

# A Divergence-free Mixture Model for Multiphase Fluids

Submission ID: 1009



Figure 1: Pressed by a glass plate from above, an elastoplastic object filled with liquid undergoes plastic deformation, and the liquid inside breaks out due to the increasing pressure.

---

## Abstract

We present a novel divergence free mixture model for multiphase flows and the related fluid-solid coupling. The new mixture model is built upon a volume-weighted mixture velocity so that the divergence free condition is satisfied for miscible and immiscible multiphase fluids. The proposed mixture velocity can be solved efficiently by adapted single phase incompressible solvers, allowing for larger time steps and smaller volume deviations. Besides, the drift velocity formulation is corrected to ensure mass conservation during the simulation. The new approach increases the accuracy of multiphase fluid simulation by several orders. The capability of the new divergence-free mixture model is demonstrated by simulating different multiphase flow phenomena including mixing and unmixing of multiple fluids, fluid-solid coupling involving deformable solids and granular materials.

## CCS Concepts

- *Computing methodologies* → *Physical simulation*;
- 

## 1. Introduction

Multiphase flows are common in our daily lives with interesting visual effects. From latte art to watercolors mixing, multiphase flows demonstrate abundant textures, shapes and colors. The complexity of multiphase flows, however, can go far beyond single-phase systems, especially when materials with significantly different properties mix together. Simulating the discontinuity at the interface between phases and the resulting behaviors has always been a challenging problem.

Simulating a multiphase system requires careful consideration of both the individual phases and their interactions. An intuitive way is to model each phase individually and treat the coupling between phases as extra forces and boundary conditions. Another approach is the so-called mixture model. It models the mixture as a single phase and solves for the averaged velocity, while the relative velocity of each phase is computed with a constitutive model. The latter approach requires much less computation than solving a separate

set of equations for each phase, and is ideal for graphical applications where efficiency matters more [RLY\*14] [YJL\*16].

However, there are still unsolved problems that limit the simulation efficiency and accuracy. First, the mixture velocity proposed by Ren et al. [RLY\*14] is not inherently divergence-free, even if all component phases are incompressible. As a result, the associated multiphase SPH model is incompatible with the incompressible solvers [SP09, ICS\*14, BK15] that are computationally more efficient. Secondly, the formulations of drift velocity and phase exchanging fail to conserve mass, and the associated error accumulates during the simulation.

To solve the aforementioned problems, we present a novel divergence-free mixture model, which enables the use of incompressible fluid solvers for multiphase systems. Our approach supports much larger time steps, and improves the efficiency significantly. We also propose a corrected drift velocity formulation, so that the mass conservation is met during simulation. We show that

the relative error is controlled below 0.001%, which is several orders of magnitude smaller than the previous work.

## 2. Related Work

**Multiphase flows.** Multiphase flows have received increasing attention in computer graphics recently. Immiscible fluids with the presence of discontinuities at the interface have been extensively studied [HK05, LSSF06, BB12, LLD\*20]. However, for miscible fluids, the mixing of phases was often simply treated as a diffusion process in some early research works [KPNS10, LLP11]. Ren et al. [RLY\*14] introduced the drift velocity model into graphics, and successfully simulated the mixing and unmixing phenomena due to the relative motion between individual phases. As stated above, this method lacks a divergence-free formulation and is incompatible with incompressible solvers. Yang et al. [YCR\*15] developed an energy-based approach to guide the phase diffusion, and captured complex phenomena like liquid-liquid extraction. However, this model does not handle the density difference between phases, and also ignores the kinetic effects from the relative motions of component phases. For a detailed overview of multiple-fluid simulation we refer readers to [RYL\*18].

**Incompressible SPH methods.** Becker and Teschner [BT07] proposed a weakly compressible SPH (WCSPH) method to control the fluid compression. Large stiffness parameters are required to enforce small compression rates, which strictly restricts the simulation time step. Solenthaler and Pajarola [SP09] proposed an iterative approach that reduces the density error with a prediction-correction scheme. Ihmsen et al. [ICS\*14] proposed an efficient solver for the Pressure Poisson Equation (PPE), and achieved further performance improvement. Both of these two methods consider only the constant density condition, as this is directly related to the visual effects. In contrast to this, Bender et al. [BK15] proposed to solve for the divergence-free velocity condition in addition to the constant density condition, which allows for even larger time steps. We adapt this approach to handle particles with varied mass and density, and implement our new multiphase solver based on it.

**Lagrangian solids.** The study of deformable solids in graphics has a long history, from which we mainly focus on the Lagrangian methods that are relevant to this work. Müller et al. [MKN\*04] used a point-based method with MLS (moving least squares) for computing deformation gradients. Keiser et al. [KAG\*05] proposed a framework to combine the pressure and viscosity using the SPH method with the MLS-based deformation gradient. Solenthaler et al. [SSP07] modified their work by using the SPH for the computation of all forces. Becker et al. [BIT09] further improved this approach to correctly handle rotations with a corotated SPH formulation. Recently, Peer et al. [PGBT17] developed an implicit formulation for incompressible linear elastic objects, which allowed for larger time steps with better performance. We adopt this solid model in our framework to support larger time steps in simulating multiphase fluid solid coupling.

Simulating elastoplastic deformations is a big challenge, since the drastic deformations sometimes lead to unpredictable shape change. For mesh-based methods, a non-trivial remeshing is needed [BWH07]. Point-based methods handle this problem more easily, since no topological information is required [JWJ\*14, GBB09,

Table 1: Definition of symbols.

Symbol	Meaning
$\alpha_k$	volume fraction of phase $k$
$c_k$	mass fraction of phase $k$
$\rho_k, \rho_m$	density of phase $k$ and the mixture
$\mathbf{u}_k, \mathbf{u}_m$	velocity of phase $k$ and the mixture
$\mathbf{u}_{mk}$	drift velocity of phase $k$
$\tau_k, \tau_m$	viscous stress
$\tau_{Tk}, \tau_{Tm}$	turbulent stress
$\tau_{Dk}, \tau_{Dm}$	interphase stress
$\mathbf{M}_k, \mathbf{M}_m$	interphase momentum exchange
$C_m$	phase separation constant
$D_m$	turbulent diffusion constant

[ZLKW13]. Recently, the Material Point Method (MPM) have gained significant success in simulating elastoplastic materials such as snow [SSC\*13], sands [KGP\*16], and phase changing objects [SSJ\*14].

**Multiphase fluid-solid coupling.** Yan et al. [YJL\*16] extended [RLY\*14] to handle liquids and solids uniformly. The stress of solids is divided into pressure and shear stress, where the pressure is computed with the equation of state, and the shear stress by constitutive models. Their method simulates the coupling between solids and multiple fluids, porous flows and dissolution phenomena. Yang et al. [YCL\*17] introduced phase field to the Helmholtz free energy method to handle the dissolution and phase change of solids. Strong artificial viscosity is required to stabilize the motion of solid particles and suppress the pressure oscillation within the solids. Our mixture model works seamlessly with these coupling approach, while the artificial viscosity can be removed due to the use of incompressible solvers.

MPM has also been applied to simulate multiphase flows including sand-water coupling and particle laden flows [TGK\*17, GPH\*18], achieving appealing visual effects. These methods use separate grids and velocity fields for different phases, and have not considered dissolution effects.

## 3. The Mixture Model

Before describing our method in detail, it is necessary to briefly recap the multiphase mixture model. As stated above, the mixture model treats the multiphase flow as a whole body, and the component phases are described by their relative movement. The fluid motion is divided into two parts: the mixture velocity of the whole body and the drift velocity of each component phase.

The mixture model assumes that the component phases are moving across each other at their terminal velocity, which means there is no relative acceleration between different phases. This holds true when the coupling between phases are strong, such as in certain particle suspension flows. The motion of the mixture can be solved in a similar way to solving a single phase fluid, while the drift velocity is directly computed with a constitutive model.

It is crucial to have an appropriate definition of the mixture

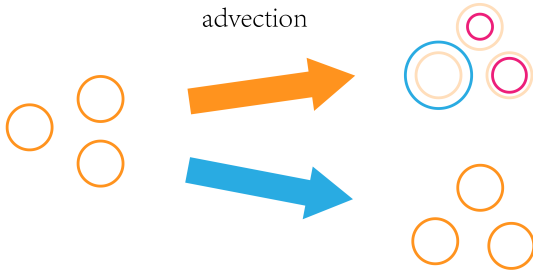


Figure 2: Advection with a non-divergence free velocity field (orange) changes the volume of particles, while advection with a divergence free velocity field (blue) keeps the particle volume constant.

velocity, which is however not very straightforward. In the work of Ren et al. [RLY\*14], the mixture velocity is defined as  $\mathbf{u}'_m = \frac{1}{\rho_m} \sum_k \alpha_k \rho_k \mathbf{u}_k$ , where  $\alpha_k, \rho_k, \mathbf{u}_k$  are the volume fraction, density and velocity of phase  $k$ , respectively, and  $\rho_m$  is the density of the mixture. The divergence of this mixture velocity is related to the density as  $\nabla \cdot \mathbf{u}'_m = -\frac{\partial \rho_m}{\partial t}$ , which is nonzero since the density  $\rho_m$  is highly variable in a multiphase mixture. Therefore, the change of  $\rho_m$  is related to the pressure, and the linear system becomes much more complicated than a common Poisson problem. In the previous work, the fluid pressure is computed with the equation of state in WCSPH instead of solving the linear system.

Besides, there are issues related to the mass conservation. Although the particle-based Lagrangian schemes are known for their natural advantage in maintaining mass conservation, this property only holds true in a multiphase system without phase exchange between particles. As show in Fig. 2, if we advect the multiphase mixture with a non-divergence free velocity field (the orange arrow), the rest volume of the fluid particles will change. As a result, some particles may hold more fluids, while others hold less. The previous work ignores this volume change when computing phase exchange, which is equivalent to scaling all particles' volumes back to the initial volume. Since the volume fractions of particles are not uniform, such scaling violates the mass conservation.

To solve the problems mentioned above, we propose a novel divergence-free mixture model based on a volume-weighted mixture velocity, which enables the use of incompressible solvers for solving the fluid pressure. The divergence free velocity field used for the advection keeps the rest volumes of particles constant (blue arrow in Fig. 2), making the mass conservation easily obeyed without special effort.

#### 4. Divergence-free Mixture Model

In this section we first give our definition of volume-weighted mixture velocity, and then derive the corresponding formulations of the drift velocity and phase diffusion. The symbols used in this paper are listed in Table 1.

#### 4.1. Mixture Velocity

In this work we assume all component phases are incompressible, and the mixing process does not introduce additional volume changes. Therefore the density of phase  $k$  can be treated as a constant. For a multiphase flow, the continuity equation for phase  $k$  is

$$\frac{\partial(\alpha_k \rho_k)}{\partial t} + \nabla \cdot (\alpha_k \rho_k \mathbf{u}_k) = 0. \quad (1)$$

Since the density  $\rho_k$  is constant, the relation  $\frac{\partial \alpha_k}{\partial t} + \nabla \cdot (\alpha_k \mathbf{u}_k) = 0$  holds. Summing over all phases yields  $\nabla \cdot (\sum_k \alpha_k \mathbf{u}_k) = 0$ . Thus we define the mixture velocity as

$$\mathbf{u}_m = \sum_k \alpha_k \mathbf{u}_k, \quad (2)$$

and it satisfies the divergence-free condition  $\nabla \cdot \mathbf{u}_m = 0$ . The proposed volume-based mixture velocity corresponds to the local volume center of fluids, while the conventional mass-based mixture velocity [RLY\*14, YJL\*16] corresponds to the local mass center.

The drift velocity of phase  $k$  is defined as its relative velocity to the mixture, i.e.  $\mathbf{u}_{mk} = \mathbf{u}_k - \mathbf{u}_m$ . Substituting the drift velocity and the relation  $\nabla \cdot \mathbf{u}_m = 0$  into Eq. (1) yields:

$$\frac{\partial \alpha_k}{\partial t} + \mathbf{u}_m \cdot \nabla \alpha_k = -\nabla \cdot (\alpha_k \mathbf{u}_{mk}). \quad (3)$$

With the left hand side denoting the material derivative  $\frac{D\alpha_k}{Dt}$  with respect to the mixture velocity  $\mathbf{u}_m$ , the above equation is used to update the volume fraction of each particle during simulation.

It is trivial to verify  $\sum_k \alpha_k \mathbf{u}_{mk} = 0$  and sequentially  $\sum_k \nabla \cdot (\alpha_k \mathbf{u}_{mk}) = 0$ . This means the volume fractions in each particle always add up to 1, and therefore the mass is well conserved during the advection in Eq. (3). However, the formulation of drift velocity  $\mathbf{u}_{mk}$  derived from a constitutive model normally violates this constraint, so we propose a corrected formulation as explained in § 4.3.

#### 4.2. Momentum Equation

As stated in [MTK96], for a multiphase flow system, the momentum equation of phase  $k$  is

$$\begin{aligned} & \alpha_k \rho_k \left( \frac{\partial \mathbf{u}_k}{\partial t} + \mathbf{u}_k \cdot \nabla \mathbf{u}_k \right) \\ &= -\alpha_k \nabla p_k + \alpha_k \rho_k \mathbf{g} + \nabla \cdot [\alpha_k (\boldsymbol{\tau}_k + \boldsymbol{\tau}_{T_k})] + \mathbf{M}_k, \end{aligned} \quad (4)$$

where  $\boldsymbol{\tau}_k, \boldsymbol{\tau}_{T_k}$  and  $\mathbf{M}_k$  are the viscous stress, the turbulent stress and the inter-phase momentum exchange, respectively. According to the local equilibrium assumption, the phases are under the same pressure, therefore the pressure of phase  $k$  can be replaced by the mixture pressure  $p_m$ . Then by substituting the phase velocity  $\mathbf{u}_k$  with the drift velocity  $\mathbf{u}_{mk}$ , taking  $\rho_k$  out of the derivatives and summing over all phases, the following equation can be obtained:

$$\begin{aligned} \frac{D\mathbf{u}_m}{Dt} &= -\nabla p_m \sum_k \frac{\alpha_k}{\rho_k} + \mathbf{g} \\ &+ \nabla \cdot (\boldsymbol{\tau}_m + \boldsymbol{\tau}_{T_m} + \boldsymbol{\tau}_{Dm}) + \mathbf{M}_m, \end{aligned} \quad (5)$$

where

$$\begin{aligned}\boldsymbol{\tau}_m &= \sum_k \frac{\alpha_k \boldsymbol{\tau}_k}{\rho_k}, \\ \boldsymbol{\tau}_{Tm} &= \sum_k \frac{\alpha_k \boldsymbol{\tau}_{Tk}}{\rho_k}, \\ \boldsymbol{\tau}_{Dm} &= - \sum_k \alpha_k \mathbf{u}_{mk} \mathbf{u}_{mk}, \\ \mathbf{M}_m &= \sum_k \frac{\mathbf{M}_k}{\rho_k}.\end{aligned}\quad (6)$$

The above equation is very similar to the momentum equation of single phase fluid with an additional factor in the pressure term and several inter-phase diffusion terms. The pressure term can be rewritten as:

$$-\nabla p_m \sum_k \frac{\alpha_k}{\rho_k} = -\frac{\gamma}{\rho_m} \nabla p_m,$$

where  $\gamma = \sum_k \frac{\alpha_k^2}{c_k}$ , and  $c_k$  is the mass fraction of phase  $k$ . This factor can be precomputed at each time step, so that the single phase fluid solvers can be easily modified to solve this equation.

### 4.3. Constitutive Model

We adopt the drag force model in [MTK96] to derive our drift velocity formulation. The model assumes that a local equilibrium between all phases is reached at every moment, so that each phase is moving at its terminal velocity relative to the whole mixture. Therefore the drift velocity  $\mathbf{u}_{mk}$  can be directly computed from the drag forces between phase  $k$  and the other phases. We further correct the resulting formulation to ensure  $\sum_k \alpha_k \mathbf{u}_{mk} = 0$ , so that the volume fraction update in Eq. (3) obeys mass conservation.

For the sake of clarity, we provide the detailed derivation in Appendix A, and directly give the drift velocity of phase  $k$  as

$$\mathbf{u}_{mk} = C_m \frac{\rho_k - \rho_m}{\rho_m} \left( \mathbf{g} - \frac{D\mathbf{u}_m}{Dt} \right) + \frac{D_m}{\alpha_k} \nabla \alpha_k, \quad (7)$$

where  $C_m$  and  $D_m$  are constants controlling the separation and diffusion effects. The first term denotes the phase separation due to the inertial difference, and the second term denotes the diffusion due to turbulent fluctuation. In our experiments, we set both  $C_m$  and  $D_m$  at around  $0.001 \sim 0.01$ .

Since the turbulent fluctuation term becomes meaningless when  $\alpha_k = 0$ , we directly set  $\mathbf{u}_{mk} = 0$  if the particle does not contain phase  $k$ . Besides, we rewrite the volume fraction update equation (3) as

$$\frac{D\alpha_k}{Dt} = -\nabla \cdot (\alpha_k \mathbf{u}_{mk0}) + \nabla \cdot \nabla (D_m \alpha_k), \quad (8)$$

where  $\mathbf{u}_{mk0}$  only contains the first term of  $\mathbf{u}_{mk}$ . This formulation now gives meaningful results for all possible volume fraction values.

The interphase momentum exchange  $\mathbf{M}_k$  is given by

$$\mathbf{M}_k = \alpha_k (\rho_k - \rho_m) \left( \mathbf{g} - \frac{D\mathbf{u}_m}{Dt} \right), \quad (9)$$

where terms other than the drag force are negligible and dropped from the equation.

Compared to directly computing the viscous stress  $\boldsymbol{\tau}_m$  and turbulent stress  $\boldsymbol{\tau}_{Tm}$  in Eq. (5), a more practical way is to define a generalized stress tensor  $\boldsymbol{\tau}_{Gm} = \boldsymbol{\tau}_m + \boldsymbol{\tau}_{Tm}$  [MTK96]. The complexity of a turbulent-related viscosity is generally considered an overkill for graphic applications. We therefore replace these terms with the artificial viscosity [BT07], which is adequate to generate visually plausible results.

## 5. Discretization

Following the previous works on multiphase fluids [RLY\*14, YCR\*15], we implement our mixture model with the SPH method and uniformly handle free surface fluids and deformable bodies. To better examine the performance of the new model, we build our solver based on both WCSPH and incompressible SPH methods such as IISPH [ICS\*14] and DFSPH [BK15]. The multiphase properties such as the volume fractions and drift velocities are stored on individual SPH particles. In each time step, the momentum equation of the mixture velocity is solved by the adapted single phase fluid solver, after which the drift velocity and phase exchange between particles are computed analytically.

### 5.1. SPH Formulation

The SPH method interpolates a variable at particle  $i$  from its neighboring particles, that is

$$A_i = \sum_j \frac{m_j}{\rho_j} A_j \nabla W_{ij}, \quad (10)$$

where  $W_{ij}$  is a kernel function, and the cubic spline function is used in this work.

The density estimation for multiphase fluids is slightly different from those for single phase fluids, since the rest density and mass of particles are different. According to [SP08], the density of particle  $i$  is estimated as

$$\rho_i = m_i \sum_j W_{ij}. \quad (11)$$

Using the above density estimation, the problem is transformed into enforcing a constant particle volume.

### 5.2. Momentum Equation using SPH

The terms in the momentum equation (5) are discretized as

$$\begin{aligned}\left(-\frac{\gamma}{\rho_m} \nabla p_m\right)_i &= -\gamma_i \sum_j m_j \left( \frac{p_j}{\rho_j^2} + \frac{p_i}{\rho_i^2} \right) \nabla W_{ij}, \\ (\nabla \cdot \boldsymbol{\tau}_{Dm})_i &= -\sum_j \frac{m_j}{\rho_j} (\boldsymbol{\tau}_{Dm,i} + \boldsymbol{\tau}_{Dm,j}) \nabla W_{ij}.\end{aligned}\quad (12)$$

The viscosity term involving  $\boldsymbol{\tau}_m$  and  $\boldsymbol{\tau}_{Tm}$  is replaced by the artificial viscosity given as

$$\begin{aligned}\frac{F_i^v}{m_i} &= -\sum_j m_j \Pi_{ij} \nabla W_{ij}, \\ \Pi_{ij} &= -\frac{2\alpha_h c_s}{\rho_i + \rho_j} \frac{\min(0, \mathbf{u}_{m,ij} \cdot \mathbf{x}_{ij})}{d^2 + \epsilon h^2}\end{aligned}\quad (13)$$



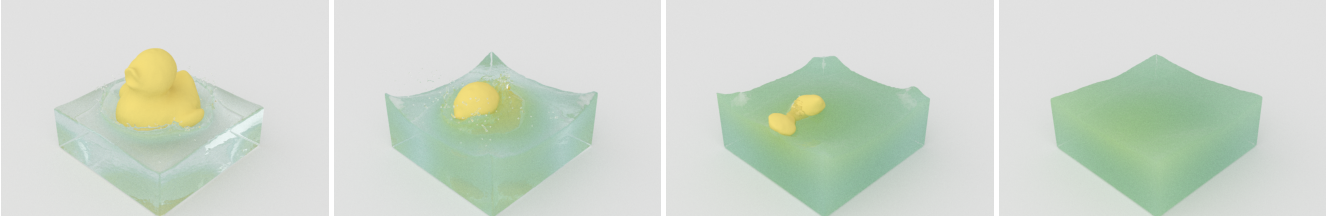


Figure 3: An elastic duck dissolving in water. Since the duck's density is half of the water density, it floats on the water surface.

---

**Algorithm 1:** Simulation Loop (IISPH)
 

---

```

forall particles  $i$  do
  estimate density  $\rho_i$  Eq. (11);
  predict velocity  $\mathbf{u}_{m,i}^{adv}$  (§ 5.2);
  compute  $\mathbf{d}_{ii}$  Eq. (14);
end
forall particles  $i$  do
  predict density  $\rho_i^{adv}$  (14);
  compute  $\mathbf{a}_{ii}$  as in IISPH;
end
 $l = 0$ ;
while  $err_{avg}^l > \eta \vee l < 2$  do
  IISPH iteration;
end
forall particles  $i$  do
   $\mathbf{u}_{m,i}(t + \Delta t) = \mathbf{u}_{m,i}^{adv} + \gamma_i \Delta t F_i^p(t) / m_i$ ;
   $\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \Delta t \mathbf{u}_{m,i}(t + \Delta t)$ ;
end
phase diffusion (Algorithm 2);
  
```

---

where  $\alpha$  is a viscosity constant,  $c_s$  the sound speed,  $h$  the smoothing distance,  $\epsilon$  a small constant (normally set to 0.01) for preventing singularity. The interphase term  $\mathbf{M}_{m,i}$  does not require SPH discretization, and it is computed at the end of each time step and stored on particles.

### 5.3. Incompressible SPH Solver

Solving the pressure with an incompressible SPH solver is straight forward, and we implemented our method on both IISPH and DFSPH. The simulation workflow is explained below using IISPH as an example, and is outlined in Alg. 1.

At the beginning of each time step, we first estimate the density and precompute the pressure factor  $\gamma$  for each fluid particle. Then the velocity and density change are predicted using the momentum equation without the pressure term, after which the pressure is solved to satisfy the divergence free condition and the particle position and velocity are integrated. Next, the drift velocity  $\mathbf{u}_{mk}$ , the interphase momentum exchange  $\tau_{Dm}$  and  $\mathbf{M}_m$  are explicitly computed. Finally, we compute the phase exchange between particles, and update the mass and rest density of each particle. This part is also outlined in Alg. 2.

With the factor  $\gamma$ , the formulation of IISPH should be changed as

$$\begin{aligned}
 \rho_i^{adv} &= \rho_i(t) + \Delta t \sum_j m_j \mathbf{u}_{m,i,j}^{adv} \cdot \nabla W_{ij}, \\
 \mathbf{d}_{ii} &= -\gamma_i \Delta t^2 \sum_j \frac{m_j}{\rho_i^2} \nabla W_{ij}, \\
 \mathbf{d}_{ij} &= -\gamma_i \Delta t^2 \frac{m_j}{\rho_j^2} \nabla W_{ij}.
 \end{aligned} \tag{14}$$

The rest of the derivation stays unchanged, and the resulting formulation is very similar to the original one.

The approach in [SP08] can also be used to deal with the density contrast of multiphase fluids. But when solving the pressure in the Jacobi fashion, as in IISPH and DFSPH, the linear system becomes ill-conditioned if the density contrast goes too high, and the calculation may fail to converge. Fortunately, the density ratio between common fluids and solids does not lead to this situation, while problems involving air-water coupling should be handled with a different approach rather than a multiphase mixture model.

### 5.4. Phase Exchange using SPH

The phase exchange is discretized as symmetric pairwise fluxes to ensure mass conservation. Following the SPH discretization in [IOS\*14], the two terms in the phase exchange equation Eq. (8) are discretized as:

$$\begin{aligned}
 \nabla \cdot (\alpha_k \mathbf{u}_{mk0})_i &= \sum_j V_{ij} (\alpha_{k,i} \mathbf{u}_{mk0,i} + \alpha_{k,j} \mathbf{u}_{mk0,j}) \cdot \nabla W_{ij} \\
 \nabla \cdot \nabla (D_m \alpha_k)_i &= 2D_m \sum_j V_{ij} (\alpha_{k,i} - \alpha_{k,j}) \frac{\mathbf{x}_{ij} \cdot \nabla W_{ij}}{\mathbf{x}_{ij} \cdot \mathbf{x}_{ij} + \epsilon}
 \end{aligned} \tag{15}$$

where  $V_{ij} = \frac{1}{2} \left( \frac{m_i}{\rho_i} + \frac{m_j}{\rho_j} \right)$ ,  $\epsilon = 0.01h^2$ . In the above formulation, the phase update of particle  $i$  is discretized into pairwise fluxes between particle  $i$  and its neighboring particles.

However, negative volume fractions may appear during the simulation, and will break the mass conservation if they are simply clamped to the interval  $[0, 1]$ . The negative volume fractions are due to the flux of the separation term, which sometimes forces a component phase to flow out of particles with very low volume fractions. Therefore, in each time step, we test whether or not the separation flux between a pair of particles will lead to negative volume fractions of any component phases. If this is true, the separation term is disabled between this pair of particles in this time step so that only the diffusion term functions. We test and disable the flux for each pair of particles independently, and sometimes the total fluxes of

**Algorithm 2:** Phase Diffusion

---

```

forall particles  $i$  do
  get particle acceleration
   $\mathbf{a}_i(t + \Delta t) = (\mathbf{u}_{m,i}(t + \Delta t) - \mathbf{u}_{m,i}(t)) / \Delta t$ ;
  compute drift velocities Eq. (7);
  compute  $\tau_{Dm}$  Eq. (6);
  compute  $M_m$  Eq. (9) and Eq. (6);
  compute phase diffusion Eq. (15);
end

```

---

Table 2: Runtime performance of our examples

Example	N	$\Delta t$ (ms)	iteration	s/frame
dambreak	203k	1.0	5.31	6.7
~ WCSPH	203k	0.2	-	12.1
pour water	61k~161k	0.5	2.43	6.2
~ WCSPH	61k~161k	0.2	-	10.1
cube	178k	0.5	3.10/1.99	6.0
duck	37.3k	0.5	3.2/2.0	16.1
sand	114k~514k	0.5	2.0	20.5
gear	993k	0.5	3.15/2.3	60.7

one particle’s several neighbors can lead to negative volume fractions. This value is clamped to zero, and it stops any out-flowing flux in the following time steps. In our experiments this small violation in one time step is found negligible.

After the phase update, volume fraction values may still deviate from the possible region due to the time discretization and floating error, therefore we clamp the values to the interval  $[0, 1]$  and re-normalize the volume fractions of the corresponding particles. With the proposed phase update approach, the relative error is controlled below 0.01%, as is shown in Fig. 4.

## 6. Results

The proposed incompressible multiphase SPH algorithm is implemented with CUDA, and all experiments presented in this section were simulated on an NVIDIA Geforce GTX1080 graphic card and an 8 core Intel i7-6700K CPU. The performance data are given in Table 2, and in this paper we measure the performance at a frame rate of 30 Hz. The average iterations of IISPH or DFSPH are also listed in the table, where cases solved by DFSPH has two iteration numbers corresponding to the density solver and the divergence solver respectively.

### 6.1. Phase Exchange

*Dam break.* A dam break case of two fluid pillars is shown in Fig. 4 to demonstrate the efficiency and accuracy of our approach. The particle spacing is 0.01 m, a fixed time step of 1 ms is used for the IISPH-based solver, and the threshold of average error of particle volume is set to 0.1%. While the previous approach [RLY\*14] is viable to simulate this scenario, a much smaller time step around 0.2 ms is required. As shown in Table 2, our approach implemented with IISPH is about 2x faster.

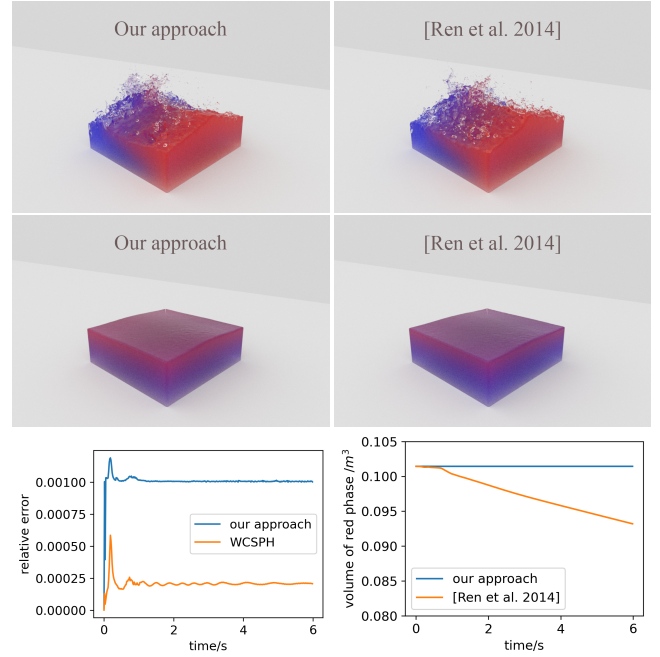


Figure 4: A dam break case using our approach and [RLY\*14]. The overall motions are similar, while the rest volumes of component phases are constantly changing for [RLY\*14].

The overall motion of fluids generated by the two approaches are similar, while the volumes of component phases are not conserved when using the previous method. In this case the densities of the red and blue phase are 1000 and 1300 kg/m<sup>3</sup> respectively. As shown in Fig. 4, red phase decreases with time in the result by [RLY\*14] and causes artificial colour changes to the mixture. The mass loss accumulates with time, and reaches above 10% at the end of the simulation, while our approach controls the mass loss below 0.01%, improving the accuracy by several orders.

The average errors of particle volumes of the two approaches are also compared. The obtained average error of WCSPH is smaller than IISPH, since we have chosen a large stiffness parameter to enforce incompressibility and eliminate pressure oscillation. However, even with a much smaller time step, we still observe larger oscillations in the figure. These comparisons confirm that using an incompressible solver does improve both the efficiency and stability of the simulation.

*Pouring water.* We have conducted another case comparing the two methods, where red phase liquid is poured into a tank half-filled with blue phase liquid. The densities of the red and blue phase are 1000 and 500 kg/m<sup>3</sup> respectively. As is shown in Fig.5, when solved with the previous method, the blue phase suffers from mass loss, and the color is incorrectly redder. As the particle number increases, the final performance is given in Table. 2.

### 6.2. Two-way Coupling

Since the incompressible SPH solver naturally handles particle collisions, the two-way coupling can be achieved more easily com-

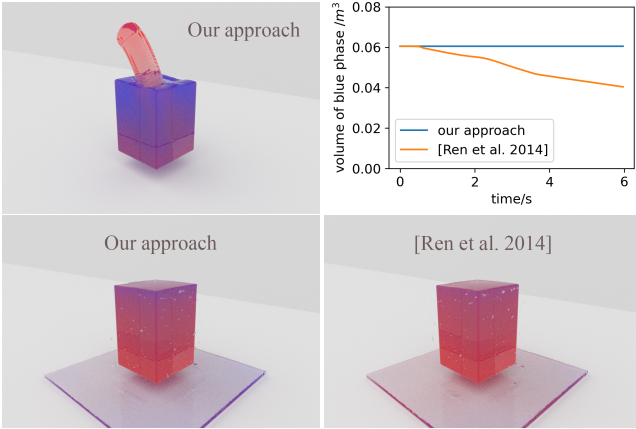


Figure 5: Another comparison of our approach with [RLY\*14]. Red phase is poured into a tank containing some blue phase liquid. The blue phase decreases when solved by the previous method, making the mixture incorrectly redder.

pared to the method in [YJL\*16]. To simulate hyperelastic solids we adopt the elastic solid model in [PGBT17] and handle the elastoplastic deformation with the model in [BWH07]. The particle spacing is 0.005 m in the following cases.

**Pressing cube.** Fig. 1 demonstrates the stability of our fluid-solid coupling model. The cube is initially filled with fluid. Pressed by a glass plate, it undergoes plastic deformation and the internal liquid eventually breaks out from the corners of the cube. When the glass board is lifted up, the cube can no longer return to its initial shape.

**Duck dissolution.** The dissolution is treated as a special kind of phase diffusion between solid and liquid particles, since the Noyes-Whitney equation has the same form of the diffusion equation, as applied by Yan et al. [YJL\*16]. In Fig. 3, an elastic duck is dropped onto a water surface. We set the density of the duck to half of the water, so that it can float on the water surface. It slowly dissolves into the water. The water is originally light blue, and the yellow duck material gives it a green tint.

**Sand dissolution.** Fig. 6 shows a sand dragon dissolves in a water tank. While water is poured at the left side of the tank, the sand dragon collapses on the floor and sand particles are flushed toward the right side by the fast flowing water. The densities of water and sand are set as  $1000 \text{ kg/m}^3$  and  $1300 \text{ kg/m}^3$ , respectively. We set the granular to be dissolvable, and a particle suspension flow is formed as the water interacts with the sand. The particle number increases when water is poured into the tank, and the final performance is 20.5 seconds per frame as listed in Table 2.

### 6.3. Fast Moving Boundaries

**Gear motions.** Using the incompressible methods, our new model can stably handle scenarios involving fast moving objects and complex boundaries. Two scenarios involving the coupling between rapidly rotating gears and liquid mixtures are set up, as shown in Fig. 7. The tank is filled with red, green, and blue liquids of densities  $500 \text{ kg/m}^3$ ,  $1000 \text{ kg/m}^3$  and  $1500 \text{ kg/m}^3$  respectively. The

particle spacing is 0.01 m, the two gears are rotating at 6 rad/s, and a fixed time step of 0.5 ms is used in the simulation. The rotation of the two gears generates strong mixing effects. We choose relatively large parameter  $C_m$  and  $D_m$  so that while mixing is obvious in the stirred region, the mixture separates when flowing into the un-stirred regions. After the gears stop rotating, the mixture settles and gradually separates back into layers.

## 7. Conclusion and Discussion

In this paper we propose a divergence free mixture model for multiphase fluids, which enables the use of incompressible fluid solver to handle multiphase phenomena, and therefore significantly improves the efficiency in simulating large and highly dynamic scenarios.

In our model, the mixture velocity is weighted by phase volume and corresponds to the local volume center of the mixture. The resulting momentum equation involves an additional volume fraction related factor, which can be easily solved by any single phase incompressible fluid solver. This definition leads to a formulation of phase exchange that obeys mass conservation, greatly improving the accuracy of multiphase fluid simulation. A series of experiments are presented to demonstrate the capability of our approach, including multiple fluid mixing, fluid-solid two way coupling, and dissolution of elastic body and granular materials.

Since our mixture model is based on the assumption that phases are moving at their terminal velocity across each other, the usage of our mixture model is restricted to cases where strong coupling exists between phases, such as liquid-particle mixtures. While more vibrant mixtures such as gas-particle mixtures involves much weaker coupling, and therefore is not suitable for this approach. Extending this approach to handle a wider range of phenomena would be an interesting topic for future research.

Besides, the current incompressible SPH method does not handle high density ratio well. The linear system becomes ill-conditioned under high density contrast, and the Jacobi fashioned approach could fail to converge. We have also shown this instability in our video. The conjugate gradient method is not a viable solution, since the varied particle mass makes the linear system nonsymmetric. Normally we do not face such problems, since the mixture model handles only strongly coupled multiphase systems. Using other Krylov space solving method such as GMRES may help to solve this problem, but requires further research.

## Acknowledgements

This work was supported by the National Key R&D Program of China (Project Number 2017YFB1002701).

## References

- [BB12] BOYD L., BRIDSON R.: Multiflip for energetic two-phase fluid simulation. *ACM Trans. Graph.* 31, 2 (Apr. 2012), 16:1–16:12. 2
- [BIT09] BECKER M., IHMSEN M., TESCHNER M.: Corotated sph for deformable solids. In *Proceedings of the Fifth Eurographics Conference on Natural Phenomena* (Aire-la-Ville, Switzerland, Switzerland, 2009), NPH'09, Eurographics Association, pp. 27–34. 2





Figure 6: A sand dragon collapses in a water tank while water is being poured from the left side. As a result, sand particles are flushed to the right side and dissolve gradually into water.

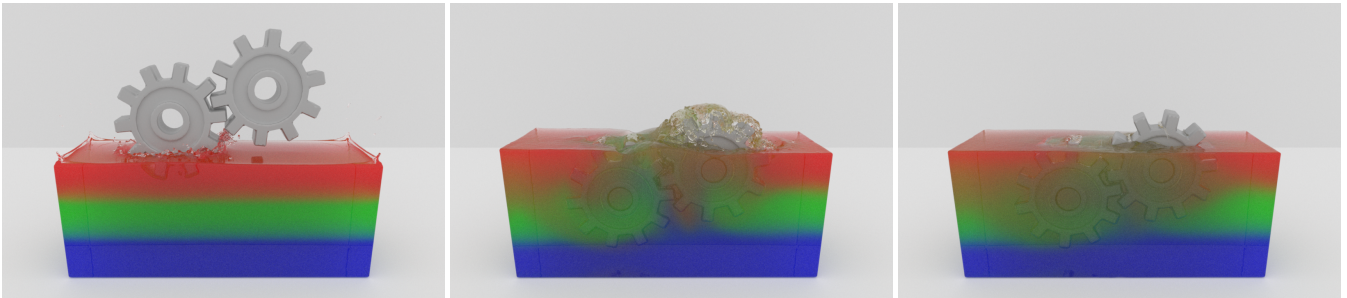


Figure 7: Two rotating gears are immersed into a tank pre-filled with three liquids (red, green and blue), rapidly stirring the liquids and creating strong mixing effects.

- [BK15] BENDER J., KOSCHIER D.: Divergence-free smoothed particle hydrodynamics. In *Symposium on Computer Animation* (2015). 1, 2, 4
- [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. 2, 4
- [BWH07] BARGTEIL A. W., WOJTAN C., HODGINS J. K., TURK G.: A finite element method for animating large viscoplastic flow. *ACM Trans. Graph.* 26, 3 (July 2007). 2, 7
- [GBO9] GERSZEWSKI D., BHATTACHARYA H., BARGTEIL A. W.: A point-based method for animating elastoplastic solids. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2009), SCA '09, ACM, pp. 133–138. 2
- [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. 2
- [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. 2
- [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. 1, 2, 4
- [IOS\*14] IHMSEN M., ORTHMANN J., SOLENTHALER B., KOLB A., TESCHNER M.: SPH Fluids in Computer Graphics. In *Eurographics 2014 - State of the Art Reports* (2014), Lefebvre S., Spagnuolo M., (Eds.), The Eurographics Association. 5
- [JWJ\*14] JONES B., WARD S., JALLEPALLI A., PERENIA J., BARGTEIL A. W.: Deformation embedding for point-based elastoplastic simulation. *ACM Trans. Graph.* 33, 2 (Apr. 2014), 21:1–21:9. 2
- [KAG\*05] KEISER R., ADAMS B., GASSER D., BAZZI P., DUTRE P., GROSS M.: A unified lagrangian approach to solid-fluid animation. In *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics, 2005*. (June 2005), pp. 125–148. 2
- [KGP\*16] KLÁR G., GAST T., PRADHANA A., FU C., SCHROEDER C., JIANG C., TERAN J.: Drucker-prager elastoplasticity for sand animation. *ACM Trans. Graph.* 35, 4 (July 2016), 103:1–103:12. 2
- [KPN10] 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. 2
- [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. 2
- [LLP11] LIU S., LIU Q., PENG Q.: Realistic simulation of mixing fluids. *The Visual Computer* 27, 3 (Mar 2011), 241–248. 2
- [LSSF06] LOSASSO F., SHINAR T., SELLE A., FEDKIW R.: Multiple interacting liquids. *ACM Trans. Graph.* 25, 3 (July 2006), 812–819. 2
- [MKN\*04] MÜLLER M., KEISER R., NEALEN A., PAULY M., GROSS M., ALEXA M.: Point based animation of elastic, plastic and melting objects. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar Germany, Germany, 2004), SCA '04, Eurographics Association, pp. 141–151. 2
- [MTK96] MANNINEN M., TAIVASSALO V., KALLIO S.: On the mixture model for multiphase flow. *VTT Technical Research Centre of Finland* (1996), 288. 3, 4, 9
- [PGBT17] PEER A., GISSLER C., BAND S., TESCHNER M.: An implicit sph formulation for incompressible linearly elastic solids: Implicit elastic sph solids. *Computer Graphics Forum* 37 (12 2017). 2, 7
- [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. 1, 2, 3, 4, 6, 7
- [RYL\*18] REN B., YANG X.-Y., LIN M. C., THUREY 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. 2

- [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. 4, 5
- [SP09] SOLENTHALER B., PAJAROLA R.: Predictive-corrective incompressible sph. *ACM Trans. Graph.* 28, 3 (July 2009), 40:1–40:6. 1, 2
- [SSC\*13] STOMAKHIN A., SCHROEDER C., CHAI L., TERAN J., SELLE A.: A material point method for snow simulation. *ACM Trans. Graph.* 32, 4 (July 2013), 102:1–102:10. 2
- [SSJ\*14] STOMAKHIN A., SCHROEDER C., JIANG C., CHAI L., TERAN J., SELLE A.: Augmented mpm for phase-change and varied materials. *ACM Trans. Graph.* 33, 4 (July 2014), 138:1–138:11. 2
- [SSP07] SOLENTHALER B., SCHLÄFLI J., PAJAROLA R.: A unified particle model for fluid-solid interactions. *Computer Animation and Virtual Worlds* 18, 1 (2007), 69–82. 2
- [TJK\*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. 2
- [YJL\*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. 2
- [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. 2, 4
- [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. 1, 2, 3, 7
- [ZLKW13] ZHOU Y., LUN Z., KALOGERAKIS E., WANG R.: Implicit integration for particle-based simulation of elasto-plastic solids. *Computer Graphics Forum* 32, 7 (2013), 215–223. 2

## Appendix A: Drift Velocity

To derive the drift velocity, we start from the momentum equation of phase  $k$ , the left hand side of which can be approximated as

$$\alpha_k \rho_k \left( \frac{\partial}{\partial t} \mathbf{u}_k + (\mathbf{u}_k \cdot \nabla) \mathbf{u}_k \right) \approx \alpha_k \rho_k \frac{D\mathbf{u}_k}{Dt}.$$

This approximation is due to  $\mathbf{u}_k \approx \mathbf{u}_m$ . According to [MTK96], the viscous stress, diffusion stress and the inter-phase term are small compared to the pressure and gravity, and can be dropped in the following derivation. The turbulent stress is kept for the turbulent diffusion, which is an essential part of the multiphase phenomena. Therefore, Eqs. (4) and (5) can be rewritten as

$$\begin{aligned} \alpha_k \rho_k \frac{D\mathbf{u}_k}{Dt} &= -\alpha_k \nabla p_k + \alpha_k \rho_k \mathbf{g} + \nabla \cdot (\alpha_k \boldsymbol{\tau}_{T_k}) + \mathbf{M}_k \\ \frac{D\mathbf{u}_m}{Dt} &= -\sum_k \frac{\alpha_k}{\rho_k} \nabla p_m + \mathbf{g} + \nabla \cdot \boldsymbol{\tau}_{T_m} \end{aligned} \quad (16)$$

The local equilibrium assumption leads to  $\frac{D\mathbf{u}_{mk}}{Dt} = 0$ , and therefore  $\frac{D\mathbf{u}_k}{Dt} = \frac{D\mathbf{u}_m}{Dt}$ . The second equation comes from the definition of  $\mathbf{u}_{mk}$ . We assume  $\nabla p_k \approx \nabla p_m$  as in § 4.2, and define  $\rho' = (\sum_k \frac{\alpha_k}{\rho_k})^{-1}$ . Substituting these into Eq. (16), the interphase momentum exchange can be expressed as:

$$\mathbf{M}_k = \alpha_k (\rho_k - \rho') \left( \frac{D\mathbf{u}_m}{Dt} - \mathbf{g} \right) + \alpha_k \rho' \nabla \cdot \boldsymbol{\tau}_{T_m} - \nabla \cdot (\alpha_k \boldsymbol{\tau}_{T_k}). \quad (17)$$

The drag force model in [MTK96] is given by

$$\mathbf{M}_k = -\beta \mathbf{u}_{ck} + \mathbf{M}'_k, \quad (18)$$

where  $\mathbf{M}'_k$  is a term caused by velocity fluctuations, and  $\beta$  is the drag coefficient determined by many attributes such as fluid densities and particle radius.  $\mathbf{u}_{ck}$  is the relative velocity of phase  $k$ , defined as  $\mathbf{u}_{ck} = \mathbf{u}_k - \mathbf{u}_c$ , where  $\mathbf{u}_c$  is the mixture velocity of the other phases. Combining Eqs. (17) and (18) closes the equation for  $\mathbf{u}_{ck}$ , of which an approximate solution is given as:

$$\mathbf{u}_{ck} = \mathbf{u}_{ck0} + \frac{D_{ck}}{\alpha_k} \nabla \alpha_k \quad (19)$$

where  $\mathbf{u}_{ck0}$  is a solution for the equation without fluctuation terms, that is,

$$\mathbf{u}_{ck0} = \frac{\rho_k - \rho'}{\beta} \left( \mathbf{g} - \frac{D\mathbf{u}_m}{Dt} \right). \quad (20)$$

$D_{ck}$  is the fluctuation coefficient related to the turbulent energy density. The drift velocity  $\mathbf{u}_{mk}$  can then be computed with the relative velocity  $\mathbf{u}_{ck}$  as  $\mathbf{u}_{mk} = (1 - \alpha_k) \mathbf{u}_{ck}$ . Therefore, the drift velocity of phase  $k$  is

$$\mathbf{u}_{mk} = (1 - \alpha_k) \left( \frac{\rho_k - \rho'}{\beta} \left( \mathbf{g} - \frac{D\mathbf{u}_m}{Dt} \right) + \frac{D_{ck}}{\alpha_k} \nabla \alpha_k \right). \quad (21)$$

However, this formulation violates the constraint  $\sum_k \alpha_k \mathbf{u}_{mk} = 0$ . This violation is caused by the approximation adopted in the above derivation. We therefore propose a corrected formulation based on this formulation:

$$\mathbf{u}_{mk} = C_m \frac{\rho_k - \rho_m}{\rho_m} \left( \mathbf{g} - \frac{D\mathbf{u}_m}{Dt} \right) + \frac{D_m}{\alpha_k} \nabla \alpha_k, \quad (22)$$

where the coefficients  $C_m$  and  $D_m$  are constants that have the same values for all phases.