Fast Multiple-fluid Simulation Using Helmholtz Free Energy

Tao Yang[1], Jian Chang[2], Bo Ren[3], Ming C. Lin[1,4], Jian Jun Zhang[2], and Shi-Min Hu[1]

[1]Tsinghua University [2]Bournemouth University [3]Nankai University [4]University of North Carolina at Chapel Hill

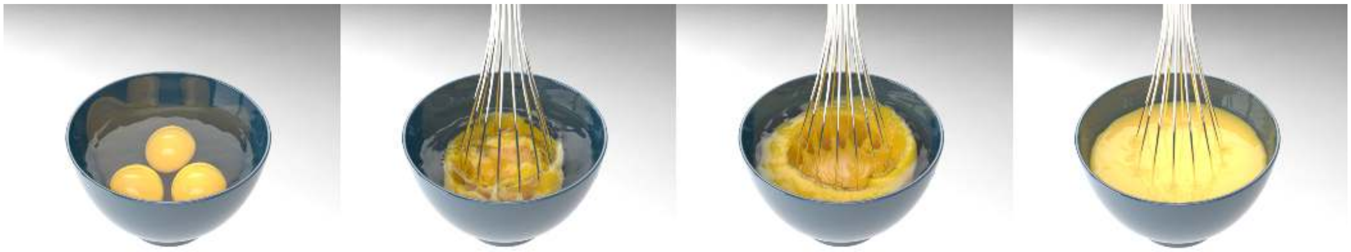

Figure 1: *Egg Mixture. In simulating the stirring of 3 eggs by a blender, the resulting simulation presents visually plausible mixing results. This mixing effect is achieved using our novel "extended mobility" formulation of the Navier-Stokes-Cahn-Hilliard Model.*

Abstract

Multiple-fluid interaction is an interesting and common visual phenomenon we often observe. In this paper, we present an energy-based Lagrangian method that expands the capability of existing multiple-fluid methods to handle various phenomena, such as extraction, partial dissolution, etc. Based on our user-adjusted Helmholtz free energy functions, the simulated fluid evolves from high-energy states to low-energy states, allowing flexible capture of various mixing and unmixing processes. We also extend the original Cahn-Hilliard equation to be better able to simulate complex fluid-fluid interaction and rich visual phenomena such as motion-related mixing and position based pattern. Our approach is easily integrated with existing state-of-the-art smooth particle hydrodynamic (SPH) solvers and can be further implemented on top of the position based dynamics (PBD) method, improving the stability and incompressibility of the fluid during Lagrangian simulation under large time steps. Performance analysis shows that our method is at least 4 times faster than the state-of-the-art multiple-fluid method. Examples are provided to demonstrate the new capability and effectiveness of our approach.

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

Keywords: Position based dynamics, Helmholtz free energy, Multiple-fluid, Navier-Stokes-Cahn-Hilliard model

1 Introduction

Multiple-fluid simulation has received increased attention from the graphics community in the past few years. Multiple-phase or multiple-component fluid simulations are crucial to reproduce realistic visual effects from a wide range of real-world phenomena, which are not adequately modeled by single-phase approaches. Previous works on multiple-fluid simulation can be divided into two main categories focusing on either immiscible fluids or on miscible or dispersed fluids. In the former category, clear interfaces exist between different phases, and many works (e.g., [Hong and Kim 2005; Losasso et al. 2006; Kim 2010; Boyd and Bridson 2012; Misztal et al. 2012; Da et al. 2014]) have obtained visually plausible results by tracking the interfaces. However, in the latter category, there are no clear interfaces between different phases, and the volume fraction method [Müller et al. 2005] has been widely adopted.

Grid-based methods (e.g., level-sets, volume-of-fluid, etc.) [Hong and Kim 2005; Losasso et al. 2006; Kim 2010; Boyd and Bridson 2012; Misztal et al. 2012; Bao et al. 2010; Kang et al. 2010] assume incompressibility with the divergence-free condition and work well on topological transitions, thus achieving plausible results. Previously, the smoothed particle hydrodynamics (SPH) method [Monaghan 1992; Monaghan 1994] gained popularity as a particle based method among the graphics community due to its properties of mass-conservation and easy-to-handle Lagrangian discretization. However, its performance and results are sensitive to particle under-sampling and fluid compressibility, which are the main subjects of more recent investigations [Becker and Teschner 2007; Solenthaler and Pajarola 2009; Ihmsen et al. 2013; Cornelis et al. 2014; Schechter and Bridson 2012; Akinci et al. 2012]. The position based dynamics method [Müller et al. 2007] provides an alternative approach and has shown its potential for various types of physically based simulations.

Previous grid and particle based methods for miscible multiple-fluid simulation mixed phases or components mainly in terms of either concentration differences [Kang et al. 2010; Bao et al. 2010; Liu et al. 2011] or drift velocities [Ren et al. 2014]. These models achieve rich visual results. However, none of these works takes energy (which is essential to many real-world phenomena) into consideration. Physically, materials evolve from high-energy levels to low-energy levels. Non-energy based models cannot easily or intuitively capture the evolution of phenomena (e.g., the extraction phenomenon commonly observed in industrial environments, partial dissolution due to saturation and (oil spill) cleaning) based on this principle. In contrast to previous works, we adopt an energy-based model and integrate the Cahn-Hilliard equation into the multiple-fluid simulation. Our method can be potentially used to simulate a wide range of common daily phenomena related to mixing and unmixing, such as the mixture of oil and water. From an energy point of view, this work provides an alternative approach that expands the capability of multiple-fluid simulation and is flexible and stable. We further extend the original Cahn-Hilliard equation rather than just follow it [Park et al. 2008] to make it capable of simulating more complex and richer visual phenomena such as egg-mixture and gray/colored patterns.

The main contributions of our work are summarized below:

- An energy-based approach to multiple-fluid simulation, extending the capability of multiple-fluid simulation to create new effects such as extraction and partial dissolution.
- An expanded view of the Cahn-Hilliard equation for phase separation, making it possible to simulate complex fluid-fluid interaction and rich visual phenomena by generalizing the formulation and introducing innovative applications of the extended model.
- Practical integrations of the multiple-fluid method into both traditional SPH and PBD frameworks, achieving real-time multiple-fluid simulation, at least 4 times faster than the state-of-the-art multiple-fluid simulation method.

Our paper is organized as follows. First we review related works in Section 2. We then introduce the models and theories in Section 3 and 4. Details of implementation can be found in Section 5. We illustrate our work with examples in Section 6. Finally, limitations and future works are discussed in Section 7.

2 Related Work

Multiple-fluid simulation has been extensively investigated during the last decade. To simulate immiscible fluids, Premǒze et al. [2003] adopted the moving-particle semi-implicit (MPS) approach. Solenthaler and Pajarola [2008] improved the standard SPH method to deal with density contrast more precisely. Amongst Eulerian approaches, the level-set method, the volume-of-fluid (VOF) method, and the finite element method are widely used for interface tracking between phases or components [Hong and Kim 2005; Losasso et al. 2006; Kim 2010; Boyd and Bridson 2012; Misztal et al. 2012].

As for miscible fluids, Müller et al. [2005] introduced the concept of volume fraction to represent the spatial distribution of different phases or components. Kang et al. [2010] and Bao et al. [2010] applied volume fractions to a grid-based solver to achieve desired mixing effects. Liu et al. [2011] and Ren et al. [2014] proved that the concept of volume fraction can be integrated into an SPH solver. Nielsen and Østerby [2013] borrowed the concept of volume fraction to simulate spray in the Eulerian domain. Kang et al. [2010], Bao et al. [2010] and Liu et al. [2011] took cells or particles composed of different phases or components as a whole mixture moving at the same bulk velocity, and therefore mixing due to concentration differences only. To capture the underlying interactions between phases due to flow motions and force distributions, Ren et al. [2014] took the drift velocity of different phases or components into consideration. This approach captures a wider range of multiple-fluid phenomena and can be easily set up. However, their method is not energy-driven and cannot handle many real-world phenomena as illustrated in our work. They also use WCSPH, and minimizing compressibility will in turn limit the time-steps, making it less effective for real-time simulations. Unlike previous works, we adopt an energy-based method that is both computationally efficient and easy to integrate with the PBD method, which more effectively brings incompressibility into the simulation.

Park et al. [2008] also adopted the Cahn-Hilliard equation to simulate multiple fluids using the Lattice Boltzmann method (LBM). They provide a unified approach to handle both immiscible and miscible fluids and show its feasibility with various examples. However, their method simply follows the original Cahn-Hilliard equation without any modification. They simply set the bulk energy, i.e., the Helmholtz free energy as the sum of the pair-wise fluid miscibility which can be a convex function or zero. This formulation substantially limits the targets of multi-fluid simulation to only basic miscible and immiscible fluids. The LBM they adopted incurs significant memory cost and is difficult to modify for simulating phenomena such as chemical reactions. Our work expands the capability of the original model to handle a much wider field of multiple-fluid phenomena and is based on the popular particle frameworks, SPH and PBD.

In fluid simulation, enforcing incompressibility has always been a crucial problem. Standard SPH uses a stiff equation, originating in ideal gas theory. Compared to the standard approach, weakly compressible SPH (WCSPH) [Becker and Teschner 2007] provides a more precise pressure estimate. Both methods require limited time steps. Using an iterative Jacobi-style method to accumulate pressure changes, predictive-corrective incompressible SPH (PCISPH) [Solenthaler and Pajarola 2009] achieves good incompressibility with large time steps. Local poisson SPH (LPSPH) [He et al. 2012] accumulates position and velocity instead of pressure. Ihmsen et al. [2013] introduce the implicit incompressible SPH (I-ISPH) to overcome the incapability of imcompressible SPH (ISPH) (e.g., [Commins and Rudman 1999; Shao and Lo 2003]) in large-scale scenarios. Hybrid methods, like fluid implicit-particle (FLIP) [Brackbill and Ruppel 1986] combine the advantages of grid and particle based methods by using a grid for pressure and particles for advection. Zhu et al. [2005] extend FLIP to simulate incompressible fluids and Raveendran et al. [2011] solve an approximately divergence-free velocity field using a coarse grid. Cornelis et al. [2014] combined IISPH and FLIP to inherit from their advantages.

The position based dynamics framework was first introduced to computer graphics by Müller in [2007]. This method operates on vertices or particles' positions directly via constraints. It accumulates position changes via a Gauss-Seidel iteration to ensure its stability. Macklin and Müller [2013] have shown that PBD can be used to simulate fluids. Borrowing the idea of a density estimator from SPH ([Monaghan 1992; Monaghan 1994]), fluid incompressibility is enforced with a density constraint [Bodin et al. 2012]. Position based fluids (PBF) provides another choice for fluid simulation via a particle system. Compared to plain SPH, it enforces incompressibility achieving similar performances of modern SPH solvers e.g., PCISPH, IISPH. Most recently, Macklin et al. [2014] established a unified dynamic framework, which can be used to simulate a wide range of physical phenomena, including immiscible multi-fluids, in a universal manner. However, their methods cannot be directly applied to miscible fluid simulations.

3 Particle Based Solvers

3.1 The Smoothed Particle Hydrodynamics Method

SPH is an interpolation method which is widely used in particle systems. According to standard SPH, a scalar quantity A_i of i-th particle at position $\boldsymbol{r}_i$ is interpolated by the weighted sum of known quantities from neighboring particles:

$$A_i = \sum_j \frac{m_j}{\rho_j} A_j W(\boldsymbol{r}_{ij}, h), \tag{1}$$

where m_j, ρ_j are mass and density of particle j respectively, $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$, h is the smooth radius, and W is a kernel function. We will use $W_{ij} = W(\boldsymbol{r}_{ij}, h)$ in the rest of this paper. In basic SPH, to simulate fluid with particles, the density ρ_i of i-th particle is given:

$$\rho_i = \sum_j m_j W_{ij}. \tag{2}$$

As in the work by Müller et al. [2003], the Poly6 kernel is used for density estimation and the Spiky kernel for gradient calculation. In our implementation, we use the pressure force and viscous force presented by Monaghan [1992] and Morris et al. [1997]. As for pressure, standard SPH uses a stiff equation:

$$p_i = \kappa(\rho_i - \rho_0), \tag{3}$$

where ρ_0 is the rest density, κ is stiffness constant. Becker and Teschner [2007] instead adopted the Tait equation to achieve better performance:

$$p_i = \frac{\kappa\rho_0}{\gamma}\Big(\big(\frac{\rho_i}{\rho_0}\big)^\gamma - 1\Big), \tag{4}$$

where γ is set to 7. WCSPH can achieve desired results, but small time steps are required for stiffer fluids, which increase the computation cost tremendously. PCISPH, LPSPH and IISPH, etc. instead enforce incompressibility under large time steps with an iterative or implicit approach.

3.2 Position Based Fluids

The PBD method has been adopted in many simulation fields since the work by Müller et al. [2007]. Maclin and Müller [2013] extended this method to fluid simulation. Following the work by Bodin et al. [2012], the density constraint on the i-th particle is defined as follows:

$$C_i = \frac{\rho_i}{\rho_0} - 1. \tag{5}$$

The position of the i-th particle is iteratively adjusted to satisfy the density constraint. The i-th particle's position change in one iteration $\Delta\boldsymbol{r}_i$ lies along the direction of constraint gradient, and the position change follows:

$$\Delta\boldsymbol{r}_i = \nabla C_i \lambda, \tag{6}$$

where λ_i is determined from the constraint values of neighborhood particles. If all particles have the same mass, λ_i is given by

$$\lambda_i = -\frac{C_i}{\sum_j |\nabla C_i|^2}. \tag{7}$$

Macklin et al. [2014] note that it is crucial to take mass into consideration when dealing with different phases or fluid-solid coupling. Then, weighted by the inverse of the mass matrix $\mathbf{M} = diag(m_1, .., m_N)$, the position change is updated by the following:

$$\Delta\boldsymbol{r}_i = \mathbf{M}^{-1} \nabla C_i \lambda, \tag{8}$$

and the updated λ_i is:

$$\lambda_i = -\frac{C_i}{\nabla C_i \mathbf{M}^{-1} \nabla C_i^T}. \tag{9}$$

Table 1: *Definition of symbols*

Symbol	Meaning
i, j	particle index
k, l	phase index
x, y	component of vector
n, N	number of phases, particles
$\varepsilon(c)$	Ginzburg-Laudau free energy
$F, F(c), F(\boldsymbol{c})$	Helmholtz free energy
G	Gibbs n-simplex
C_i	density constrain of particle i
$\Delta\boldsymbol{r}_i$	position change of particle i
c_k, c_l	mass ratio of phase k, l
$\mathbf{M}$	mass matrix
M_i	mobility of particle i
$\boldsymbol{SF}$	reactive stress
$\boldsymbol{c}_i$	mass fraction of particle i
s_k	target mass ratio
s_g	gray value
R, G, B	RGB color value
μ_k	chemical potential of phase k
$\boldsymbol{\mu}_i$	chemical potential vector of particle i
$\beta(\boldsymbol{c})$	Laplacian multiplier
$\boldsymbol{u}_i$	bulk velocity of particle i
ν_k	viscosity of phase k
VBN_k	viscosity blending number of phase k
$\boldsymbol{r}_i, \boldsymbol{r}_j$	position of particle i, j
m_i, ρ_i	mass, density of particle i
$\dot{\gamma}$	strain rate magnitude
$\tilde{D}$	rate-of-deformation tensor
$\tilde{\rho}_i$	adapted density of particle i
ϵ, α, ξ	user-defined coefficients

To overcome particle under-sampling, which is a common problem in SPH simulation, an artificial pressure term [Monaghan 2000] is introduced. This additional term improves the simulation performance while not being purely physical. Undesired damping can also be introduced to the PBD framework too. To deal with this problem, the vorticity confinement method introduced by Fedkiw et al. [2001] and extended by Hong et al. [2008] and Lentine et al. [2011] is adopted. The XSPH method [Monaghan 1989; Schechter and Bridson 2012] is also adopted to provide better particle distributions in the simulation. We refer readers to the literature [Macklin and Müller 2013] for more information on position based fluids.

4 Navier-Stokes-Cahn-Hilliard Model

The Cahn-Hilliard equation [Cahn and Hilliard 1958] comes from mathematical physics and aims to describe the process of phase separation. It has been adopted to model many physical multiple-fluid phenomena (e.g., [Jacqmin 1999]). The Navier-Stokes-Cahn-Hilliard (NSCH) model combines the Cahn-Hilliard equation with the Navier-Stokes equation for fluid simulation problems. It was originally used for two-phase systems and was later extended to ternary or multiple-phase systems. We first briefly describe the NSCH model, then demonstrate how it can be used to enhance multiple-fluid simulation from an energy point of view.

4.1 Governing Equations

Real-world phenomena tend to evolve from high-energy states to low-energy states. In the Cahn-Hilliard equation, the energy law serves as the key source of components' fraction changes. For a certain phase, it gives:

$$\frac{\partial c}{\partial t} + \nabla \cdot (c\boldsymbol{u}) = \nabla \cdot (M \nabla \mu), \tag{10}$$

where c is the order parameter that represents the relative concentration of one of the phases or components. We take this to be the mass ratio of one of the phases or components in a heterogeneous mixture. M is the degenerate mobility, which can be constant or may vary according to the value of c. In our implementation, we define M in a way that does not rely on c. μ is the chemical potential. In the case of two-phase flows, it can be calculated as:

$$\mu = \frac{\partial \varepsilon(c)}{\partial c}, \tag{11}$$

where $\varepsilon(c)$ is the so-called Ginzburg-Landau free energy density:

$$\varepsilon(c) := F(c) + \frac{\epsilon^2}{2} \left|\nabla c\right|^2, \tag{12}$$

where Ω is the domain occupied by the system. $F(c)$ is the Helmholtz free energy. The form of $F(c)$ plays a key role in simulations of various multiple-fluid phenomena, which we will discuss in detail in §4.2. ϵ is associated with diffuse-interfaces and can be vanishingly small describing immiscible fluids. The Ginzburg-Landau free energy is the sum of two distinct parts: bulk part, which is independent of the spatial derivatives of phase quantities; and interfacial part, which depends on the spatial gradient of the phase variable. In L^2 space, combining Eqn (11) and Eqn (12), the formulation of chemical potential can be expressed as:

$$\mu = \frac{dF(c)}{dc} - \epsilon^2 \nabla^2 c. \tag{13}$$

Cahn-Hilliard equation is well-known for its properties of energy dissipation and mass conservation when the natural no-flux boundary conditions are used.

According to Badalassi et al. [2003], Jacqmin [2000], and Kim [2005; 2007; 2012], the equations of NSCH for an n-phase fluid are:

$$\frac{D\boldsymbol{u}}{Dt} = -\frac{1}{\rho}\nabla p + \boldsymbol{g} + \nabla \cdot (\nu \nabla \boldsymbol{u}) + \boldsymbol{SF}, \tag{14}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{15}$$

$$\frac{Dc_k}{Dt} = \nabla \cdot (M \nabla \mu_k), \tag{16}$$

$$\mu_k = \frac{\partial F}{\partial c_k} - \epsilon^2 \nabla^2 c_k + \beta(\boldsymbol{c}), \tag{17}$$

where ρ is the aggregate density, ν is the aggregate viscosity, and $\boldsymbol{u}$ is the bulk velocity. $\boldsymbol{SF}$ is a reactive stress term that captures the interaction forces between different phases. c_k, μ_k are respectively the k-th phase's mass ratio and chemical potential. For a consistent labeling convention, we use subscripts k, l to represent phases and i, j for particles. $\boldsymbol{c}$ is a vector that represents the mass fraction of a single particle. D/Dt is the substantial derivative used to replace the Eulerian expression $\partial/\partial t + \boldsymbol{u} \cdot \nabla$. Admissible states will belong to the Gibbs n-simplex:

$$G := \left\{ \boldsymbol{c} \in R^n \Big| \sum_{i=1}^{n} c_i = 1, c_i \in [0, 1] \right\}. \tag{18}$$

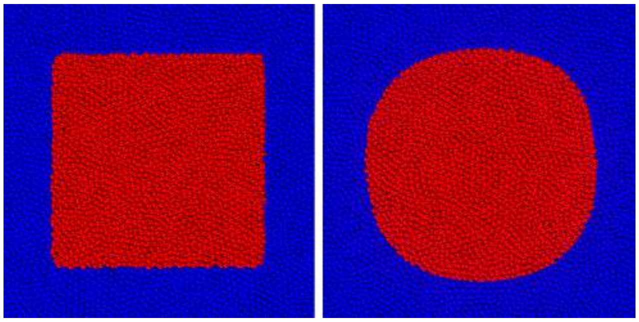

Figure 2: *The effect of surface tension force. The surface tension coefficient is set to be a large value in order to obtain notable results.* ***Left:*** *initial state;* ***Right:*** *final state after driven by* $\boldsymbol{SF}$.

The $\beta(\boldsymbol{c})$ term can be taken according to Garcke et al. [1998] as:

$$\beta(\boldsymbol{c}) = -\frac{1}{n} \sum_k \frac{\partial F}{\partial c_k}. \tag{19}$$

As M does not rely on c, it is not difficult to verify that the existence of $\beta(\boldsymbol{c})$ makes the summed change of c_k zero, ensuring physical meaningfulness during the fraction change process. However, in our experiments, Eqn (18) cannot always be satisfied due to numerical errors and discretization. We adopt a correction process according to the work by Ren et al. [2014].

The reactive stress term can be set according to Kim [2009] as:

$$\boldsymbol{SF} = \sum_{k,l(l>k)} \frac{\sigma_{k,l}}{2} \Big(\boldsymbol{sf}(c_k) + \boldsymbol{sf}(c_l) \Big) \chi(c_k, c_l), \tag{20}$$

where $\sigma_{k,l}$ is the surface tension coefficient between phase k and l. $\boldsymbol{sf}(c_k) = -6\sqrt{2}\epsilon \nabla \cdot \left(\frac{\nabla c_k}{|\nabla c_k|}\right) |\nabla c_k| \nabla c_k$ and $\chi(c_k, c_l) = 5 c_k c_l$. Eqn (20) is known as the generalized continuous surface tension force proposed to avoid the solvability problem imposed by the over-determined system [Boyer and Lapuerta 2006]. The effect of this surface tension term is illustrated in Fig.2, where the tension force minimizes the surface areas and slows the diffusion between phases (resulting sharper interfaces).

4.2 Helmholtz Free Energy and Gibbs Triangle

According to Eqn (10), the state of a multiple-phase fluid will evolve in the direction of decreasing total chemical potential, which in turn relies on the Helmholtz free energy. The Helmholtz free energy is an internal energy of the material that determines the final stable state. By properly setting different forms of Helmholtz free energy equations, the NSCH model can produce various energy-driven visual effects. For cases containing less than 3 phases, proper energy functions can be designed with the help of energy plots or the so-called Gibbs triangle. We will demonstrate such designation method in this section.

We first begin with two-phase fluids. In this case the energy plot can be represented by an energy-curve plot relying on the mass fraction of one of the two phases, which is between 0 and 1. To capture rich visual results in a two-phase system, we use the following free energy equation unless explicitly mentioned:

$$F(\boldsymbol{c}) = \alpha (c_1 - s_1)^2 (c_2 - s_2)^2, \tag{21}$$

where α is a user-defined diffusional coefficient. $s_1, s_2 \in [0, 1]$. Generally, there are two minima i.e., $(s_1, 1 - s_1), (1 - s_2, s_2)$ in

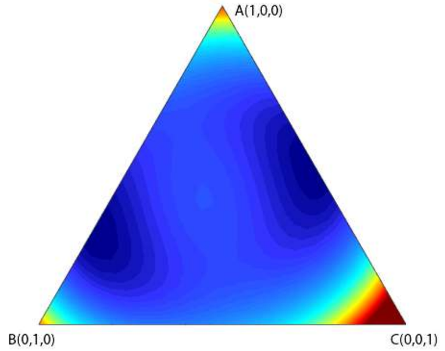

Figure 3: *Contour plot of the Helmholtz free energy on the Gibbs triangle. Each point in this triangle relates to a state in which the local fluid contains the corresponding fraction of each of the three phases. Values of Helmholtz free energy in each state are color-plotted (red for high energy and blue for low). The two minima in this plot correspond to the final stable states.*

this energy plot. By setting s_1, s_2 to different values, this energy function can be adopted to simulate different two-phase phenomena. For example, to generate the process of phase separation, s_1, s_2 are set to 0, while they are set to 0.5 in order to capture the opposite phenomenon of complete dissolution. The liquid partial dissolution (Fig.7) happens when two miscible fluids cannot be arbitrarily mixed and have fully-saturated fraction limits within each other. This will lead to the formation of two separated fully-saturated mixtures with different phases as solvent. In our experiments, this phenomenon can be captured by placing two local minima in the energy plot at the corresponding saturation points, i.e., simply setting $s_1 = s_2 = 0.4$ in our example.

For three-phase fluids the energy plot becomes a Gibbs triangle. In Fig.3, each vertex represents the state in which the local fluid contains only a single phase, and each point in the triangle represents a mixed state with corresponding fractions of each of the three phases. The Helmholtz energy function defined on this triangle is color-plotted. In this case we use an energy equation describing chemical extraction phenomena as follows:

$$F(\boldsymbol{c}) = \alpha\Big(c_1^2(c_1-0.3)^2+(c_2-0.7)^2(c_2-0.5)^2+c_3^2(c_3-0.5)^2\Big). \tag{22}$$

The two minima of $F(\boldsymbol{c})$ are $(0.3, 0.7, 0), (0, 0.5, 0.5)$. As can be observed in Fig.3, the state with phase A dissolved in phase C has a lower energy minimum than the state with phase A dissolved in phase B, so when C is added to the mixture of A and B, it will extract A from B, resulting in a mixture of A and C separated from a nearly-pure phase B.

Some important principles should be followed when designing energy functions: (1) The partial derivative of the chemical potential with respect to a certain phase should be related to the mass ratio of this phase only, otherwise mass conservation can be broken and unexpected phases would appear at undesired places. (2) The minima of energy functions should locate within the area of Gibbs triangle to ensure the stability of the final states.

By placing minima at different positions in the Gibbs triangle and adjusting their energy values, the interactive behavior of the three phases can be intuitively designed for flexible simulation.

For cases containing more than 4 phases, the basic idea of designing the Helmholtz free energy function remains unchanged. For example, we can use a Gibbs tetrahedron instead of a Gibbs triangle when there are 4 phases. When there are many phases of the simulation, it can be beneficial to adopt the phase labeling method (PLM) originated from the traditional particle labeling method, separating the phases into several "interacting groups". The number of concerned phases in a single group can be much smaller, making the energy functions easier to set up.

5 Implementation

We introduced the ideas of particle based simulations (SPH and PBF) and NSCH respectively in Sections 3 and 4. In this section, we provide practical integration implementations. Inspired by Macklin and Müller [2013] and Ren et al. [2014], we update every particle's mass ratio first in each time step and then apply the effects of the extra reactive stress along with the viscosity. Our PBD based algorithm is outlined in Algorithm 1.

Algorithm 1 Simulation Loop

```
for all particles $i$ do
    apply external forces to predict position
end for
for all particles $i$ do
    find neighboring particles
end for
for all fluid particles $i$ do
    for each phase $k$ do
        apply Cahn-Hilliard equation
    end for
end for
while $iter<solverIterations$ do
    accumulate position changes via density constraint
end while
for all particles $i$ do
    update velocity
    apply vorticity confinement, viscosity and reactive stress
    update position
end for
```

Most of our implementations are based on Algorithm 1 in order to take advantage of enforcing incompressibility and stability with large time steps. The accuracy of Algorithm 1 will be discussed in §5.1.

The NSCH model can also be readily integrated with standard SPH solvers. In our experiments, we take this energy-driven Cahn-Hilliard equation as a phase diffusion process and apply it before the main process of SPH in every time step. Ren et al. [2014] add a pressure correction step after the update of volume fraction. It brings more physical basis during the volume fraction change process and can potentially provide better incompressibility. Since we do not consider phase-related drift velocity and pressure, and as a result Eqn (15) can be applied to handle incompressibility, it is not necessary to include such a correction step for achieving plausible visual simulation.

5.1 SPH Formulation

Eqn (2) is commonly used in particle systems for density calculation of particles. Solenthaler and Pajarola [2008] provided a modification to simulate immiscible multiple-fluids with high density ratio. Their method uses the adapted density:

$$\tilde{\rho}_i = m_i \sum_j W_{ij}. \tag{23}$$

Either equation can be used to estimate density. In our experiments, Eqn (23) gives better results, because of its ability for per-particle self-adaptation, especially in the case of high density ratios.

The gradient or Laplacian of a function defined on particles is calculated according to Monaghan [1992], Morris et al. [1997], Müller et al. [2003], and Ihmsen et al. [2014]. For stability, we always use symmetric formulations. For each particle:

$$\nabla \boldsymbol{\mu}_i = \rho_i \sum_j m_j \Big(\frac{\boldsymbol{\mu}_i}{\rho_i^2} + \frac{\boldsymbol{\mu}_j}{\rho_j^2} \Big) \nabla W_{ij}, \tag{24}$$

$$\nabla \cdot (M_i \nabla \boldsymbol{\mu}_i) = \sum_j \frac{m_j}{\rho_j} (M_i + M_j) \boldsymbol{\mu}_{ij} \frac{\boldsymbol{r}_{ij} \cdot \nabla W_{ij}}{r_{ij}^2 + \eta^2}, \tag{25}$$

$$\nabla^2 \boldsymbol{c}_i = 2 \sum_j \frac{m_j}{\rho_j} \boldsymbol{c}_{ij} \frac{\boldsymbol{r}_{ij} \cdot \nabla W_{ij}}{r_{ij}^2 + \eta^2}, \tag{26}$$

where $\boldsymbol{\mu}_i$, $\boldsymbol{c}_i$ are n-component vectors for the i-th particle for an n-phase fluid system. $\boldsymbol{\mu}_{ij} = \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$. $\boldsymbol{c}_{ij} = \boldsymbol{c}_i - \boldsymbol{c}_j$. M_i is the mobility of i-th particle. η is a small positive scalar and is set to $0.1h$ (e.g., [Ihmsen et al. 2014]). Other spatial derivatives in NSCH equations can be found in a similar way to the equations above.

There exists a fourth-order spatial derivative in Cahn-Hilliard equation. Due to the sensitivity of high order derivatives, stable simulation cannot be achieved if using the fourth-order derivative directly, especially with kernel function in particle systems. We approximate this fourth-order derivative with a bi-Laplacian formulation as in previous works. The accuracy of this formulation in SPH has been well explored (e.g., [Hirschler et al. 2014]).

This fourth-order derivative in space also introduces a time step restriction for stability because of the large difference stencils. However, in our experiments, even large time steps under PBD framework are proper. PBD uses a small number of iterations to accumulate position changes and is unlikely to converge in every single step. This property slows the diffusion process. External forces such as shaking the bottles are adopted to increase the probability of particle collision for the purpose of diffusing concentrations.

5.2 Variable Density and Viscosity

We use the harmonic average for the aggregate density as follows:

$$\rho = \Big(\sum_k \frac{c_k}{\rho_k} \Big)^{-1}, \tag{27}$$

where k indexes the set of phases. It is not difficult to verify that the above average using mass fraction values is equivalent to linear average using the volume fraction values.

The aggregate viscosity can be handled in a similar way as in the original NSCH model. However, in our experiments, this method is a simple but imperfect choice. To estimate the viscosity of a mixture more precisely, we choose the Refutas method [Maples 2000] which has been commonly used in engineering calculations instead. To calculate the viscosity of a mixture, a viscosity blending number (VBN) of each phase should first be calculated as follows:

$$VBN_k = 14.534 \times \log\big(\log(\nu_k + 0.8)\big) + 10.975, \tag{28}$$

then the VBN of the mixture is:

$$VBN = \sum_k c_k VBN_k, \tag{29}$$

finally, the viscosity of the mixture is estimated below:

$$\nu = \exp\Big(\exp(\frac{VBN - 10.975}{14.534}) \Big) - 0.8. \tag{30}$$

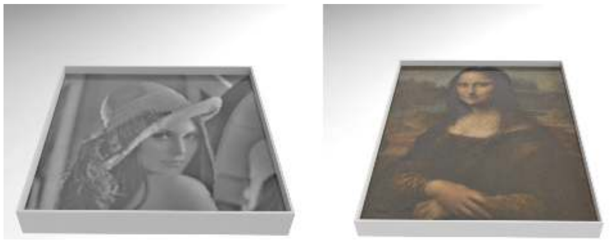

Figure 4: *Position based patterns. We extend the free energy functions to be associated with position, achieving various interesting patterns with our multiple-fluid solver.* ***Left:*** *gray image;* ***Right:*** *colored image.*

5.3 Extended Mobility

In the original NSCH model, degenerate mobility is only associated with mass fraction and in most cases is constant. We extend this mobility to expand the simulation range of NSCH models to capture more complex fluid-fluid interaction, such as handling certain motion-related mixing phenomena. Such phenomena usually involve several phases, which stay separated at rest, but become mixed under kinetic interaction. A commonly observed case is whisking eggs in a bowl. To capture these phenomena, extended mobility is a feasible approach.

Since more obvious mixing phenomena are observed with larger kinetic differences, we define the extended mobility in a continuous way:

$$M = \frac{1}{1 + e^{-\xi(\dot{\gamma} - \dot{\gamma}_0)}}, \tag{31}$$

where $\dot{\gamma}$ is the strain rate magnitude. $\dot{\gamma}_0$ indicates the threshold. ξ relates to the steepness of M.

The strain rate magnitude is defined as:

$$\dot{\gamma} = \sqrt{2\tilde{D} : \tilde{D}}, \tag{32}$$

where $\tilde{D}$ is the rate-of-deformation tensor, which is defined below:

$$\tilde{D} = \frac{1}{2} \Big\{ \frac{\partial u_x}{\partial r_y} + \frac{\partial u_y}{\partial r_x} \Big\}, \tag{33}$$

where x, y indicate the components of vector. Following the work by Monaghan [1992], the (x, y)'s component of $\tilde{D}$ of i-th particle can be computed in particle systems using standard SPH style estimator:

$$(\tilde{D}_{x,y})_i = \frac{1}{2} \sum_j \frac{m_j}{\rho_j} \Big((\boldsymbol{u}_{ij})_x \nabla_y W_{ij} + (\boldsymbol{u}_{ij})_y \nabla_x W_{ij} \Big), \tag{34}$$

where $(\boldsymbol{u}_{ij})_x, \nabla_x W_{ij}$ are the x's components of $\boldsymbol{u}_{ij}$ and ∇W_{ij}.

The extended mobility in Eqn (31) is related to the shear rate which is rotationally invariant. In practice, we set the threshold in Eqn (31) to a sufficiently large value, minimizing the value of mobility when the magnitude of strain rate is (close to) zero to prevent unexpected mixing under slight disturbance. There also exists a numerical upper bound in Eqn (31), which is critical to ensure numerical stability especially when the shear rate is too high. Desired visual results are achieved when simulating motion-related mixing phenomena such as egg-mixing shown in Fig.1.

5.4 Position Based Pattern

Using our energy driven multiple-fluid solver, the final stable states can be intuitively designed by placing local minima of free energy functions at different positions in the energy plot or the Gibbs triangle for two- or three-phase fluids respectively, as discussed in §4.2. Helmhotz free energy functions are typically designed with pre-settled final states which are user-defined constants. We extend this design formulation to generate interesting visual patterns.

The key is to extend the traditional Helmholtz free energy to be associated with position-controlled potential energy. In this way, any desired patterns can be achieved. For example, gray images can be straightforwardly considered to be mixtures of two phases which are pure black and pure white. To simplify the numerical analysis, we change Eqn (21) as follows:

$$F(\boldsymbol{c}) = \alpha\big((c_1 - s_1)^2 + (c_2 - s_2)^2\big). \tag{35}$$

To generate a gray image pattern as shown in Fig.4 using the multiple-fluid system, particles play the role of sampling points and their stable states vary in space. For instance, in 2D, the stable state of a particle at the position of (r_x, r_y) should relate to the gray value ($s_g, s_g \in [0,1]$) at the mapping coordinate (R_x, R_y) in the gray image. We set $s_1 = s_g, s_2 = 1 - s_g$.

Since the stable states vary in space, this would bring some numerical effects to the NSCH equations. Substituting Eqn (35) to Eqn (17), then Eqn (16) gives:

$$\frac{Dc_k}{Dt} = \nabla \cdot (M\nabla\tilde{\mu}_k) - 2\alpha\nabla \cdot (M\nabla s_k), k \in \{1,2\}, \tag{36}$$

where $\tilde{\mu}_k = 2\alpha c_k - \epsilon^2\nabla^2 c_k + \beta(\boldsymbol{c})$, which is irrelevant to s_k. If s_k is constant, the second term of the right side of Eqn (36) vanishes. If s_k varies in space as discussed above, this term matters. The effect of this non-vanishing term in this section depends on how s_k is set in space and it can accelerate or slow the diffusion process compared to previous vanishing cases. Since $\sum_k s_k = 1$, this non-vanishing term accelerates the diffusion of some phases but slows that of others. All in all, it steers phases to the final result resting in desired patterns.

As shown in Fig.4, our extended method can also be used to obtain colored patterns rather than just gray ones. To obtain colored patterns, we assume there exist 4 different phases. The RGB color (R, G, B) of each pixel in the image is used to represent the target mass fraction:

$$s_1 = \frac{R}{3}, s_2 = \frac{G}{3}, s_3 = \frac{B}{3}, s_4 = 1 - \frac{R+G+B}{3}, \tag{37}$$

where $R, G, B \in [0,1]$. With a generalization of Eqn (35), the energy function can be chosen as:

$$F(\boldsymbol{c}) = \alpha\sum_k (c_k - s_k)^2, k \in \{1,2,3\}. \tag{38}$$

We only use c_1, c_2, c_3 to represent colors. The numerical effects are just the same as discussed previously.

Position based patterns are commonly observed phenomena e.g., water eutrophication and electrolyte solution. We provide a flexible and reasonable approach to artistically control their behaviors.

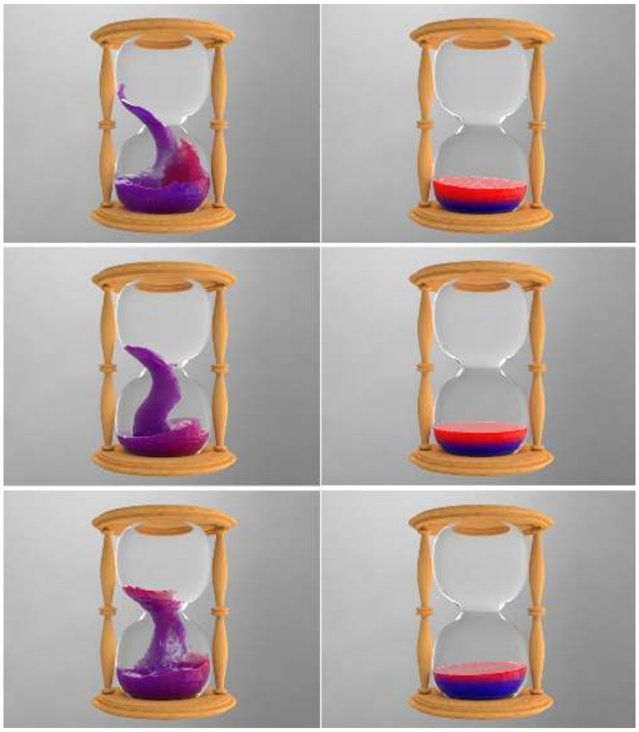

Figure 5: *Performance Comparison. From left to right, two liquids (Red and Blue) mix and then unmix in a hourglass. All three implements achieve similar results.* ***Top:*** *PBD based NSCH;* ***Middle:*** *WCSPH based NSCH;* ***Bottom:*** *[Ren et al. 2014].*

5.5 Chemical Reaction

Visualization of a chemical reaction can also be a practical extension of our method. Owing to the convenience of the mass fraction representation, this phenomenon can be simply handled by adding an in-particle re-balance step in every time step [Ren et al. 2014]. Since Helmholtz free energy is associated with phase variables, it matters how we label the newly generated phases. In our experiments, phases that take part in a chemical reaction as reactants are considered to be within the same group, and each newly generated phase is considered to be within a separate group. This labeling simplifies the designing work for energy functions and prevents the unexpected diffusions between reactants and products.

5.6 Boundary Condition

We use particles to represent anomalous boundaries. Thus, it is critical to prevent fluid particles penetrating and clustering. Akinci et al. [2012] provided a versatile rigid-fluid coupling approach using per-particle correction. We adopt this method in our SPH based implementation. However, as discussed in the work by Macklin et al. [2014], there are no explicit pressure forces existing in the original PBF. To solve this problem, we treat the boundary-fluid pressure and friction forces computed according to Akinci et al. [2012] as part of external forces in our implementations. Typical penetrating and clustering problems can be avoided in our experiments as shown in all examples.

Boundary particles do not take part in the phase diffusion process. No boundary particles are considered in line 9 of Alg.1 where Eqn.(16) is applied. This no-flux boundary conserves both the total and phases' masses.

6 Performance and Results

6.1 Performance Analysis

We compare our results mainly to the state-of-the-art work [Ren et al. 2014], which takes drift velocity of different phases or components into account and achieves various real-world multiple-fluid phenomena.

Implementation & Visual Comparison: Our method can be implemented with both SPH and PBD frameworks. Although the work by Ren et al. [2014] can also be readily integrated into SPH framework, we found the integration of their method with PBD rather complex and not straight-forward. To ensure the fairness of comparison, we set up similar mixing/unmixing simulations and then simulate them using the WCSPH based method by Ren et al. [2014] and our WCSPH and PBD based approaches respectively. The simulation results can be found in Fig.5. All these implementations achieve similar results. The PBD-based implementation presents its abilities on stability and enforcing impressibility under large time steps.

Speed Comparison: We implement our algorithm with CUDA and run on an NVIDIA GeForce GTX 980 GPU. Performance for these examples can be found in Table 2. The particle numbers given include both fluid and boundary particles. Overall, for a simulation with an equal number of particles and phases, we observe about one order of magnitude performance improvement over the work by Ren et al. [2014] with PBD framework and 4 times speed-up with SPH framework.

6.2 Results

Fluid Extraction: Example 1 (Fig.6) shows that our model can be used to simulate the energy-driven process of extraction. The energy function is set to Eqn (22). In the beginning, a blue liquid and a mixture of red and green liquids (shown as a yellow liquid) are initialized in a separating funnel. The green liquid has greater solubility in the blue liquid than in the red one. Thus, after vigorous mixing, the blue liquid extracts the green liquid out of the red liquid, resulting in a layer of red liquid on top of a cyan mixture.

Partial Dissolution: Example 2 (Fig.7) is an example of partial dissolution from chemistry. The energy function is set to Eqn (21) where $s_1 = s_2 = 0.4$. From left to right, the same volume of blue liquid is used for each of the 3 cases while the volume of red liquid doubles each time. The results show that when the volume of one of the liquids, either red or blue, is greater than the other, the two liquids generate a homogeneous one-phase mixture at the end. If both liquids are saturated with the other, they generate two separate fully-saturated mixtures.

Egg Mixture: Example 3 (Fig.1) demonstrates that the capability of the NSCH model can be expanded via extended mobility using an everyday example of mixing egg white and egg yolk. Unlike other examples where the mobilities are constant, we use the extended mobility discussed in §5.3. And the energy function is set to Eqn (21) where $s_1 = s_2 = 0.5$. The mixture of egg white and egg yolk at any mass fraction remains stable when it is still. When slightly shaking or rotating the bowl as a whole, there is no obvious mixing. However, when stirring the mixture with a whisk, the local shear rate becomes significant, resulting in strong mixing.

Multiple Liquids & Chemical Reaction: Example 4 (Fig.8) demonstrates the ability of our approach to deal with simulations with more than 3 phases as well as to simulate the phenomena of chemical reactions. There are 5 dambreaks at the beginning and 7 phases in total. We adopt the phase labeling method to divide them into several "interacting groups". From left to right, the first and second dambreaks contain 3 phases which consist of same phase group and present the process of extraction as shown in Example 1 (Fig.6). The middle two simulations of dambreaks preform the phenomena of chemical reactions (i.e., the reaction is A + B = 2C), producing a new green phase. The two reactants are in the same group while the product is in another group. The last simulation of dambreak is immiscible with the other 6 phases, in another single group.

Table 2: *Performance on Benchmark Examples*

Example	solver	phases	particles	steps/frame	msec/frame
Extraction	PBD	3	110k	4	92.5
Partial	PBD	2	112k/131k/181k	4	78.4/88.9/114.2
Egg	PBD	2	75k	3	69.3
Bunny	PBD	7	260k	3	115.4
Lena	PBD	2	177k	4	66.7
Mona Lisa	PBD	4	309k	4	120.9
Hourglass	PBD	2	134k	4	68.9
	WCSPH	2	134k	16	133.3
	[Ren et al. 2014]	2	132k	16	566.8

There exist some user-specified parameters (i.e., ϵ, α, ξ, $\dot{\gamma}_0$) that provide more flexibility for artistic design. ξ and $\dot{\gamma}_0$ are only used in Example 3 (Fig.1), and we set $\epsilon = 0.01$ in all our examples. The simulation can remain stable even if these parameters vary, due to the unconditional stability of PBD.

7 Conclusion and Discussion

In this paper, we have presented an energy based method to simulate multiple-fluid phenomena. The results show that our method produces visually plausible results while expanding the capability of existing models. Examples like fluid extraction and particle dissolution are beyond the capability of previous methods that we are aware of. Real-world phenomena like modeling and simulation of mixtures of water and oil, visual simulation of mud sliding, sedimentation processes, and avalanches due to natural disasters are highly correlated with energy. These are real-world scenarios in which our work could be useful, offering several further possible extensions. Our method can be readily integrated with state-of-the-art SPH solvers, showing its potential for various applications. Offering practical integration with PBD framework, this approach is more flexible and stable with larger time steps than traditional SPH-based methods.

As discussed in §5.1, our method suffers from slow convergence, which is the result of the bi-Laplacian formulation, discretization, and PBD. Although our method can ensure stable simulations with large time steps, special attention is required to avoid the occurrence of negative values of c as much as possible, i.e., the time steps and the Helmhotz free energy should be chosen in a way to ensure that the changes of c during a single step do not exceed the values of c. In our experiments, negative values of c do exist due to discretization, numerical errors, and casually chosen parameters. The simulations can stay stable with our correction process as mentioned in §4.1, alhtrough they will lose the capability of conserving the masses of phases.

Ignoring the drift velocity simplifies the implementation, analysis, and control, enabling better performance, e.g., incompressibility as discussed in §5. However, without the drift velocity, our method inevitably loses some capabilities. For instance, in the process of unmixing, different phases separate at every corner of the simulation domain simultaneously and then form layers driven by gravity

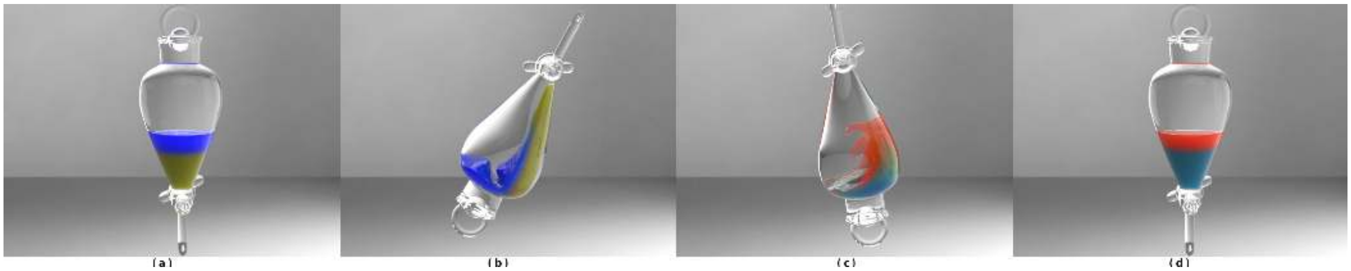

Figure 6: *Fluid extraction. From left to right,(a) At the beginning, the blue liquid and yellow mixture are put in a separating funnel. (b) The funnel is inverted. (c) Shaking vigorously mixes the fluids. (d) Turning the funnel upright results in a clear interface between a red liquid and a cyan mixture. Due to greater solubility of the green liquid within the blue liquid, the blue liquid extracts the green liquid from the red liquid. The whole process is driven by Helmholtz free energy and conserves volume.*

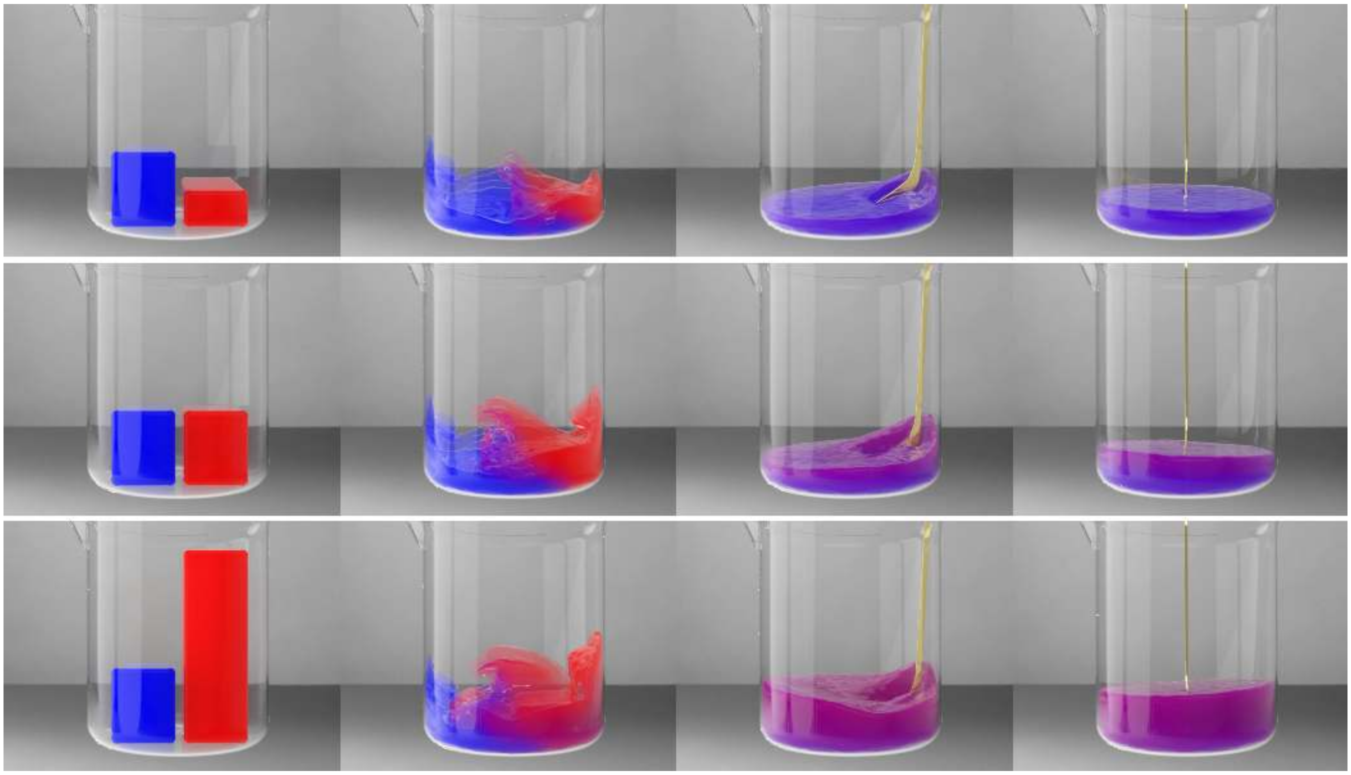

Figure 7: *Partial dissolution. From top to bottom, three different cases are initialized with the same volume of green liquid and different volumes of red liquid. From left to right, all three cases illustrate the process of dambreak and mixing with a yellow stick. The final states in the three cases are very different because of fully-saturated fraction limits. In the first case, there is more blue liquid, and the red liquid dissolves into the blue liquid completely. In the second case, both the red and blue liquids are comparable in volume, and each partially dissolves in the other. In the third case, there is more red liquid, and the blue liquid completely dissolves in the red liquid. Both first and third cases result in homogeneous mixtures while the second produces two fully saturated mixtures.*

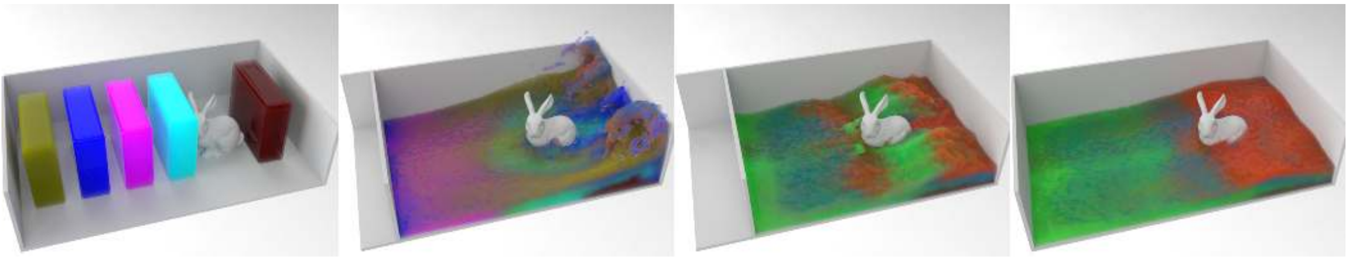

Figure 8: *Multiple-fluid simulation. Our model can easily be adopted to simulate fluids with an arbitrary number of phases as well as handle phenomena of chemical reactions. From left to right in the first image: The first and second simulations of dambreaks with 3 phases are in a group and present the process of extraction, the middle two simulations of dambreaks achieve the visual effects of chemical reaction and generate a new green phase, and the last dambreak is independent with other phases.*

and buoyancy with our method. This whole process differs from real-world layering, e.g., sedimentation. Furthermore, with the drift velocity, different phases move at different velocities in the mixture. These discrepant motions play a key role in the process of motion-induced mixing, resulting in relative mass migration. However, this motion-induced mixing cannot be well addressed without the drift velocity, and this points out the essential reason for the slow diffusion with our method apart from the slow convergence as mentioned above. We hope to take the drift velocity into consideration in the future.

The aggregate viscosity is crucial for real-world simulation. We have introduced the Refutas method, which is simple and plausible. However, this issue is much more complex than discussed, i.e., the viscosities of non-Newtonian fluids are mostly shear-dependent, however we haven't taken this into account. It deserves more efforts and further research.

The PBD framework enables the use of large time steps which are critical for real-time simulation. Our method can offer simulation in real time but the results require computationally expensive volume rendering. We hope to develop a real-time multiple-fluid rendering technique in the future.

Nielsen and Østerby [2013] simulated spray in the Eulerian domain using a two-continua approach, which treats mist in a miscible fluid kind of way. It would be practical to combine their work with ours.

We note that temperature plays important roles in multiple-fluid simulation. We aim to take into consideration the further expansion of the simulation range for multiple-fluid in the future.

Acknowledgements

We would like to thank Sarah George-Waterfield and Ehtzaz Chaudhry for helping to proofread the final version of this paper. We also thank the anonymous reviewers for their constructive comments. This work is supported in part by the Ministry of Education of the People's Republic of China, the EU project Dr. Inventor (FP7-ICT-2013.8.1 611383), and the U.S. National Science Foundation. The earlier research leading to this project also received funding from the People Programme (Marie Curie Actions) of the European Unions Seventh Framework Programme F-P7 (2007-2013) under REA grant agreement No.612627-“AniNex”, and University of North Carolina Arts & Sciences Foundation.

References

AKINCI, N., IHMSEN, M., AKINCI, G., SOLENTHALER, B., AND TESCHNER, M. 2012. Versatile Rigid-Fluid Coupling for Incompressible SPH. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012) 31*, 4, 62:1–62:8.

BADALASSI, V. E., CENICEROS, H. D., AND BANERJEE, S. 2003. Computation of multiphase systems with phase field models. *Journal of Computational Physics 190*, 2, 371–397.

BAO, K., WU, X., ZHANG, H., AND WU, E. 2010. Volume fraction based miscible and immiscible fluid animation. *Computer Animation and Virtual Worlds 21*, 3-4 (May), 401–410.

BECKER, M., AND TESCHNER, M. 2007. Weakly compressible SPH for free surface flows. *In Proceedings of SCA '07*, 209–217.

BODIN, K., LACOURSIÉRE, C., AND SERVIN, M. 2012. Constraint Fluids. *IEEE Transactions on Visualization and Computer Graphics 18*, 3 (Mar.), 516–526.

BOYD, L., AND BRIDSON, R. 2012. MultiFLIP for Energetic Two-Phase Fluid Simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012) 31*, 2 (Apr.), 16:1–16:12.

BOYER, F., AND LAPUERTA, C. 2006. Study of a three component Cahn-Hilliard flow model. *ESAIM:M2AN 40*, 4, 653–687.

BRACKBILL, J. U., AND RUPPEL, H. M. 1986. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational Physics 65*, 2, 314–343.

CAHN, J. W., AND HILLIARD, J. E. 1958. Free Energy of a Nonuniform System. I. Interfacial Free Energy. *Journal of Chemical Physics 28*, 2, 258–267.

COMMINS, S., AND RUDMAN, M. 1999. An SPH Projection Method. *Journal of Computational Physics 152*, 2, 584–607.

CORNELIS, J., IHMSEN, M., PEER, A., AND TESCHNER, M. 2014. IISPH-FLIP for incompressible fluids. *Computer Graphics Forum (Proceedings of Eurographics 2014) 33*, 2, 255–262.

DA, F., BATTY, C., AND GRINSPUN, E. 2014. Multimaterial Mesh-Based Surface Tracking. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014) 33*, 4, 112:1–112:11.

FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual Simulation of Smoke. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2001)*, 15–22.

GARCKE, H., NESTLER, B., AND STOTH, B. 1998. On Anisotropic Order Parameter Models for Multi-Phase Systems and Their Sharp Interface Limits. *Physica D: Nonlinear Phenomena 115*, 1-2, 87–108.

HE, X., LIU, N., WANG, H., AND WANG, G. 2012. Local Poisson SPH For Viscous Incompressible Fluids. *Computer Graphics Forum 31*, 6, 1948–1958.

HIRSCHLER, M., HUBER, M., SÄCKEL, W., KUNZ, P., AND NIEKEN, U. 2014. An Application of the Cahn-Hilliard Approach to Smoothed Particle Hydrodynamics. *Mathematical Problems in Engineering 2014(694894):10.*

HONG, J.-M., AND KIM, C.-H. 2005. Discontinuous Fluids. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2005) 24*, 3 (July), 915–920.

HONG, J.-M., LEE, H.-Y., YOON, J.-C., AND KIM, C.-H. 2008. Bubbles Alive. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008) 27*, 3 (Aug.), 48:1–48:4.

IHMSEN, M., CORNELIS, J., SOLENTHALER, B., HORVATH, C., AND TESCHNER, M. 2013. Implict Incompressible SPH. *IEEE Transactions on Visualization and Computer Graphics 20*, 03 (July), 426–435.

IHMSEN, M., ORTHMANN, J., SOLENTHALER, B., KOLB, A., AND TESCHNER, M. 2014. SPH Fluids in Computer Graphics. In *Eurographics 2014 - State of the Art Reports*, The Eurographics Association, 21–42.

JACQMIN, D. 1999. Calculation of Two-Phase Navier-Stokes Flows Using Phase-Field Modeling. *Journal of Computational Physics 155*, 1, 96–127.

JACQMIN, D. 2000. Contact-line dynamics of a diffuse fluid interface. *Journal of Fluid Mechanics 402*, 01, 57–88.

KANG, N., PARK, J., NOH, J., AND SHIN, S. Y. 2010. A Hybrid Approach to Multiple Fluid Simulation using Volume Fractions. *Computer Graphics Forum 29*, 2, 685–694.

Kim, J. 2005. A continuous surface tension force formulation for diffuse-interface model. *Journal of Computational Physics 204*, 2, 784–804.

Kim, J. 2007. A numerical method for the Cahn-Hilliard equation with a variable mobility. *Communications in Nonlinear Science and Numerical Simulation 12*, 8, 1560–1571.

Kim, J. 2009. A generalized continuous surface tension force formulation for phase-field models for immiscible multi-component fluid flows. *Computer Methods in Applied Mechanics and Engineering 198*, 37-40, 3105–3112.

Kim, B. 2010. Multi-Phase Fluid Simulations Using Regional Level Sets. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2010) 29*, 6 (Dec.), 175:1–175:8.

Kim, J. 2012. Phase-Field Models for Multi-Component Fluid Flows. *Communications in Computational Physics 12*, 3, 613–661.

Lentine, M., Aanjaneya, M., and Fedkiw, R. 2011. Mass and Momentum Conservation for Fluid Simulation. *In Proceedings of SCA '11*, 91–100.

Liu, S., Liu, Q., and Peng, Q. 2011. Realistic simulation of mixing fluids. *The Visual Computer 27*, 3 (Mar.), 241–248.

Losasso, F., Shinar, T., Selle, A., and Fedkiw, R. 2006. Multiple Interacting Liquids. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006) 25*, 3 (July), 812–819.

Macklin, M., and Müller, M. 2013. Position Based Fluids. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2013) 32*, 4 (July), 104:1–104:12.

Macklin, M., Müller, M., Chentanez, N., and Kim, T.-Y. 2014. Unified Particle Physics for Real-Time Applications. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014) 33*, 4, 153:1–153:12.

Maples, R. E. 2000. *Petroleum Refinery Process Economics*. PennWell Books.

Misztal, M., Erleben, K., Bargteil, A., Fursund, J., Christensen, B., Barentzen, J., and Bridson, R. 2012. Multiphase Flow of Immiscible Fluids on Unstructured Moving Meshes. *In Proceedings of SCA '12*, 97–106.

Monaghan, J. J. 1989. On the problem of penetration in particle methods. *Journal of Computational Physics 82*, 1, 1–15.

Monaghan. 1992. Smoothed Particle Hydrodynamics. *Annual Review of Astronomy and Astrophysics 30*, 1, 543–574.

Monaghan. 1994. Simulating Free Surface Flows with SPH. *Journal of Computational Physics 110*, 2 (Feb.), 399–406.

Monaghan. 2000. SPH without a Tensile Instability. *Journal of Computational Physics 159*, 2 (Apr.), 290–311.

Morris, J., Fox, P., and Zhu, Y. 1997. Modeling Low Reynolds Number Incompressible Flows Using SPH. *Journal of Computational Physics 136*, 1, 214–226.

Müller, M., Charypar, D., and Gross, M. 2003. Particle-Based Fluid Simulation for Interactive Application. *In Proceedings of SCA '03*, 154–159.

Müller, M., Solenthaler, B., Keiser, R., and Gross, M. 2005. Particle-Based Fluid-Fluid Interaction. *In Proceedings of SCA '05*, 237–244.

Müller, M., Heidelberger, B., Hennix, M., and Ratcliff, J. 2007. Position Based Dynamics. *Journal of Visual Communication and Image Representation 18*, 2 (Apr.), 109–118.

Nielsen, M. B., and Østerby, O. 2013. A Two-Continua Approach to Eulerian Simulation of Water Spray. *ACM Transaction on Graphics (Proceedings of SIGGRAPH 2013) 32*, 4 (July), 67:1–67:10.

Park, J., Kim, Y., Wi, D., Kang, N., Shin, S. Y., and Noh, J. 2008. A Unified Handling of Immiscible and Miscible Fluids. *Computer Animation and Virtual Worlds 19*, 3-4 (September), 455–467.

Premože, S., Tasdizen, T., Bigler, J., Lefohn, A., and Whitaker, R. T. 2003. Particle-Based Simulation of Fluids. *Computer Graphics Forum 22*, 3, 401–410.

Raveendran, K., Wojtan, C., and Turk, G. 2011. Hybrid Smoothed Particle Hydrodynamics. *In Proceedings of SCA '11*, 33–42.

Ren, B., Li, C., Yan, X., Lin, M. C., Bonet, J., and Hu, S.-M. 2014. Multiple-fluid SPH Simulation Using a Mixture Model. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2014) 33*, 5 (Aug.), 171:1–171:11.

Schechter, H., and Bridson, R. 2012. Ghost SPH for Animating Water. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012) 31*, 4, 61:1–61:8.

Shao, S., and Lo, Y. 2003. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Advances in water resources 26*, 7, 787–800.

Solenthaler, B., and Pajarola, R. 2008. Density contrast SPH Interfaces. *In Proceedings of SCA '08*, 211–218.

Solenthaler, B., and Pajarola, R. 2009. Predictive-corrective incompressible SPH. *ACM Transactions on Graphics (Proceeding of SIGGRAPH 2009) 28*, 3, 40:1–40:6.

Zhu, Y., and Bridson, R. 2005. Animating Sand as a Fluid. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2005) 24*, 3, 965–972.