Multiple-Fluid SPH Simulation Using a Mixture Model

BO REN
Tsinghua University
CHENFENG LI
Swansea University
XIAO YAN
Tsinghua University
MING C. LIN
University of North Carolina at Chapel Hill
JAVIER BONET
Swansea University
and
SHI-MIN HU
Tsinghua University

This article presents a versatile and robust SPH simulation approach for multiple-fluid flows. The spatial distribution of different phases or components is modeled using the volume fraction representation, the dynamics of multiple-fluid flows is captured by using an improved mixture model, and a stable and accurate SPH formulation is rigorously derived to resolve the complex transport and transformation processes encountered in multiple-fluid flows. The new approach can capture a wide range of real-world multiple-fluid phenomena, including mixing/unmixing of miscible and immiscible fluids, diffusion effect and chemical reaction, etc. Moreover, the new multiple-fluid SPH scheme can be readily integrated into existing state-of-the-art SPH simulators, and the multiple-fluid simulation is easy to set up. Various examples are presented to demonstrate the effectiveness of our approach.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Animation*; I.6.8 [**Simulation and Modeling**]: Types of Simulation—*Animation*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Physically based animation, fluid simulation, multiphase and multicomponent flow, miscible and immiscible fluids, smoothed particle hydrodynamics, volume fraction model, mixture model

ACM Reference Format:

Bo Ren, Chenfeng Li, Xiao Yan, Ming C. Lin, Javier Bonet, and S.-M. Hu. 2014. Multiple-fluid SPH simulation using a mixture model. ACM Trans. Graph. 33, 5, Article 171 (August 2014), 11 pages.
DOI: http://dx.doi.org/10.1145/2645703

1. INTRODUCTION

Over the past decade, multiple-fluid simulation has received considerable attention in the graphics community. Much of these works focused on inter-facial flows (e.g., [Losasso et al. 2006; Hong et al. 2008; Boyd and Bridson 2012; Misztal et al. 2012]), a special class of multiple-fluid systems where the fluids are immiscible with each other and clear interfaces exist between different phases or components. Another category of multiple-fluid flows involves miscible or dispersed fluid mixtures where interfaces can be difficult to track continuously, or even do not exist. Like inter-facial flows, modeling miscible or dispersed fluids is also important for the visual plausibility of graphic applications, such as water-spray dynamics [Nielsen and Osterby 2013] used in modeling of waterfalls, water jets, and stormy seas. Miscible flows are more flexible to achieve polytropic appearances featuring continuously varying color details, as in colloidal dispersions or dissolving mixtures, which is largely different in visual effect from the surface-rich inter-facial flow. However, very little work has involved this aspect. Meanwhile,

This work is supported by the National Basic Research Project of China (project no. 2011CB302205), the Natural Science Foundation of China (project no. 61120106007), and the National High Technology Research and Development Program of China (project no. 2012AA011503). The authors would also like to acknowledge the financial support provided by the National Science Foundation (NFS IIS-1320644) and UNC Carolina Development Foundation, and Ser Cymru National Research Network in Advanced Engineering and Materials.

Authors' addresses: B. Ren, Beijing Higher Institution Engineering Research Center of Intelligent Processing of Visual Media and Content Security at Tsinghua University, Haidian, Beijing, China; C. Li, College of Engineering, Swansea University, Singleton Park, Swansea SA2 8PP, UK; X. Yan, Beijing Higher Institution Engineering Research Center of Intelligent Processing of Visual Media and Content Security at Tsinghua University, Haidian, Beijing, China; M. C. Lin, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC; J. Bonet, College of Engineering, Swansea University, Singleton Park, Swansea SA2 8PP, UK; S.-M. Hu (corresponding author), Beijing Higher Institution Engineering Research Center of Intelligent Processing of Visual Media and Content Security at Tsinghua University, Haidian, Beijing, China; email: shimin@tsinghua.edu.cn.

DOI: http://dx.doi.org/10.1145/2645703

the majority of the research on multiple-fluid simulation use Eulerian methods, and simulating multiple-fluid flows with Lagrangian methods remains a challenging task. We propose a robust Smoothed Particle Hydrodynamics (SPH) approach to simulate multiple-fluid flows. A special focus is placed on multiple-fluid flows that do not necessarily have (or are difficult to track) clear and persistent interfaces.

Inter-facial flows can be solved in a similar way as the single-phase flow, with the main modeling challenge arising from continuous tracking of fluids' interfaces. However, when different phases or components can mix with each other, that is, are *miscible* with each other, whether in a continuous manner (such as water solution) or dispersed manner (such as slurry flows), the motion and distribution of different phases or components cannot be captured by the single-phase fluid simulation. Different phases or components in a multiple-fluid flow have different fluid properties (e.g., density, viscosity, etc.) and as a result move at different velocities, causing relative motions between phases or components. The various visually interesting mixing/unmixing processes are the combined result of the advection driven by the fluid's bulk motion, the turbulence caused by fluids' instability, the diffusion driven by concentration difference, and, most dominantly, the relative motion determined by the interactions between phases or components. The major challenge in modeling multiple-fluid flow is to resolve the aforesaid dynamic interactions between phases or components [Yeoh and Tu 2009; Crowe et al. 2011].

We introduce a *mixture model* for simulating multiple-fluid flows, in which the distribution of different phases or components is represented by their volume fractions and does not rely on continuous tracking of fluids' interfaces. Also, we compute analytically the drift velocities defined as the phase (or component) velocities relative to the mixture average. As a result, the governing equations of multiple-fluid flow are retained similar to the single-phase flow, even for mixtures consisting of an arbitrary number of phases or components. This SPH multiple-fluid simulation method has the following properties.

—*Versatile Mathematical Model.* The mathematical model is aimed to simultaneously capture a whole range of multiple-fluid phenomena, including mixing and unmixing effects between miscible and immiscible phases or components, diffusion effects, chemical reaction, etc. The model should also enable flexible inclusion/exclusion of different mixing/unmixing phenomena and multiphysics interactions.

—*Robust Numerical Scheme.* The numerical simulation scheme is aimed to be robust and stable under widely varying parameter settings for animators and game designers, without requiring them to understand Computational Fluid Dynamics (CFD).

—*Practical Implementation.* The new method is designed to be easy to implement as an extension to existing single-phase fluid simulators, without adding high computational cost. The simulation of multiple-fluid flow should be simple and intuitive to set up and should not require excessive parameter tuning.

The rest of the article is organized as follows. The related work is reviewed in Section 2, where we also further discuss the relation and difference between our work and previous research. In Section 3, we introduce the mixture model of multiple-fluid flow, as well as how we improve it to better suit graphics applications. Based on the mixture model, a novel SPH simulation scheme is presented in Section 4, after which implementation details are given in Section 5. A number of multiple-fluid flow examples are illustrated in Section 6, which include mixing of miscible and immiscible liquids, unmixing due to centrifugal force, chemical reaction and phase transition, etc.

2. PREVIOUS WORK

In computer graphics, multiple-fluid simulation has received increasing attention in the past decade. Most notably, inter-facial flows have been extensively studied. Following the Lagrangian approach, Premoze et al. [2003] presented the Moving-Particle Semi-implicit (MPS) method to simulate immiscible fluids, and Solenthaler and Pajarola [2008] employed an improved SPH scheme to deal with high-density contrast between immiscible fluids. Many more works have been done using grid-based fluid solvers, including gas bubbles in liquid [Kim et al. 2007; Hong et al. 2008; Busaryev et al. 2012] and interacting fluids [Hong and Kim 2005; Losasso et al. 2006; Kim 2010; Boyd and Bridson 2012; Misztal et al. 2012]. Continuously tracking the interfaces between different phases or components is essential for inter-facial flow simulations. Many of these interface tracking techniques are related to the level-set method and the volume-of-fluid method (see, e.g., Hong and Kim [2005], Mihalef et al. [2006], Losasso et al. [2006], Kim [2010], and Boyd and Bridson [2012]), while the finite element method has also been used recently to directly capture the interface with a moving mesh [Misztal et al. 2012].

For multiple-fluid systems involving miscible fluids, the concept of volume fraction was first introduced into the graphics community by Müller et al. [2005] to represent the spatial distribution of different phases or components. Both grid-based solvers [Kang et al. 2010; Bao et al. 2010] and the SPH solver [Liu et al. 2011] have been coupled with the volume fraction representation to simulate multiple-fluid flows. All of these works assumed different phases or components move at the same bulk velocity as the mixture and mixing is only caused by the diffusion effect due to concentration difference. Doing so completely ignores the mixing and unmixing effects in multiple fluid that are primarily driven by flow motions and force distributions. At the cost of increased memory requirement, the Lattice Boltzmann Method (LBM) has also been adapted to deal with multiple-fluid flows without clear interfaces [Zhu et al. 2006; Park et al. 2008]. Recently, Nielsen and Osterby [2013] adapted a two-phase flow model to simulate water spray. In contrast to earlier works, our volume-fraction-based SPH formulation, while easy to implement, can faithfully capture complex mixing and unmixing phenomena due to relative motions, turbulent interactions, and varying force distribution among multiple fluids.

The following is also relevant to this work in the general context of multiple-fluid phenomena: Mullen et al. [2007] presented a Eulerian geometry processing technique that can mimic certain effects of miscible fluids. Cleary et al. [2007] presented a particle-based method to generate realistic visual effects of bubbles. Kim et al. [2010] modeled the dispersed bubble flow using a continuous fraction field. Ihm et al. [2004] and Kang et al. [2007] considered flow phenomena coupled with chemical reactions using molar concentration description to simulate gaseous chemical kinetics. Keiser et al. [2005] and Solenthaler et al. [2007] addressed the visual effects of melting and solidification. Ando and Tsuruno [2010] described the simulation of 2D multiphase flows using vector graphs. Kim et al. [2012] used SPH particles in grid-based liquid simulation to control bubble shapes. Ihmsen et al. [2012] proposed a postprocessing method to add spray, foam, and air bubbles to particle-based fluids. Other than simulation, Gregson et al. [2012] introduced a 3D imaging method based on stochastic tomography to capture real-world liquid mixing and dye immersion. Their results captured from real-world fluids can have much higher resolutions than graphics

Table I. Definition of Symbols

Symbol	Meaning
α_k	volume fraction of phase k
c_k	mass fraction of phase k
ρ_k, ρ_m	rest density of phase k and the mixture
$\mathbf{u}_k$, $\mathbf{u}_m$	velocity of phase k and the mixture
p_k, p_m	pressure acting on phase k and the mixture
$\mathbf{T}_k$, $\mathbf{T}_m$	stress tensor on phase k and the mixture
$\mathbf{u}_{mk}$	drift velocity
$\mathbf{g}$	gravity
ρ_{mj}, m_j	rest density and rest mass of particle j
$\overline{\rho_j}$	interpolated density of particle j
$W(\mathbf{r}, h)$	smoothing kernel function
∇W_{ij}	short for $\nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h)$
α_{ki}, α_{kj}	α_k value of the i-th, j-th particle
$\mathbf{u}_{mi}$, $\mathbf{u}_{mj}$	$\mathbf{u}_m$ value of the i-th, j-th particle
$\mathbf{u}_{mki}$, $\mathbf{u}_{mkj}$	$\mathbf{u}_{mk}$ value of the i-th, j-th particle
$\mathbf{r}_i$, $\mathbf{r}_j$	position of the i-th, j-th particle
μ_k	viscosity of phase k
μ_i, μ_j	aggregate viscosity of particles i, j
κ, τ, σ	constant coefficients

simulation, but application of such methods may be limited by its inflexibility in scene construction and tuning.

Multiple-fluid flows have been extensively studied in the context of computational fluid dynamics (CFD) for several decades, primarily driven by oil and gas, chemical engineering, and nuclear power industries. Most commercial CFD packages for multiphase or multicomponent flows are based on grid-based fluid solvers while, more recently, the SPH approach has been applied to simulate inter-facial flows [Colagrossi and Landrini 2003; Hu and Adams 2006; Monaghan and Rafiee 2013]. Various mathematical models have been developed to quantitatively describe multiple-fluid flows [Kolev 2005; Yeoh and Tu 2009; Crowe et al. 2011], mainly including the homogeneous model, the mixture model, and the full multiphase model. We introduce the concept of "drift velocity" from the grid-based mixture model and adapt it for SPH formulations by incorporating appropriate pressure relationships and modifications, thereby achieving stable and efficient SPH multiple-fluid simulation.

3. THE MIXTURE MODEL OF MULTIPLE-FLUID FLOW

In this section we briefly recap the *mixture model* [Manninen et al. 1996; Yeoh and Tu 2009], the most widely used mathematical model in engineering for grid-based multiple-fluid flows. Section 3.1 summarizes the governing equations of the mixture model, while Section 3.2 describes the drift velocity required in the model. Later in Section 4, we describe how we extend this mixture model using SPH formulations to more efficiently support visual applications. For simplicity, individual phases or components in a multiple-fluid flow are uniformly referred to as phases for the rest of the article.

3.1 Governing Equations

In the mathematical theory of multiple-fluid flow, the presence of a phase k is represented by its own volume fraction α_k (the relative fraction of an infinitesimal volume it occupies) and velocity $\mathbf{u}_k$, and the continuity and momentum equations for each phase k are

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

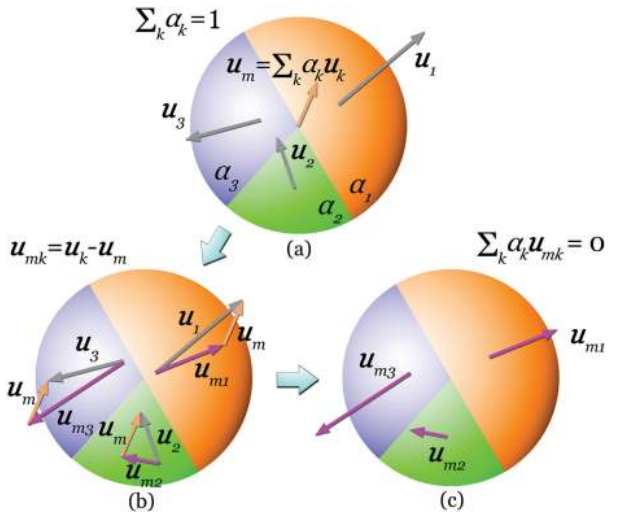


Fig. 1. Suppose a mixture has three phases with unit density, the multiple-fluid variables are illustrated above. (a) Volume fraction α_k, and relationship between the phase velocities $\mathbf{u}_k$ and the mixture velocity $\mathbf{u}_m$; (b) obtaining drift velocities $\mathbf{u}_{mk}$ from $\mathbf{u}_k$ and $\mathbf{u}_m$; (c) drift velocities $\mathbf{u}_{mk}$.

$$\frac{\partial}{\partial t}(\alpha_k \rho_k \mathbf{u}_k) + \nabla \cdot (\alpha_k \rho_k \mathbf{u}_k \mathbf{u}_k) = \alpha_k \rho_k \mathbf{g} - \alpha_k \nabla p_k + \nabla \cdot (\alpha_k \mathbf{T}_k) + \mathbf{F}_k, \tag{2}$$

where ρ_k is the rest density of phase k (assumed as constant), p_k the pressure, $\mathbf{g}$ the external body forces such as gravity, $\mathbf{T}_k$ the viscous stress tensor, and $\mathbf{F}_k$ the inter-facial momentum source. The preceding equations are similar to that of the single-phase flow, except for the last term in Eq. (2). The term $\mathbf{F}_k$ accounts for the interactions between phases, such as drag and frictional forces. In the graphics community, the multiple-fluid flow model defined in Eqs. (1) and (2) was recently adopted by Nielsen and Osterby [2013] in the case of a two-phase flow to simulate water spray. The volume fractions α_k are bounded between 0 and 1 and must add up to 1:

$$\sum_k \alpha_k = 1, \alpha_k \geq 0. \tag{3}$$

The continuity and momentum equations for the mixture follow from Eqs. (1) and (2) by summing over the phases

$$\frac{\partial}{\partial t}\rho_m + \nabla \cdot (\rho_m \mathbf{u}_m) = 0, \tag{4}$$

$$\frac{\partial}{\partial t}(\rho_m \mathbf{u}_m) + \nabla \cdot (\rho_m \mathbf{u}_m \mathbf{u}_m) = -\nabla p_m + \rho_m \mathbf{g} + \nabla \cdot \mathbf{T}_m + \nabla \cdot \mathbf{T}_{Dm}, \tag{5}$$

where $\rho_m = \sum_k \alpha_k \rho_k$ is the mixture density, $\mathbf{u}_m = \frac{1}{\rho_m} \sum_k \alpha_k \rho_k \mathbf{u}_k$ is the mixture velocity (i.e., the velocity at the mass center), the mixture's pressure p_m is defined by the relation $\nabla p_m = \sum_k \alpha_k \nabla p_k$, the mixture's viscous stress tensor $\mathbf{T}_m$ is defined to satisfy $\nabla \cdot \mathbf{T}_m = \sum_k \nabla \cdot (\alpha_k \mathbf{T}_k)$, and the term $\mathbf{T}_{Dm} = -\sum_k \alpha_k \rho_k \mathbf{u}_{mk} \mathbf{u}_{mk}$ is derived from the left-hand side of the momentum equation, representing the convective momentum transfer between phases. Here, the *drift velocity* $\mathbf{u}_{mk}$ is defined as

$$\mathbf{u}_{mk} = \mathbf{u}_k - \mathbf{u}_m. \tag{6}$$

The *drift velocity* $\mathbf{u}_{mk}$ denotes the velocity of phase k relative to the centre of the mixture mass. The interaction forces $\mathbf{F}_k$ do not appear explicitly in the momentum equation (5) because they are canceled when summing over all phases. Using a simple case with three phases of unit density (i.e., $\rho_k = 1$ for all three phases), Figure 1

illustrates the concepts and relationships of the multiple-fluid variables.

Substituting Eq. (6) into Eq. (1), the phase velocity $\mathbf{u}_k$ can be eliminated from the continuity equation of phase k:

$$\frac{\partial \alpha_k}{\partial t} + (\mathbf{u}_m \cdot \nabla)\alpha_k = -\alpha_k \nabla \cdot \mathbf{u}_m - \nabla \cdot (\alpha_k \mathbf{u}_{mk}). \tag{7}$$

Substituting Eq. (4) into Eq. (5), the momentum equation of mixture can be reorganized as

$$\frac{\partial}{\partial t}\mathbf{u}_m + (\mathbf{u}_m \cdot \nabla)\mathbf{u}_m = -\frac{\nabla p_m}{\rho_m} + \mathbf{g} + \frac{\nabla \cdot \mathbf{T}_m}{\rho_m} + \frac{\nabla \cdot \mathbf{T}_{Dm}}{\rho_m}. \tag{8}$$

Defined in Eqs. (7) and (8) are the governing equations of the mixture model for multiple-fluid flows. Here, the spatial distribution of each phase k is fully represented by its volume fraction α_k, hence it is not necessary to track the interfaces between different phases.

3.2 Drift Velocity

It is clear from Eq. (7) that the nonuniform distribution of velocity fields will lead to changes in the volume fraction of each phase. In a multiple-fluid flow, this motion-induced mixing effect is quite intuitive: different phases move at different velocities in the mixture, and their discrepant motions will naturally result in relative mass migration. The drift velocities $\mathbf{u}_{mk}$ play a key role in this interaction mechanism responsible for various miscible and immiscible phenomena.

Based on the assumption of local equilibrium and appropriate drag force approximations, the drift velocities $\mathbf{u}_{mk}$ defined in Eq. (6) can be solved analytically; the rigorous mathematical derivation can be found in Manninen et al. [1996]. For simplicity, the analytical expression of drift velocities is directly given as

$$\begin{aligned}\mathbf{u}_{mk} \;=\; & \tau\left(\rho_k - \sum_{k'} c_{k'}\rho_{k'}\right)\mathbf{a} - \tau\left(\nabla p_k - \sum_{k'} c_{k'}\nabla p_{k'}\right) \\ & -\sigma\left(\frac{\nabla \alpha_k}{\alpha_k} - \sum_{k'} c_{k'}\frac{\nabla \alpha_{k'}}{\alpha_{k'}}\right),\end{aligned} \tag{9}$$

where τ and σ are user-defined constant coefficients to be discussed later and $c_k = \frac{\alpha_k \rho_k}{\rho_m}$ is the mass fraction of the k-th component. The acceleration $\mathbf{a}$ is

$$\mathbf{a} = \mathbf{g} - (\mathbf{u}_m \cdot \nabla)\mathbf{u}_m - \frac{\partial \mathbf{u}_m}{\partial t}, \tag{10}$$

which denotes the difference between the gravity acceleration and the substantial derivative of the mixture velocity.

To compute the drift velocities following Eq. (9), the relation between the phase pressure p_k and the mixture pressure p_m must also be provided. The standard mixture model mainly deals with immiscible fluids and the following pressure relation has been widely adopted:

$$p_k = p_m. \tag{11}$$

For immiscible fluids, the phase pressure p_k is identical to the mixture pressure p_m such that the second term in Eq. (9) vanishes. The intuitive explanation of it is that, for immiscible fluids where pressure equilibrium is established between phases, the uniform pressure shared with the mixture does not cause the immiscible phases to move into each other. To cope with totally miscible fluids in graphics applications, we extend the standard mixture model by incorporating the following pressure relation [Kolev 2005]:

$$p_k = \alpha_k p_m. \tag{12}$$

For miscible fluids, phase pressures p_k differ from each other dependent on their volume fractions, thus miscible phases are accelerated within the mixture mass to move into each other.

Determined in Eq. (9), the drift velocity $\mathbf{u}_{mk}$ contains three terms. The first term accounts for the inertia effect, and in particular the velocity differences caused by body forces are modeled by this term. The second term accounts for the pressure effect, that is, within the mixture mass a phase accelerates in the direction from high pressure to low pressure. The third term accounts for the diffusion effect, that is, a phase tends to move from more concentrated regions to less concentrated regions. The constant coefficients τ and σ are essentially the strength factors of these fluid-dynamics effects. Specifically, higher τ values will cause stronger inertia and pressure effects (thus faster unmixing and mixing speeds due to these two effects), higher σ values will cause stronger diffusion effects (thus faster mixing speed due to the diffusive effect), and vice versa. In our work τ varies between 10^{-8} and 10^{-6}, and σ is around 10^{-4} to 10^{-3}. Further discussions of these three multiphase transportation effects are provided in Section 5.3.

Once the drift velocities $\mathbf{u}_{mk}$ are determined following Eq. (9), the solution of the governing Eqs. (7) and (8) is reduced to solving for the mixture velocity $\mathbf{u}_m$ (instead of all phase velocities $\mathbf{u}_k$) and phase volume fractions α_k. This significantly reduces the computational cost of solving multiple-fluid flows. Owing to the high efficiency and versatility of the mixture model, it is widely adopted in commercial CFD packages for multiple-fluid flows, including Eulerian fluid solvers ANSYS CFX and FLUENT.

4. SPH FORMULATION

In contrast to the grid-based formulations presented in Manninen et al. [1996], we use SPH particles to discretize the multiple-fluid system such that the SPH particles carry the mixture mass and move at the mixture velocity $\mathbf{u}_m$. These mixture particles also carry all the physical quantities associated with individual phases.

In the governing equations of the mixture model, namely Eqs. (7) and (8), the left-hand side of Eq. (8) denotes the substantial derivative of the mixture velocity $\mathbf{u}_m$, while the left-hand side of Eq. (7) represents the substantial derivative of the volume fraction of phase k, also with respect to the mixture velocity $\mathbf{u}_m$. In addition, it is noted that after the drift velocities $\mathbf{u}_{mk}$ are computed following the analytical solution in Eq. (9), the solutions to Eq. (7) and Eq. (8) can be decoupled if an explicit time-integration scheme is adopted. These favorable Lagrangian properties of the mixture model fit nicely into the SPH approach.

In this section we provide the SPH formulation of the governing Eqs. (7) and (8), for which we start from the drift velocity solution in Eq. (9). For each mixture particle i, we have

$$(\nabla p_k)_i = \sum_j \frac{m_j}{\bar{\rho}_j}(p_{kj} - p_{ki})\nabla W_{ij}, \tag{13}$$

$$(\nabla \alpha_k)_i = \sum_j \frac{m_j}{\bar{\rho}_j}(\alpha_{kj} - \alpha_{ki})\nabla W_{ij}, \tag{14}$$

where the summation is performed over all neighborhood particles j and $\nabla W_{ij} = \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h)$ is the gradient of the smoothing kernel function with support h. Adopting the formulation in Müller et al. [2003], we use the poly6 kernel for density interpolation and the spiky kernel for all other calculations involving the derivative of the smoothing kernel function. In Eqs. (13) and (14), and all equations thereafter, m_j and $\bar{\rho}_j$ in the summation over particles j represent the mass and interpolated density of the mixture particle j, respectively. Eqs. (13) and (14) are standard symmetric SPH

formulations for gradient terms of scalars. Using the ∇p_k and $\nabla \alpha_k$ expressions, the drift velocity $\mathbf{u}_{mk}$ can be computed from Eq. (9).

Then we examine Eqs. (7) and (8). First, for Eq. (7), it should be noted that we cannot assume $\nabla \cdot \mathbf{u}_m = 0$ here. For multiple-fluid flows, volume fractions α_k change both over time and over space, and phase velocities $\mathbf{u}_k$ also differ from each other. Therefore, neither the divergence of the mixture velocity nor that of the phase velocity is zero, even when all phases are incompressible. To describe the continuity of multiple-fluid flows we must refer to the fundamental mass conservation law in Eq. (1).

For each mixture particle i, directly applying the SPH approximation rule to the right-hand-side terms in Eq. (7) yields

$$(\alpha_k \nabla \cdot \mathbf{u}_m)_i = \alpha_{ki} \sum_j \frac{m_j}{\bar{\rho}_j} \mathbf{u}_{mj} \cdot \nabla W_{ij}, \tag{15}$$

$$(\nabla \cdot (\alpha_k \mathbf{u}_{mk}))_i = \sum_j \frac{m_j}{\bar{\rho}_j} \alpha_{kj} \mathbf{u}_{mkj} \cdot \nabla W_{ij}. \tag{16}$$

However, the previous SPH approximations are not symmetric and do not lead to stable simulation. Based on previous SPH stabilization techniques, we modify them and propose to use the following symmetric formulations (see Appendix A for derivation):

$$(\alpha_k \nabla \cdot \mathbf{u}_m)_i = \sum_j \frac{m_j}{\bar{\rho}_j} \frac{\alpha_{kj} + \alpha_{ki}}{2} (\mathbf{u}_{mj} - \mathbf{u}_{mi}) \cdot \nabla W_{ij}, \tag{17}$$

$$(\nabla \cdot (\alpha_k \mathbf{u}_{mk}))_i = \sum_j \frac{m_j}{\bar{\rho}_j} (\alpha_{kj} \mathbf{u}_{mkj} + \alpha_{ki} \mathbf{u}_{mki}) \cdot \nabla W_{ij}. \tag{18}$$

Intuitively, Eq. (17) reflects the change of volume fraction due to the aggregate motion of the mixture, namely the relative motion of the mixture particles, while Eq. (18) reflects the change of volume fraction due to the discrepancy between phase velocities, that is, the difference of drift fluxes $\alpha_k \mathbf{u}_{mk}$ between particles.

Second, we examine the last term in Eq. (8). It represents the convective momentum change due to the drift velocities. We directly list next its symmetric formulation, and provide the detailed derivation in Appendix A:

$$\begin{aligned}(\nabla \cdot \mathbf{T}_{Dm})_i \; = \; & -\sum_j \frac{m_j}{\bar{\rho}_j} \sum\nolimits_k [\rho_k (\alpha_{kj} \mathbf{u}_{mkj} (\mathbf{u}_{mkj} \cdot \nabla W_{ij}) \\ & + \alpha_{ki} \mathbf{u}_{mki} (\mathbf{u}_{mki} \cdot \nabla W_{ij}))]. \end{aligned} \tag{19}$$

Finally, we deal with the remaining terms in Eq. (8). These terms are similar to those of the single-phase flow and hence their treatments are essentially no different to the single-phase SPH formulation. The pressure gradient in Eq. (8) can be expressed by

$$(\nabla p_m)_i = \sum_j m_j \frac{p_{mi} + p_{mj}}{2\, \bar{\rho}_j} \nabla W_{ij}. \tag{20}$$

The formulation proposed in Cleary [1996] and González et al. [2009] is adopted to compute the divergence of viscosity tensor

$$(\nabla \cdot \mathbf{T}_m)_i = \sum_j \frac{m_j}{\bar{\rho}_j} (\mu_i + \mu_j)(\mathbf{u}_{mj} - \mathbf{u}_{mi}) \frac{(\mathbf{r}_j - \mathbf{r}_i) \cdot \nabla W_{ij}}{(\mathbf{r}_j - \mathbf{r}_i)^2}, \tag{21}$$

where $\mu_j = \sum_k \alpha_{kj} \mu_k$ is the aggregate viscosity of particle j. This formulation is obtained from the integral representation of second-order derivatives of the viscosity term.

Defined in Eqs. (7) and (8), the governing equations of the mixture model share a similar format as the single-phase flow, with the inclusion of the drift velocity term. This similarity allows us to easily apply the state-of-the-art techniques developed for single-phase flow. For the calculation $\bar{\rho}_i$, the interpolated density of the mixture particles, the standard SPH formulation is

$$\bar{\rho}_i = \sum_j m_j \nabla W_{ij}. \tag{22}$$

Recently, Solenthaler and Pajarola [2008] proposed to use a modified density calculation method for immiscible multiple-fluid simulations with high density ratio. Their method changes the density interpolation equation to

$$\bar{\rho}_i = \sum_j m_i \nabla W_{ij}, \tag{23}$$

and then substitutes it into the standard formulation to achieve desired simulation results. Both Eq. (22) and Eq. (23) can be used in our SPH framework. In our experiments, for miscible fluid simulations featuring smooth changes of particle rest densities and without interfaces, the standard approach produces better results, especially in cases with relatively low-density contrast ratios.

In SPH simulations, the pressure value of each particle is related to the interpolated particle density through the equation of state. In the standard SPH scheme, the following linear relation is adopted

$$p_{mi} = \kappa(\bar{\rho}_i - \rho_{mi}), \tag{24}$$

where κ is the gas constant. In more recent simulation methods such as the Weakly Compressible SPH (WCSPH) [Becker and Teschner 2007], the Tait equation is usually adopted to enhance incompressibility of the fluid appearance

$$p_{mi} = \frac{\kappa \rho_{mi}}{\gamma} \left(\left(\frac{\bar{\rho}_i}{\rho_{mi}} \right)^{\gamma} - 1 \right), \tag{25}$$

where $\gamma = 7$. Compared to the standard approach, the Tait equation effectively results in much higher pressure changes with the same amount of density variation. Again, both state equations can be used in our approach to remain compatible with the standard simulation method and the WCSPH scheme.

5. IMPLEMENTATION

This section further explains the implementation issues for the SPH formulation of multiple-fluid flows, after which the algorithm framework is summarized.

5.1 Volume Fraction Correction

The bound of α_k described in Eq. (3) is not automatically satisfied when solving Eq. (7). It is necessary to introduce a correction step after advancing the volume fraction.

(1) If $\alpha_k < 0$, set $\alpha_k = 0$.
(2) Rescale α_k values for all components such that $\sum_k \alpha_k = 1$.

In order to ensure equilibrium after correcting the volume fraction, it is also necessary to perform a pressure adjustment for all phases. Intuitively, the occupance of a volume fraction exceeding one (i.e., $\alpha_k > 1$) means that, in the current time step, the flux of the k-th phase entering the mixture particle is larger than allowed and the fluid is overcompressed. However, if the time step is set sufficiently small, the pressure computed from the particle density as Eq. (24) or Eq. (25) will raise and sequentially stop the k-th phase from entering the mixture particle, so that the α_k value remains below one. Hence, the adjustment of the α_k value at the next time step

should relate to a pressure adjustment for the current time step as well.

For a quantitative formulation, we should calculate the derivative of pressure with respect to volume fraction. Eq. (24) can be rewritten as

$$p_{mi} = \kappa \left(\sum_j m_j W_{ij} - \sum_k \alpha_{ki} \rho_k \right). \quad (26)$$

The derivative of pressure with respect to volume fraction is

$$\frac{\partial p_{mi}}{\partial \alpha_{ki}} = -\kappa \rho_k \quad (27)$$

and this leads to the following relation that links the change of volume fraction to the change of pressure

$$\Delta p_{mi} = \sum_k -\kappa \rho_k \Delta \alpha_{ki}. \quad (28)$$

Similarly, for the Tait equation in the WCSPH approach, taking the partial derivative of Eq. (25) over α_{ki} yields

$$\frac{\partial p_{mi}}{\partial \alpha_{ki}} = -\frac{\kappa \rho_k}{\gamma} \left((\gamma - 1) \left(\frac{\bar{\rho}_i}{\rho_{mi}} \right)^{\gamma} + 1 \right), \quad (29)$$

giving a different form of pressure adjustment:

$$\Delta p_{mi} = \sum_k -\frac{\kappa \rho_k}{\gamma} \left((\gamma - 1) \left(\frac{\bar{\rho}_i}{\rho_{mi}} \right)^{\gamma} + 1 \right) \Delta \alpha_{ki}. \quad (30)$$

The minus sign in Eqs. (28)–(30) assures correct direction of adjustment. The adjusted pressure is then given by

$$\tilde{p}_{mi} = p_{mi} + \Delta p_{mi}. \quad (31)$$

We then use the adjusted pressure $\tilde{p}_{mi}$ for gradient calculation in Eq. (20).

The purpose of the preceding correction step is to ensure that Eq. (3) holds in line with the underlying physics. There may be other, more sophisticated correction approaches, however, in practice we found simulations with the proposed correction steps give good results while keeping the physical meaningfulness required by Eq. (3).

5.2 Chemical Reaction

One advantage of combining the volume fraction representation with the SPH representation is the convenience of being able to deal with chemical reactions between phases by simply adding an in-particle rebalance step at the end of each simulation loop. For illustration, a simple case of reactants A and B reacting to produce resultant C is considered here. For all particles carrying both phases A and B, the masses of A and B are decreased by a controlled amount at the end of each simulation loop, while at the same time the mass of C is accordingly rebalanced. That is,

$$\Delta(m_C) = -\Delta(m_A) - \Delta(m_B) = -\Delta(\alpha_A)\rho_A V - \Delta(\alpha_B)\rho_B V, \quad (32)$$

where V is the volume of the mixture particle before reaction. At each time step, the amount of reactants $\Delta\alpha_A$ and $\Delta\alpha_B$ is set proportionally to the volume fractions of both source phases. Specifically, for the reaction $x\mathrm{A} + y\mathrm{B} = z\mathrm{C}$, the reactants are set as $\Delta\alpha_A = xC_r\alpha_A\alpha_B$ and $\Delta\alpha_B = yC_r\alpha_A\alpha_B$, where C_r is a coefficient controlling the reaction speed. We also ensure that the reacted amount of reactants does not exceed the current volume fraction value of each phase. This chemical rebalance procedure can also involve total rest volume change to the mixture particle. Since the new phase masses are known, the new volume fraction value of each phase within the mixture particle is recalculated after the rebalancing, as well as the aggregate rest density of the mixture particle. The momentum is automatically preserved, since both particle mass and aggregate velocity of the mixture particle are not affected.

5.3 Mixing and Unmixing of Immiscible and Miscible Fluids

Uniform Particle Description. Various mixing and unmixing effects are captured by the drift velocity solved in Eq. (9). For immiscible fluids, the second term in Eq. (9) automatically vanishes by setting $p_k = p_m$, and the inertia effect modeled by the first term will separate different phases as the mixture flows. Note that this does not mean immiscible fluids will always have a sharp interface between phases, since in reality even immiscible phases can get temporarily mixed in a vibrantly flowing mixture, such as in the form of dispersed phases in a suspension. Such temporary mixing of vibrant immiscible fluids is not an artificial smoothing and it is the natural reflection of the real world, where sharp interfaces will begin to appear once the flow motion is no longer violent. The inertia term provides an unmixing mechanism for the multiple-fluid flow. For miscible fluids, the second term in Eq. (9) is nonzero and the pressure effect will work against the unmixing trend caused by the inertia effect and keep the phases mixed. This pressure-driven mixing mechanism should not be confused with diffusion, which is modeled separately by the third term in Eq. (9). The diffusion effect is purely driven by concentration difference and will monotonously eliminate any volume fraction difference across the space. The mixing caused by the pressure effect is, however, related to the flow motion. In particular, when the flow motion stops, the inertia term and the pressure term cancel each other, thus sequentially terminating the pressure-driven mixing. Under this framework, purely immiscible fluids are simulated by setting the pressure relation $p_k = p_m$, while purely miscible fluids are simulated by setting $p_k = \alpha_k p_m$.

Particle Labeling Approach. Our approach can also treat the miscible/immiscible behaviors in a slightly different manner, when dealing with a simulation that contains groups of phases that are miscible within the same group but are immiscible between different groups, or when it is desired to forbid volume fraction transfer between different groups that are immiscible to each other. In such cases, the traditional assumption that immiscible phases cannot exist together within a mixture particle is used and the particles are labeled into different "miscible groups". The calculation of related terms will then be limited to particles within the same group. That is, in the calculation of Eqs. (17) and (18), the summation should only be performed for the set of particles j that are in the same miscible group as the current particle i, eliminating the volume fraction transition between different groups; and in the calculation of Eq. (19), the summation should also be limited to the set of particles j that are in the same miscible group as the current particle i, eliminating the momentum transition due to the volume fraction transition between different groups. If desired, one can also use Eq. (22) within the same miscible group and Eq. (23) between particles in different groups.

5.4 Algorithm Framework

The multiple-fluid system is represented by a set of mixture particles, each mixture particle i carrying aggregate values (m_i, ρ_{mi}, $\mathbf{u}_m$, μ_i) and component-wise values (α_{ki}). During each simulation loop, the SPH simulator sequentially performs the following tasks.

(1) Compute the drift velocity of each phase according to Eq. (9) using p_k calculated from Eq. (11) or Eq. (12). The SPH formulation of the gradient terms is given in Eqs. (13) and (14). The diffusion effect can be switched on by assigning a positive value to the constant σ, and off by setting σ to zero.

(2) Advect volume fraction values according to Eq. (7), where the relevant SPH formulations are given in Eqs. (17) and (18).

(3) Check the bound of volume fraction according to Eq. (3) and, if the bound is invalidated, correct the volume fraction within the particle and calculate the pressure adjustment accordingly as described in Section 5.1. For particles with a corrected volume fraction, update into the pressure term the pressure adjustment as Eq. (31).

(4) Calculate the acceleration of the mixture particle according to Eq. (8). SPH formulations of the related terms are provided in Eqs. (19)–(21).

(5) Advect mixture particles using their accelerations and velocities.

(6) In the event of chemical reaction, rebalance the phase mass within each involved particle as in Eq. (32) and recalculate the volume faction value of each phase, as well as the aggregate rest density.

Essentially, this algorithm framework is very similar to that of the single-phase fluid simulation. The main difference is that multiple-fluid simulation involves volume fraction advance/correction. Moreover, a new term on the right-hand side of the governing equations requires the drift velocities to be calculated in advance, which is given in an analytical form. The runtime of each step depends on the number of particles and the number of phases; we discuss his aspect in detail in Section 6.1.

5.5 Time Stepping

The Courant-Friedrichs-Lewy (CFL) condition is adopted for determining the time step. Similar to Monaghan [1992], Desbrun and paule Gascuel [1996], and Becker and Teschner [2007], the time step is controlled by

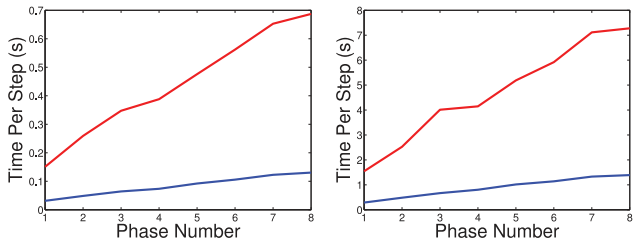
$$\Delta t = \min\left\{\min_i\left(\frac{0.25h}{|\mathbf{f}_i|}\right), \frac{0.4h}{c_s + 0.6(c_s + 2\cdot\max_i \mu_i)}\right\}, \quad (33)$$

where $\mathbf{f}_i$ denotes external forces acting on the mixture particle and c_s is the sound speed denoting the maximum possible particle speed in the fluid motion, which is related to the gas constant κ in the equations of state and has $c_s \propto \sqrt{\kappa}$. Simply speaking, the CFL condition requires that a particle should not travel more than a certain fraction of its smoothing radius in one time step. In Desbrun and paule Gascuel [1996] and Dagenais et al. [2012] the time step can be bounded using $\Delta t = \frac{0.3h}{|\max_i \mathbf{u}_{mi}|}$, which is a simpler form of the second term in Eq. (33).

In multiple-fluid simulation, the drift velocity also adds a constraint to the time step following the CFL condition:

$$\Delta t = \frac{0.3h}{|\max_i(\max_k \mathbf{u}_{mki})|}. \quad (34)$$

Then we choose the smaller Δt computed from Eqs. (33) and (34) as the upper bound of time steps. The examples in Section 6 typically run at time steps around 10^{-3} second. The viscous armadillo example runs at time steps around 10^{-5} second because its large viscosity dominates the calculation of the time step.



(a) (b)

Fig. 2. Performance under different particle numbers and phase numbers. Results are separated into two subdiagrams to avoid overcompression in the vertical axis. The runtime of each step is linear to the total particle number and, given fixed particle number, the runtime is sublinear to the number of phases. (a) blue: 48,000 particles; red: 239,000 particles; (b) blue: 476,000 particles; red: 2,368,000 particles.

Table II. Performance

Example Name	Phase Number	Particle Number	average time (second/step)
Dam-breaking	3	344,000	0.477
Armadillo	2	313,000	0.339
Reacting Swirl	4	198,000-418,000	0.247-0.579
Unmixing	4	231,000	0.376
Vaporization	4	402,000	0.382
Rainbow Wave	8	756,000	1.889

6. PERFORMANCE AND RESULTS

6.1 Performance Analysis

GPU parallelization of standard SPH simulators is quite straightforward. A simple scheme uses a uniform grid structure to simplify and accelerate the neighbor search in the GPU. At the start of each time step, SPH particles are assigned into the grid structure and sorted based on their positions. The neighbor search only needs to consider particles in adjacent grids afterwards, which can be continuously traversed in the sorted order. Then all computational tasks can be parallelly executed over the particles. The GPU implementation of the mixture model follows exactly the same procedure. The proposed SPH multiple-fluid simulator has been implemented using CUDA 5. Each step described in Section 5.4 is executed by a CUDA kernel function that parallels the computing task over each fluid particle. All the variables including property values (e.g., aggregate velocity) and intermediate values (e.g., drift velocity of each phase) related to each particle are stored and updated in the graphical memory during the simulation loops in order to minimize the CPU-GPU communication.

The computational efficiency is mainly determined by the number of mixture particles and the number of phases adopted in the simulation. Typically, for a three-phase flow simulated by using 344,000 mixture particles and the simulation runs at 0.477 second per time step on a NVIDIA GeForce GTX 680 4GB GPU. We analyze runtime under different particle numbers and phase numbers. The performance is shown in Figure 2. The runtime of each step is linear to the total particle number and, given a fixed particle number, the runtime is sublinear to the number of phases, that is, each extra phase will increase the runtime by approximately 50% of the single-phase runtime. The performance data of examples in the article are provided in Table II.

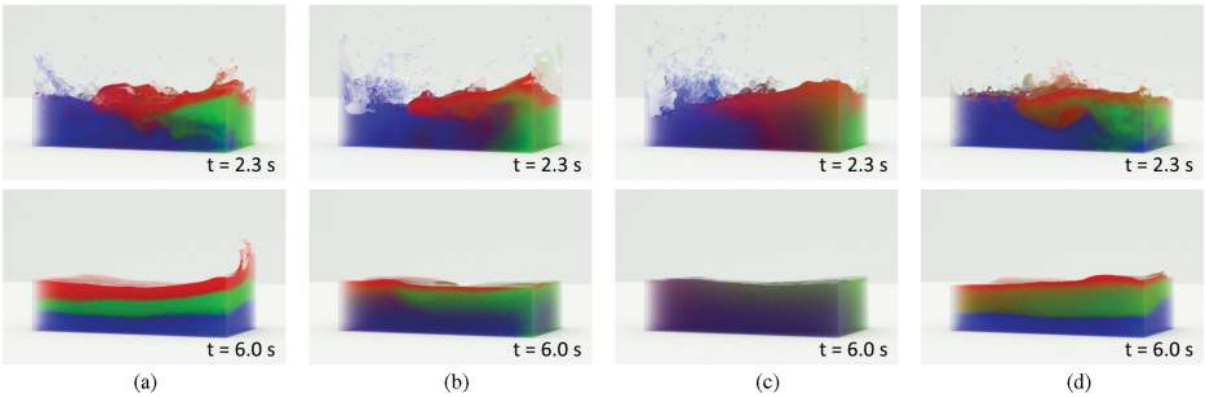


Fig. 3. Three-phase liquid dam breaking (red, green, and blue). (a) All three phases are immiscible with each other and get separated into three layers with clear interfaces; (b) all three phases are miscible with other and, with the diffusion effect disabled, get mixed due to interaction between phases; (c) all three phases are miscible with each other and, with the diffusion effect enabled, get mixed more uniformly; (d) the red and blue phases are miscible with each other, but both are immiscible with the blue phase.

6.2 Results

Using a triple dam-breaking setup, Example 1 (Figure 3) demonstrates the new method's capacity in capturing realistic phase-interacting effects with different combinations of miscible and immiscible fluids. The density ratio between the three phases is set as *Red*:*Green*:*Blue* = 1:2:3. In Figure 3(a), all three phases are immiscible with each other, whereas in Figure 3(b), all three phases are miscible with each other and the diffusion effect is disabled; in Figure 3(c), all three phases are miscible with each other and the diffusion effect is enabled; in Figure 3(d), the red and green phases are miscible with each other, but immiscible with the blue phase. Our approach successfully simulates mixing and unmixing effects in all these circumstances. As shown in Figure 3(a), the immiscible phases are clearly separated into three layers. At the end of the simulation the immiscible phases are fully separated, and the volume fraction penetrations in mixture particles near either side of the sharp color interfaces are negligible. For the miscible fluids shown in Figures 3(b) and (c), the mixing effects look smooth and natural and, due to the inclusion of the diffusion effect, the final result in Figure 3(c) is mixed more uniformly than that in Figure 3(b). The simulations in Figures 3(a), (b), and (c) adopt the uniform particle description as explained in Section 5.3, while the simulation in Figure 3(d) adopts the particle labeling approach to simulate miscible and immiscible phases interacting with each other simultaneously. For all other examples in this article, we have used the uniform particle description in the simulation.

Example 2 (Figure 4) simulates a two-phase flow with high viscosity contrast. The red armadillo formed by a highly viscous phase (1,000× viscosity and 2× density) drops into a rectangular container filled by a transparent phase (1× viscosity, 1× density, and immiscible with the red phase). The mixture flow is shown in the top row of Figure 4, where the red armadillo deforms in a highly viscous manner and does not get diluted by the transparent phase. This example shows our method can cope with very high viscosity contrast and still achieve realistic and stable multiple-fluid simulations. Then, after the red armadillo has settled at the bottom of the container, the red phase is reset to 1× viscosity, 0.5× density, and miscible with the transparent phase. The mixture flow is shown in the bottom row of Figure 4, where the now lighter red phase undergoes volume expansion and rises up rapidly, where it soon gets diluted into the transparent phase. In both stages, the diffusion effect has been disabled.

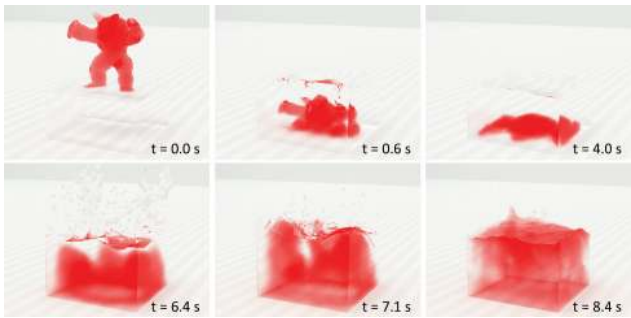


Fig. 4. Viscous armadillo. Shown in the top row: a red armadillo formed by a highly viscous phase (1,000× viscosity and 2× density) drops into a container filled by a transparent phase (1× viscosity, 1× density, and immiscible with the red phase); the red armadillo deforms in a highly viscous manner and does not get diluted. Shown in the bottom row: after the red armadillo's settlement at the bottom of the container, the red phase is reset to 1× viscosity, 0.5× density, and miscible to the transparent phase, which then undergoes volume expansion and rises up rapidly. A vibrant mixing is observed during the two-phase interaction.

In Example 3 (Figure 5), red and green liquids are injected into a cylindrical container filled with transparent liquid solvent. As the red and green liquids are injected from opposite sides of the container, they drive the liquid mixture to swirl. During mixing, red and green phases react to produce a blue liquid. All four phases (red, green, transparent, and blue) are miscible with each other. In the top row is the simulation result using our approach with the diffusion effect disabled. The center of the container largely remains occupied by the transparent solvent during the swirling motion, resulting in a rotating and dynamically evolving *S* shape, where the whole scene contains simultaneously vigorous mixing and chemical reaction. The previous multifluid approach (e.g., [Kang et al. 2010; Bao et al. 2010; Liu et al. 2011]) considers Brownian diffusion only. In such approaches the mixing between different phases and thus the chemical reaction purely relies on the existence of Brownian diffusion. Due to the physical nature of Brownian diffusion, it will gradually eliminate the polytropic color variations in multiple-fluid flows, leading to an undesired homogeneous appearance. In the bottom row we show the simulation result of a previous

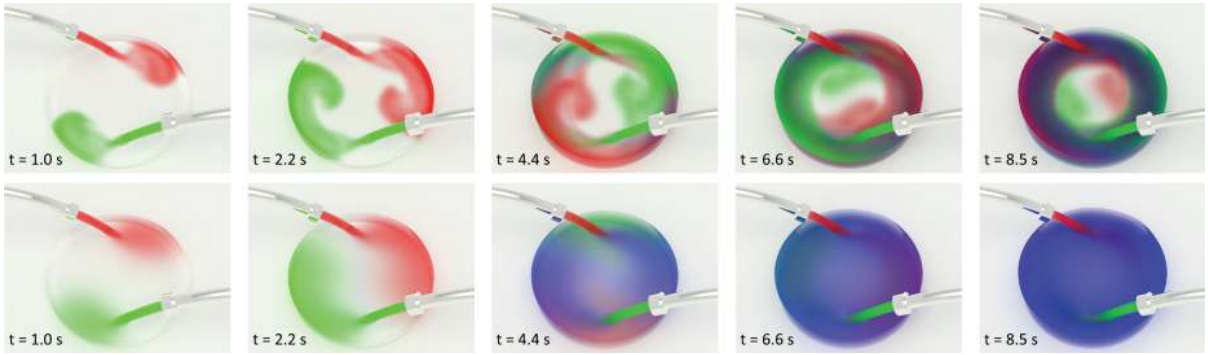


Fig. 5. Reacting swirl. Red and green liquids are injected from opposite sides into the container filled with transparent solvent, causing the mixture to swirl. Upon meeting, the red and green liquids react to produce a blue liquid. All four phases are miscible with each other. Top row: Our method, with the diffusion effect disabled, forms a rotating and dynamically evolving S symbol at the centre of the container, while the whole scene contains simultaneously vigorous mixing and chemical reaction. Bottom row: Using only Brownian diffusion to simulate mixing [Kang et al. 2010; Bao et al. 2010; Liu et al. 2011] results in a homogeneous appearance.

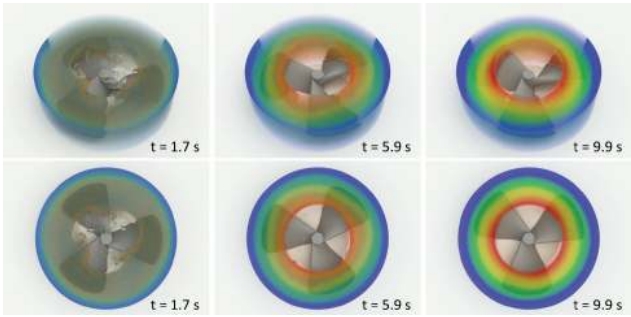


Fig. 6. Unmixing. Four immiscible liquids (red, yellow, green, and blue) are artificially set to a "fully mixed" state in a circular container with a rotating turbine at the centre. During spinning they get "fully separated" due to the centrifugal effect. Top: perspective view. Bottom: top view.

multifluid approach, where the polytropic color variations gradually attain a homogeneous appearance. It is clear in this example that our approach is able to avoid the undesired variation eliminating effect and the homogeneous appearance due to Brownian diffusion, while simultaneously keeping vigorous mixing and reaction featuring polytropic colors throughout the whole simulation.

Example 4 (Figure 6) demonstrates an unmixing process taking place in a disk-shaped container that has a spinning turbine installed at its centre. The container is filled with four immiscible phases with the density ratio *Red*:*Yellow*:*Green*:*Blue* = 1:1.5:2.5:3, and the mixture occupies about 3/4 of the volume. Initially, the four phases are set at a "fully mixed" state, resulting in a greyish color, and are unmixed due to the centrifugal effect during spinning, resulting in a colorful ring-shaped pattern. The top row in Figure 6 shows a perspective view of this unmixing process, while the bottom row shows the top view. The gravity force is not considered in this example. This unmixing effect cannot be captured by multiple-fluid simulations where only the diffusion effect is modeled, and also it is hard to achieve through the inter-facial-flow simulation approach.

Example 5 (Figure 7) simulates a four-phase vaporization process with high-density contrast. A dome is filled with transparent air and two liquids (red and green) are injected into the dome from two magic sources. The red and green liquids meet at the centre of the floor and react to produce a vapor phase that rises up towards the ceiling. In this example, all four phases are immiscible with each other and their density ratio is *Red liquid*:*Green liquid*:*Transparent air*:*Vapor phase* = 1000:1000:2:1. Under high-density contrast, the vaporization process is successfully simulated.

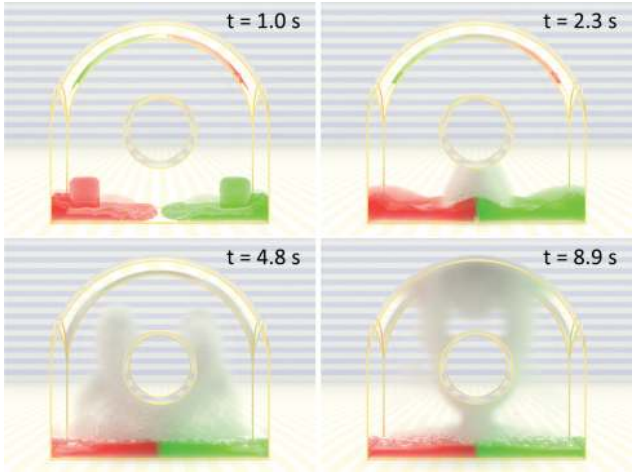


Fig. 7. Vaporization. The dome is filled with transparent air and, from two magic sources, red and green liquids are injected into the dome. The two liquids meet at the centre of the floor and react to produce a vapor phase that rises towards the ceiling. The density ratio is *Red liquid*:*Green liquid*:*Transparent air*:*Vapor phase* = 1000:1000:2:1.

In Example 6 (Figure 8), a tank with a movable wall on the left is filled with transparent liquid. To the left of the tank, there are four "reacting regions" marked in *red*, *yellow*, *blue*, and *purple*. Upon entering these reacting regions, the transparent liquid reacts to produce a new liquid with the color of the region. The wall on the left moves back and forth periodically to drive the liquid mixture to flow in the tank. Three more chemical reactions are introduced: the red liquid reacting with the yellow liquid to produce an orange liquid, the yellow liquid reacting with the blue liquid to produce a green liquid, and the blue liquid reacting with the purple liquid to produce an indigo liquid. All eight phases are set as miscible with each other, thus the mixing flow in the tank creates a lively rainbow wave with seven naturally colored streams adjacent to each

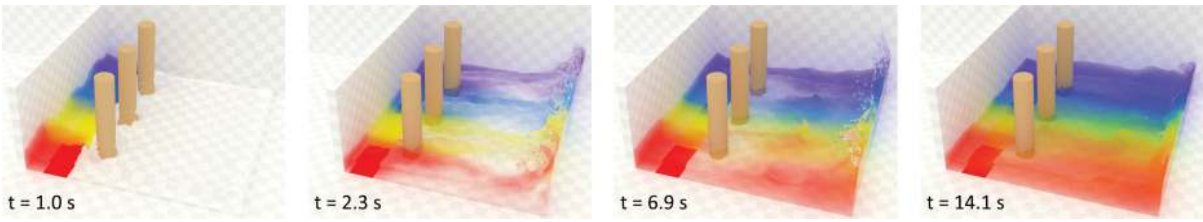


Fig. 8. Rainbow wave. The transparent liquid is propelled by the periodically moving wall on the left and, when the transparent liquid passes over the four magic regions in the left of the tank, it reacts to produce four new phases (red, yellow, blue, and purple). The adjacent new phases can also react with each other to produce another three phases (orange, green, and indigo). All eight phases are miscible with each other and the diffusion effect is disabled. A lively rainbow wave is formed in the tank.

other: *red*, *orange*, *yellow*, *green*, *blue*, *indigo*, and *purple*. Again, the diffusion effect is disabled in this example to avoid a stiff and uniform appearance.

7. CONCLUSION AND DISCUSSION

By combining the mixture model in computational fluid dynamics and the SPH method, we have developed a novel simulation approach for multiple-fluid flows. Verified in various numerical experiments, the new approach is versatile and can simultaneously capture a wide range of multiple-fluid phenomena, including mixing/unmixing of miscible and immiscible fluids, diffusion and chemical reaction, etc. The new method is robust and can achieve stable and realistic simulation under widely varying parameter settings, including high viscosity and density contrast. Implementation of our new scheme is straightforward as an extension to existing single-phase fluid simulators. Moreover, the multiple-fluid simulation can be easily set up with a minimum requirement of multiple-fluid information and without unnecessary parameter tuning. Compared to the simple diffusion model (e.g., [Kang et al. 2010; Bao et al. 2010; Liu et al. 2011]), the proposed approach captures a wider range of multiple-fluid phenomena, thus allowing efficient production of various interesting and visually realistic multiple-fluid results with fine details throughout the whole simulation.

SPH simulations commonly assume constant particle mass over time and in cases of vaporization this can lead to dramatic increases of the effective volume of particles due to large density drop, which sequentially degrades the simulation resolution. To maintain the simulation resolution, one possible extension to this work would be to introduce an efficient, adaptive refinement strategy to split the SPH particles. Another possible future work is to incorporate the energy equation into the current theoretical framework to simulate relatively less commonly observed energy-related effects, such as extraction in chemistry.

Previous numerical strategies that enforce the incompressibility based on the divergence-free property of the flow field cannot be directly applied to the multiple-fluid simulation, since neither the divergence of the mixture velocity nor that of the phase velocity is zero, even when all phases are incompressible. At the cost of smaller time steps, this limitation can be partially overcome by increasing the gas constant in the equations of state to a higher value, but it will be beneficial to investigate new pressure correction methods enforcing incompressibility of the multiple-fluid simulation at larger time steps, where component-wise incompressibility relations should be taken into account. Another challenge is that incorporating the mixture model with the Predictive-Corrective Incompressible SPH (PCISPH) is not straightforward. PCISPH presumes uniform particle mass and rest density between particles, however, these properties usually vary between particles in the mixture model. This makes the precomputation in the pressure correction of PCISPH impossible, resulting in erroneous values where there is particle deficiency.

APPENDIX

APPENDIX A. DERIVATION OF EQNS. (17–19)

This appendix shows the detailed derivations of Eqs. (17)–(19). The SPH approximation for the divergence operator can be expressed as [Colagrossi and Landrini 2003]

$$(\nabla \cdot \mathbf{A})_i = \sum_j dV_j \mathbf{A}_j \cdot \nabla W_{ij} = \sum_j dV_j (\mathbf{A}_j \pm \mathbf{A}_i) \cdot \nabla W_{ij}. \quad (35)$$

Eq. (18) and Eq. (19) are direct results following Eq. (35).

The derivation of Eq. (17) is more involved. Applying standard SPH formulation to the right-hand side of the following identity

$$(\alpha_k \nabla \cdot \mathbf{u}_m)_i = (\nabla \cdot (\alpha_k \mathbf{u}_m))_i - (\mathbf{u}_m \cdot \nabla \alpha_k)_i \quad (36)$$

leads to

$$(\alpha_k \nabla \cdot \mathbf{u}_m)_i = \sum_j dV_j \alpha_{kj} (\mathbf{u}_{mj} - \mathbf{u}_{mi}) \cdot \nabla W_{ij}. \quad (37)$$

However, applying Eq. (35) to the left-hand side of Eq. (36) yields:

$$(\alpha_k \nabla \cdot \mathbf{u}_m)_i = \sum_j dV_j \alpha_{ki} (\mathbf{u}_{mj} - \mathbf{u}_{mi}) \cdot \nabla W_{ij}. \quad (38)$$

The two preceding equations are almost the same, and directly averaging the right-hand side yields Eq. (17).

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their constructive comments.

REFERENCES

R. Ando and R. Tsuruno. 2010. Vector fluid: A vector graphics depiction of surface flow. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering (NPAR'10).* ACM Press, New York, 129–135.

K. Bao, X. Wu, H. Zhang, and E. Wu. 2010. Volume fraction based miscible and immiscible fluid animation. *Comput. Animat. Virtual Worlds* 21, 3–4, 401–410.

M. Becker and M. Teschner. 2007. Weakly compressible SPH for free surface flows. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'07).* Eurographics Association, 209–217.

L. Boyd and R. Bridson. 2012. Multiflip for energetic two-phase fluid simulation. *ACM Trans. Graph.* 31, 2, 16:1–16:12.

O. Busaryev, T. K. Dey, H. Wang, and Z. Ren. 2012. Animating bubble interactions in a liquid foam. *ACM Trans. Graph.* 31, 4, 63:1–63:8.

P. W. Cleary. 1996. New implementation of viscosity: Tests with couette flows. Tech. rep. DMS-C96/32, CSIRO Division of Math and Statistics.

P. W. Cleary, S. H. Pyo, M. Prakash, and B. K. Koo. 2007. Bubbling and frothing liquids. *ACM Trans. Graph.* 26, 3.

A. Colagrossi and M. Landrini. 2003. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *J. Comput. Phys.* 191, 2, 448–475.

C. T. Crowe, J. D. Schwarzkopf, M. Sommerfeld, and Y. Tsuji. 2011. *Multiphase Flows with Droplets and Particles*. CRC Press, Boca Raton, FL.

F. Dagenais, J. Gagnon, and E. Paquette. 2012. A prediction-correction approach for stable SPH fluid simulation from liquid to rigid. In *Proceedings of the Computer Graphics International.*

M. Desbrun and M. Paule Gascuel. 1996. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Proceedings of the Eurographics Workshop on Computer Animation and Simulation.* Springer, 61–76.

L. M. Gonzalez, J. M. Sanchez, F. Macia, and A. Souto-Iglesias. 2009. Analysis of WCSPH laminar viscosity models. In *Proceedings of the 4th ER-COFTAC SPHERIC Workshop on SPH Applications.*

J. Gregson, M. Krimerman, M. B. Hullin, and W. Heidrich. 2012. Stochastic tomography and its applications in 3D imaging of mixing fluids. *ACM Trans. Graph.* 31, 4, 52:1–52:10 (to appear).

J.-M. Hong and C.-H. Kim. 2005. Discontinuous fluids. *ACM Trans. Graph.* 24, 3, 915–920.

J.-M. Hong, H.-Y. Lee, J.-C. Yoon, and C.-H. Kim. 2008. Bubbles alive. *ACM Trans. Graph.* 27, 3, 48:1–48:4.

X. Hu and N. Adams. 2006. A multi-phase {SPH} method for macroscopic and mesoscopic flows. *J. Comput. Phys.* 213, 2, 844–861.

I. Ihm, B. Kang, and D. Cha. 2004. Animation of reactive gaseous fluids through chemical kinetics. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* Eurographics Association, 203–212.

M. Ihmsen, N. Akinci, G. Akinci, and M. Teschner. 2012. Unified spray, foam and air bubbles for particle-based fluids. *Vis. Comput.* 28, 6–8, 669–677.

B. Kang, Y. Jang, and I. Ihm. 2007. Animation of chemically reactive fluids using a hybrid simulation method. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 199–208.

N. Kang, J. Park, J. Noh, and S. Y. Shin. 2010. A hybrid approach to multiple fluid simulation using volume fractions. *Comput. Graph. Forum* 29, 2, 685–694.

R. Keiser, B. Adams, D. Gasser, P. Bazzi, P. Dutre, and M. Gross. 2005. A unified lagrangian approach to solid-fluid animation. In *Proceedings of the 2nd Eurographics/IEEE/VGTC Conference on Point-Based Graphics (SPBG'05).* Eurographics Association, 125–133.

B. Kim. 2010. Multi-phase fluid simulations using regional level sets. *ACM Trans. Graph.* 29, 6, 175:1–175:8.

B. Kim, Y. Liu, I. Llamas, X. Jiao, and J. Rossignac. 2007. Simulation of bubbles in foam with the volume control method. *ACM Trans. Graph.* 26, 3.

D. Kim, O.-Y. Song, and H.-S. Ko. 2010. A practical simulation of dispersed bubble flow. *ACM Trans. Graph.* 29, 70:1–70:5.

P.-R. Kim, H.-Y. Lee, J.-H. Kim, and C.-H. Kim. 2012. Controlling shapes of air bubbles in a multi-phase fluid simulation. *Vis. Comput.* 28, 6–8, 597–602.

N. I. Kolev. 2005. *Multiphase Flow Dynamics 1: Fundamentals*. Springer.

S. Liu, Q. Liu, and Q. Peng. 2011. Realistic simulation of mixing fluids. *Vis. Comput.* 27, 3, 241–248.

F. Losasso, T. Shinar, A. Selle, and R. Fedkiw. 2006. Multiple interacting liquids. *ACM Trans. Graph.* 25, 3, 812–819.

M. Manninen, V. Taivassalo, and S. Kallio. 1996. On the mixture model for multiphase flow. http://www.vtt.fi/inf/pdf/publications/1996/P288.pdf.

V. Mihalef, B. Unlusu, D. Metaxas, M. Sussman, and M. Y. Hussaini. 2006. Physics based boiling simulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'06).* Eurographics Association, 317–324.

M. K. Misztal, K. Erleben, A. Bargteil, J. Fursund, B. B. Christensen, J. A. Bærentzen, and R. Bridson. 2012. Multiphase flow of immiscible fluids on unstructured moving meshes. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'12).* Eurographics Association, 97–106.

J. Monaghan. 1992. Smoothed particle hydrodynamics. *Ann. Rev. Astron. Astrophys.* 30, 543–574.

J. J. Monaghan and A. Rafiee. 2013. A simple sph algorithm for multi-fluid flow with high density ratios. *Int. J. Numer. Methods Fluids* 71, 5, 537–561.

P. Mullen, A. McKenzie, Y. Tong, and M. Desbrun. 2007. A variational approach to eulerian geometry processing. *ACM Trans. Graph.* 26, 3.

M. Muller, D. Charypar, and M. Gross. 2003. Particle-based fluid simulation for interactive applications. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'03).* Eurographics Association, 154–159.

M. Muller, B. Solenthaler, R. Keiser, and M. Gross. 2005. Particle-based fluid-fluid interaction. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'05).* ACM Press, New York, 237–244.

M. B. Nielsen and O. Osterby. 2013. A two-continua approach to eulerian simulation of water spray. *ACM Trans. Graph.* 32, 4, 67:1–67:10.

J. Park, Y. Kim, D. Wi, N. Kang, S. Y. Shin, and J. Noh. 2008. A unified handling of immiscible and miscible fluids. *Comput. Animat. Virtual Worlds* 19, 3–4, 455–467.

S. Premoze, T. Tasdizen, J. Bigler, A. E. Lefohn, and R. T. Whitaker. 2003. Particle-based simulation of fluids. *Comput. Graph. Forum* 22, 3, 401–410.

B. Solenthaler and R. Pajarola. 2008. Density contrast sph interfaces. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'08).* Eurographics Association, 211–218.

B. Solenthaler, J. Schlafli, and R. Pajarola. 2007. A unified particle model for fluid-solid interactions: Research articles. *Comput. Animat. Virtual Worlds* 18, 1, 69–82.

G. H. Yeoh and J. Tu. 2009. *Computational Techniques for Multiphase Flows*. Butterworth-Heinemann.

H. Zhu, X. Liu, Y. Liu, and E. Wu. 2006. Simulation of miscible binary mixtures based on lattice boltzmann method: Research articles. *Comput. Animat. Virtual Worlds* 17, 3–4, 403–410.

Received September 2013; revised March 2014; accepted April 2014