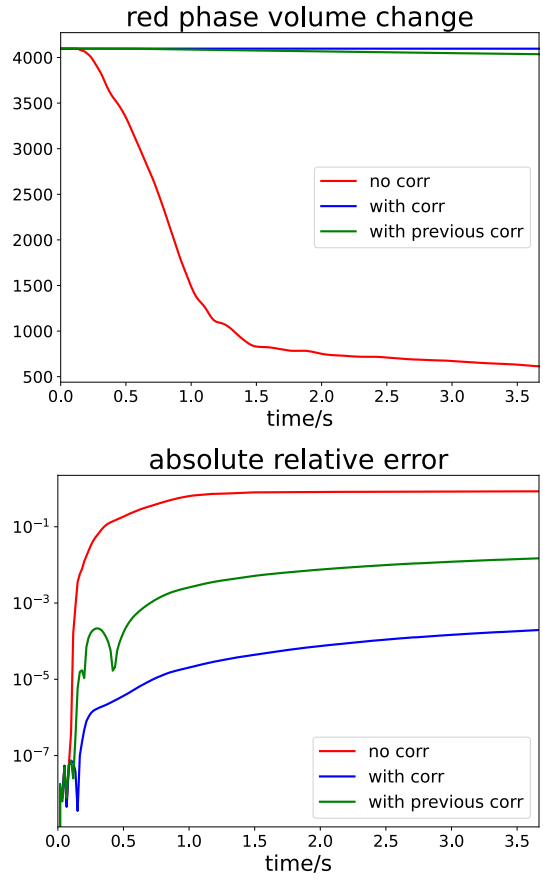

Figure 11: *When we plot the error, we see the volume fraction in the first case dropping down rapidly, while the third case still suffers from a slight mass loss. With our correction scheme the loss is controlled near* 0.01%.

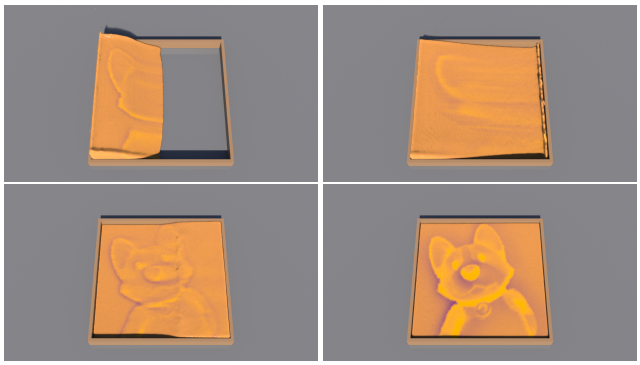

Figure 12: *By applying a spatially varying force field to one of the two phases, we can control the color of the multiphase fluids and generate image patterns.*

7. Conclusion & Discussion

In this work we present a dynamic non-equilibrium mixture model for multiphase fluids with highly dynamic relative motions. By abandoning the local equilibrium assumption and redesigning the computation workflow of the mixture model, we explicitly track and decouple the velocities of all phases. The phases share the same pressure, with which the incompressibility of the mixture is enforced. The combination of the explicit velocity tracking and the shared pressure allow us to capture highly dynamic scenes with negligible overheads. We have also proposed a novel transport correction scheme to handle the transport error at fluid boundaries, ensuring the mass conservation. By decoupling the phase velocities, we can achieve multiphase fluid control by exerting external forces to individual component phases. We have conducted a series of experiments demonstrating the capability of our mixture model, producing interesting visual effects.

The new method is not without limitation. Since we use a single set of particles to discretize the whole mixture, the resolution of the simulation is limited. When the phases in an isolated particle have separating motions, the particle itself could not split, and therefore these sub-particle scale motions are not well represented. Adaptive sampling could alleviate this problem, however the resampling of multiphase fluids is non-trivial and it requires further research.

Like previous methods, we use Fourier diffusion for the phase diffusion, which is more suitable for equilibrium cases. For more dynamic diffusion, a non-Fourier diffusion model is desired [JGE10,NNH14,XSH*20], which requires further research. We assume the component phases are under the same pressure and viscous stress tensor, which does not always hold true. As a result, the current method is not suitable for fluids with different viscosity values, which can be solved by introducing a viscous model that is suitable for multiphase fluids. Besides, we have only considered Newtonian fluids in this work, where the viscosity is handled simply by the artificial viscosity model. Extending our mixture model to handle other materials including solids or non-Newtonian fluids is an interesting topic, and will be pursued in our future work. Also, we would like to explore the integration of our mixture model with phase transition or chemical potential guided phase transport.

Acknowledgements

This work was supported by the National Key R&D Program of China (Project Number 2017YFB1002701) and Shenzhen Fundamental Research Program (No. GXWD20200827110020001).

References

[BB12] BOYD L., BRIDSON R.: Multiflip for energetic two-phase fluid simulation. *ACM Trans. Graph. 31*, 2 (Apr. 2012), 16:1–16:12.

[BK15] BENDER J., KOSCHIER D.: Divergence-free smoothed particle hydrodynamics. In *Symposium on Computer Animation* (2015).

[BT07] BECKER M., TESCHNER M.: Weakly compressible sph for free surface flows. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2007), SCA '07, Eurographics Association, pp. 209–217.

[FQL*20] FANG Y., QU Z., LI M., ZHANG X., ZHU Y., AANJANEYA M., JIANG C.: IQ-MPM: an interface quadrature material point method for non-sticky strongly two-way coupled nonlinear solids and fluids.

[GPH*18] GAO M., PRADHANA A., HAN X., GUO Q., KOT G., SIFAKIS E., JIANG C.: Animating fluid sediment mixture in particle-laden flows. *ACM Trans. Graph. 37*, 4 (July 2018), 149:1–149:11.

[HK04] HONG J.-M., KIM C.-H.: Controlling fluid animation with geometric potential. 147–157.

[HK05] HONG J.-M., KIM C.-H.: Discontinuous fluids. In *ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), SIGGRAPH '05, ACM, pp. 915–920.

[ICS*14] IHMSEN M., CORNELIS J., SOLENTHALER B., HORVATH C., TESCHNER M.: Implicit incompressible sph. *IEEE Transactions on Visualization and Computer Graphics 20*, 3 (Mar. 2014), 426–435.

[JGE10] JOHN B., GU X.-J., EMERSON D. R.: Investigation of heat and mass transfer in a lid-driven cavity under nonequilibrium flow conditions. *Numerical Heat Transfer, Part B: Fundamentals 58*, 5 (2010), 287–303.

[JLDH20] JIANG Y., LI C., DENG S., HU S. M.: A divergence-free mixture model for multiphase fluids. *Computer Graphics Forum 39*, 8 (2020), 69–77.

[KPNS10] KANG N., PARK J., NOH J., SHIN S. Y.: A hybrid approach to multiple fluid simulation using volume fractions. *Computer Graphics Forum 29*, 2 (2010), 685–694.

[LCY*19] LU J., CHEN X., YAN X., LI C., LIN M., HU S.: A rigging-skinning scheme to control fluid simulation. 501–512.

[LLD*20] LI W., LIU D., DESBRUN M., HUANG J., LIU X.: Kinetic-based multiphase flow simulation. *IEEE Transactions on Visualization and Computer Graphics* (2020), 1–1.

[LLP11] LIU S., LIU Q., PENG Q.: Realistic simulation of mixing fluids. *The Visual Computer 27*, 3 (Mar 2011), 241–248.

[LSSF06] LOSASSO F., SHINAR T., SELLE A., FEDKIW R.: Multiple interacting liquids. *ACM Trans. Graph. 25*, 3 (July 2006), 812–819.

[Mad] MADILL D. M. J.: Target particle control of smoke simulation. 8.

[MMS09] MIHALEF V., METAXAS D., SUSSMAN M.: Simulation of two-phase flow with sub-scale droplet and bubble effects. *Comput. Graph. Forum 28* (04 2009), 229–238.

[MTK96] MANNINEN M., TAIVASSALO V., KALLIO S.: On the mixture model for multiphase flow. *VTT Technical Research Centre of Finland* (1996), 288.

[NNH14] NUSKE P., NIASAR V., HELMIG R.: Non-equilibrium in multiphase multicomponent flow in porous media: An evaporation example. *International Journal of Heat and Mass Transfer 74* (07 2014), 128–142.

[NO13] NIELSEN M. B., ØSTERBY O.: A two-continua approach to eulerian simulation of water spray. *ACM Trans. Graph. 32*, 4 (July 2013).

[PPK] PYO S. H., PRAKASH M., KOO B. K.: Bubbling and frothing liquids. 6.

[RLY*14] REN B., LI C., YAN X., LIN M. C., BONET J., HU S.-M.: Multiple-fluid sph simulation using a mixture model. *ACM Trans. Graph. 33*, 5 (Sept. 2014), 171:1–171:11.

[RYL*18] REN B., YANG X.-Y., LIN M. C., THUEREY N., TESCHNER M., LI C.: Visual simulation of multiple fluids in computer graphics: A state-of-the-art report. *Journal of Computer Science and Technology 33*, 3 (May 2018), 431–451.

[SP08] SOLENTHALER B., PAJAROLA R.: Density contrast sph interfaces. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2008), SCA '08, Eurographics Association, pp. 211–218.

[TGK*17] TAMPUBOLON A. P., GAST T., KLÁR G., FU C., TERAN J., JIANG C., MUSETH K.: Multi-species simulation of porous sand and water mixtures. *ACM Trans. Graph. 36*, 4 (July 2017), 105:1–105:11.

[TKPR09] THÜREY N., KEISER R., PAULY M., RÜDE U.: Detail-preserving fluid control. 221–228.

[XSH*20] XUE T., SU H., HAN C., JIANG C., AANJANEYA M.: A novel discretization and numerical solver for non-fourier diffusion. *ACM Trans. Graph. 39*, 6 (Nov. 2020).

[YCL*17] YANG T., CHANG J., LIN M. C., MARTIN R. R., ZHANG J. J., HU S.-M.: A unified particle system framework for multi-phase, multi-material visual simulations. *ACM Trans. Graph. 36*, 6 (Nov. 2017), 224:1–224:13.

[YCR*15] YANG T., CHANG J., REN B., LIN M. C., ZHANG J. J., HU S.-M.: Fast multiple-fluid simulation using helmholtz free energy. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 201:1–201:11.

[YJL*16] YAN X., JIANG Y.-T., LI C.-F., MARTIN R. R., HU S.-M.: Multiphase sph simulation for interactive fluids and solids. *ACM Trans. Graph. 35*, 4 (July 2016), 79:1–79:11.