# Simulation of Fluid Mixing with Interface Control

Xiaowei He[†♭], Huamin Wang[‡], Fengjun Zhang[†], Hongan Wang[†], Guoping Wang[♮], Kun Zhou[§], and Enhua Wu[†‖]

[†]State Key Lab. of CS, ISCAS [‡]The Ohio State University [♮]Peking University [§]Zhejiang University
[♭]University of Chinese Academy of Sciences [‖]University of Macau

**Figure 1:** *Fluid mixing with different diffusion speeds. Under the Eulerian framework, our convection scheme simulates different fluid mixing effects from immiscible to highly miscible. In this example, the diffusion speeds (D) from left to right are: 0, 0.005, 0.01, 0.02, and 0.04.*

## Abstract

The simulation of fluid mixing under the Eulerian framework often suffers from numerical dissipation issues. In this paper, we present a mass-preserving convection scheme that offers direct control on the shape of the interface. The key component of this scheme is a sharpening term built upon the diffusive flux of a user-specified kernel function. To determine the thickness of the ideal interface during fluid mixing, we perform theoretical analysis on a one-dimensional diffusive model using the Fick's law of diffusion. By explicitly controlling the interface thickness using a spatio-temporally varying kernel variable, we can use our scheme to produce realistic fluid mixing effects without numerical dissipation artifacts. We can also use the scheme to control interface changes between two fluids, due to temperature, pressure, or external energy input. This convection scheme is compatible with many advection methods and it has a small computational overhead.

**CR Categories:** I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

**Keywords:** Fluid mixing, miscible/immiscible fluids, diffuse interface, phase field, fluid control.

## 1 Introduction

Fluid mixing is common in the real world. However, the simulation of fluid mixing under the Eulerian framework [Park et al. 2008; Kang et al. 2010; Bao et al. 2010; Nielsen and Østerby 2013] suffers from numerical dissipation. As a result, the simulated interface is often more blurred than it should be, causing rich mixing details lost. A naïve way to fix this problem is to sharpen the interface by reversing the diffusion process. Unfortunately, the resulting sharpening equation is numerically unstable and it is unclear how sharp the interface should be at each time instant.

In this paper, we present a mass-preserving convection scheme to simulate fluid mixing effects without numerical dissipation artifacts. The key component in this scheme is a sharpening term formulated by the use of a pre-defined kernel function. Using this scheme, we can adjust the shape of the kernel function to easily control the thickness of the simulated fluid interface. Based on the Fick's law of diffusion, we perform a theoretical study on the parameter of the kernel function for an ideal diffusion case. The conclusion from this study allows our convection scheme to formulate the parameter as a spatio-temporally varying function, for simulating realistic fluid mixing behaviors as shown in Figure 1. A controllable interface between two fluids also offers several other advantages for simulation purposes. For example, we can formulate the interface thickness as a function of the temperature, to simulate interface changes caused by heat gain or loss. Meanwhile, we can simulate fluid stratification by reducing the interface thickness over time, as shown in Figure 11. For animation production, artists can use our scheme to directly control the interface smoothness, which is not possible in the past due to numerical dissipation.

Our contributions in this paper are:

- A convection scheme with controllability on the interface thickness, by using a pre-defined kernel function.
- Theoretical analysis on the thickness of a diffusing interface, based on the Fick's law of diffusion.
- A simple method to ensure mass conservation, by distributing surplus mass in one grid cell to others.

Our convection scheme is compatible with many advection methods and it has a small computational overhead.

## 2 Related Work


SCA '15, August 7 - 9, 2015, Los Angeles, California.
 ISBN 978-1-4503-3496-9/15/08…$15.00
DOI: http://dx.doi.org/10.1145/2786784.2786791

**Immiscible flow.** During the past decade, varieties of methods have been proposed to simulate immiscible flows with sharp interfaces. Under the Eulerian framework, early works on fluid simulation are focused on single phase flows such as free-surface water [Foster and Fedkiw 2001; Enright et al. 2002] based on the level set method. Hong and Kim [2005] studied incompressible viscous multi-phase fluids and used the particle level set method for surface tracking. Based on the similar idea, Losasso et al. [2006] developed a more generic approach to simulate multiple phases. An alternative way to simulate immiscible flows is the Volume-of-Fluid (VOF) method, which uses volume fraction to tracking material for each phase [Hirt and Nichols 1981]. Other researchers used particles to track the fluid in an Eulerian grid [Zhu and Bridson 2005]. Recently, Boyd and Bridson [2012] extended the FLIP method to handle two-phase flows.

**Miscible flow.** To create miscible effects, Zhu and collaborators [2006; 2007] studied using the Lattice Boltzmann Method (LBM) to simulate miscible flows. Park and colleagues [2008] extended LBM to efficiently handle both miscible and immiscible flows, but their method suffers from volume loss. To overcome the limitations of LBM, Kang and collaborators [2010] and Bao and colleagues [2010] used volume fraction to model both miscible and immiscible fluids, and they successfully enforced incompressibility on multiple miscible fluids. Our work is aimed at addressing numerical dissipation and interface diffusion/sharpening, by providing direct control on the interface thickness.

**Advection method.** In Eulerian methods, the advection step is important to simulation robustness and accuracy. To use large time steps, Stam [1999] propose a semi-Lagrangian method to trace the material backward along the streamline. This method is robust and easy to implement, but it suffers from volume loss. To improve the accuracy, Kim and colleagues [2005] integrated the Back and Forth Error Compensation and Correction (BFECC) scheme into semi-Lagrangian advection, to reduce dissipation and diffusion in fluid simulation. Song and collaborators [2005] applied constrained-interpolation-profile-based advection and converted potentially dissipative cells into Lagrangian droplets or bubbles. Mullen and colleagues [2007] used an upwind scheme based on the Godunov piecewise-constant approximation to conserve the total volume. To improve quantity conservation and numerical stability when using large time steps, Lentine and colleagues [2011a; 2011b] proposed a conservative semi-Lagrangian advection method. Chentanez and Müller [2012] improved this method for parallel computing on GPU. The convection scheme developed in this work is compatible with many advection methods.

**Phase-field methods.** Our work is also relevant to the phase-field method, a common technique used in computational physics to model complex multi-phase effects, such as solidification [Boettinger et al. 2002] and phase separation [Badalassi et al. 2003]. The basic idea behind the phase field method is to model the interface implicitly using a scalar phase field. Ding and colleagues [2007] studied using this method to simulate incompressible two-phase flows with large density ratios. While their technique conserves mass, the fourth-order derivative term in the Cahn–Hilliard equation makes it difficult to implement numerically. To solve this problem, Sun and collaborators [2007] developed a generic way to track sharp interface using the Allen–Cahn equation [1976] without considering mass conservation. Our method is also relevant to the Allen–Cahn equation, and we are interested in interface control for two-phase fluids of different mixing effects.

**Algorithm 1** `Convection_Update`($\Delta t$)

| | |
|---|---|
| ▹ *Solve the advective flow* $\mathbf{u}$ | |
| Advect the velocity field $\mathbf{u}$; | |
| Apply viscosity on $\mathbf{u}$; | |
| Add gravity and surface tension forces to $\mathbf{u}$; | |
| Perform pressure projection on $\mathbf{u}$; | |
| ▹ *Perform convection with the diffuse flow* $\nabla \cdot \mathbf{J}$ | |
| Advect the diffusion coefficient field $w$; | |
| Update $w$ by the diffusion model; | (Equation 12) |
| Advect the volume fraction field $\phi$; | (Equation 13) |
| Sharpen the volume fraction field $\phi$; | (Equation 14) |
| Apply diffusion on $\phi$; | (Equation 15) |
| Perform volume fraction correction on $\phi$; | (Section 4.2) |

## 3 System Overview

Suppose that the overall volume does not change when one fluid is dispersed into another, we define the volume fraction of the dispersed fluid in an infinitesimal volume as $\phi$ (for $\phi \in [0, 1]$), which is a function of the spatial domain. We perform density convection by updating $\phi$ using both advection and diffusion:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi - \nabla \cdot \mathbf{j} = 0, \tag{1}$$

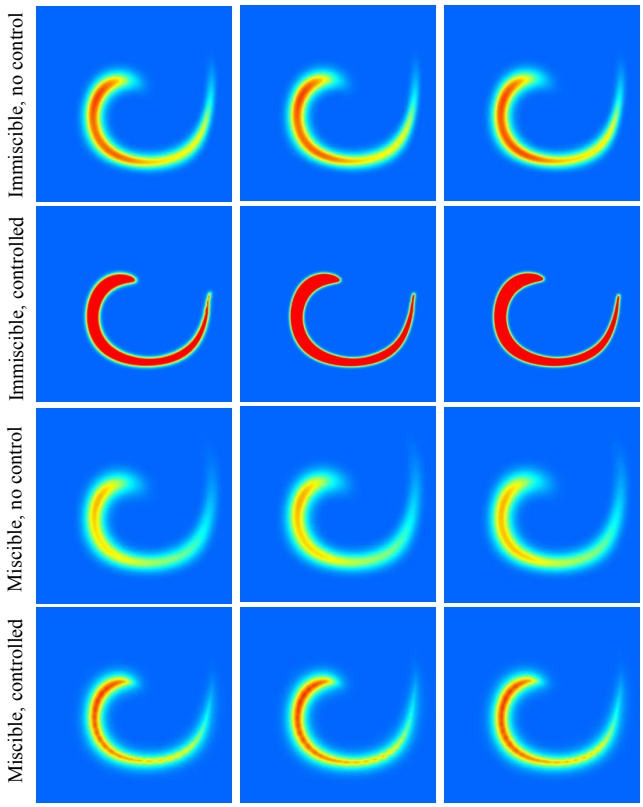
in which $\mathbf{u}$ is the advective velocity field and $\mathbf{j}$ is the diffusive flux. Based on Equation 1, we propose to formulate our system in two processes as Algorithm 1 shows. In the first process, the system solves the two-phase advective flow. Similar to many single-phase systems, it calculates velocity advection, applies viscosity, adds external and surface tension forces, and finally performs pressure projection. The outcome of this process is the advective velocity field $\mathbf{u}$. In the second process, the system uses $\mathbf{u}$ to calculate density advection, and applies a sharpening step and a diffusion step. At the end of the second process, the system performs volume correction to ensure that $\phi \in [0, 1]$ everywhere.

## 4 Two-Phase Diffusive Flow

To model two-phase diffusive flow, the key question is how to formulate the diffusive flux $\mathbf{j}$. A straightforward way is to assume that $\mathbf{j}$ is linearly proportional to the gradient of the volume fraction, so $\mathbf{j} = \alpha \nabla \phi$, in which $\alpha$ controls the flux magnitude. Theoretically, when $\alpha > 0$, the two fluids will be gradually mixed together until $\phi$ becomes constant over the whole domain. Otherwise, if we consider the diffusion process backward, we may sharpen the interface between two fluids as well by $\alpha < 0$. Unfortunately in practice, we cannot control fluid mixing by modifying $\alpha$ for numerical reasons. When $\alpha > 0$, numerical dissipation introduced by density advection will cause the two fluids mixed faster than they should be, as shown in Figure 2; and when $\alpha < 0$, Equation 1 becomes unstable.

To provide controllability on the diffusion and its inverse process, we propose to split the diffusive flux into a diffusive term and a sharpening term: $\mathbf{j} = \alpha \mathbf{j}_\mathrm{d} + \beta \mathbf{j}_\mathrm{s}$, in which $\alpha$ and $\beta$ are two nonnegative coefficients adjusting the magnitudes of these two terms. If $\alpha > \beta$, the two phases will be mixed gradually until the saturation density is reached everywhere. Otherwise, if the two phases are over-dispersed ($\alpha < \beta$), the two phases will be separated. Since we plan to control the thickness of the interface directly, we treat the interface under control as a dynamic equilibrium state, in which the diffusive term and the sharpening term are balanced. So $\alpha = \beta$.

The main question is how do we determine $\mathbf{j}_\mathrm{s}$? Our idea is to define the desired shape of an interface by a kernel function. Here




(a) [Stam 1999] (b) [Mullen et al. 2007] (c) [Lentine et al.2011]

**Figure 2:** *Convection results using different advection methods. By providing direct control on the interface, our method can prevent the interface from being smeared out due to numerical dissipation.*

we choose the function proposed by Boettinger and collaborators [2002] under the diffuse interface model:

$$\phi(d) = \frac{1}{2}\left[1 + \tanh\left(\frac{d}{2w}\right)\right], \tag{2}$$

in which $d$ is the signed distance to the central line of the front and $w$ is a diffusion coefficient. A nice feature of this interface is that we can formulate $\|\nabla\phi\|$ without $d$:

$$\|\nabla\phi\| = \frac{\partial\phi}{\partial d} = \frac{\phi(1-\phi)}{w}. \tag{3}$$

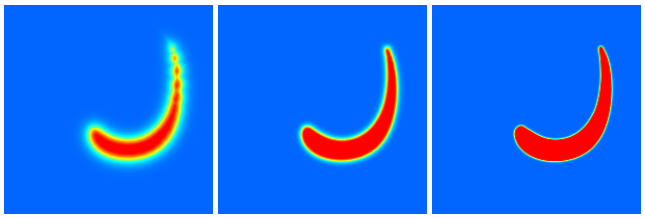
So we can model the sharpening flux $\mathbf{j}_s$ as:

$$\mathbf{j}_s = \frac{\phi(1-\phi)}{w}\mathbf{n}, \tag{4}$$

which is equivalent to $\mathbf{j}_s = \nabla\phi$ when the interface reaches the equilibrium state. Here $\mathbf{n} = \frac{\nabla\phi}{\|\nabla\phi\|}$ denotes the direction for the sharpening flux. Combining $\mathbf{j}_d$ and $\mathbf{j}_s$ together into $\mathbf{j}$, we get:

$$\frac{\partial\phi}{\partial t} + \mathbf{u}\cdot\nabla\phi = \alpha\left[\nabla^2\phi - \nabla\cdot\left(\frac{\phi(1-\phi)}{w}\mathbf{n}\right)\right], \tag{5}$$

in which $\alpha$ is now a control magnitude constant. Intuitively, Equation 5 measures the diffusive flux calculated by the current interface and the diffusive flux calculated by a kernel-shaped interface. It then uses their difference to guide the diffusive flow, until the desired interface thickness is reached. The parameter $\alpha$ controls the speed of this process. According to [Sun and Beckermann 2007], $\alpha$ should be a function of $w$ and $\mathbf{u}_{max}$. For simplicity, we use $\alpha = h$ in our experiment, in which $h$ is the grid cell size.



(a) $w = 2h$ (b) $w = h$ (c) $w = h/2$

**Figure 3:** *The effect of w. By using different diffusion coefficient w, we can adjust the interface thickness in an equilibrium state.*

Using Equation 5, we can modify $w$ to control the interface width as Figure 3 shows. We typically set $w \in [h/2, +\infty)$. A larger $w$ causes a blurred interface. Increasing $w$ causes the fluids to mix from low miscibility to high miscibility. In the extreme case when $w$ goes to infinity, Equation 5 simply diffuses the fluid over the whole domain until the mixture becomes homogeneous.

### 4.1 Determining the Diffusion Coefficient

As we gain the ability to control the interface thickness using our convection scheme, the immediate question next is: *how can we determine the interface thickness in the fluid mixing process*? Since the thickness is controlled the diffusion coefficient $w$, essentially we need to know how $w$ should vary during fluid mixing. To answer this question, we perform theoretical analysis on a purely diffusive flow based on the Fick's law of diffusion.

Let the sharp interface between two fluids be defined by the Heaviside function at time $t = 0$:

$$\phi(x,0) = H(x) = \begin{cases} 0, x < 0 \\ 1, x \geq 0 \end{cases}, \tag{6}$$

we solve the diffusion equation $\phi_t = D^2\nabla^2\phi.$, in which $D$ is the diffusion speed. For simplicity, we assume that there is no density advection. When $D$ is constant, we get the analytical solution as:

$$\phi(x,t) = \frac{1}{2D\sqrt{\pi t}}\int_{-\infty}^{\infty} H(\xi)\, e^{-(x-\xi)^2/4D^2t} d\xi. \tag{7}$$

Since Equation 7 is similar to Gauss error function erf, next we can simplify it into:

$$\phi(x,t) = \frac{1}{2}\left[1 + \mathrm{sgn}(x)\,\mathrm{erf}\left(\frac{|x|}{2D\sqrt{t}}\right)\right], \tag{8}$$

where $\mathrm{sgn}(x)$ is the sign of $x$. This function monotonically increases from 0 to 1. If we assume that the interface is defined as $\phi \in (0.05, 0.95)$, we can calculate the interface thickness as:

$$W = x|_{\phi=0.95} - x|_{\phi=0.05} = 4D\sqrt{t}\cdot\mathrm{erf}^{-1}(0.9). \tag{9}$$

Meanwhile, the interface thickness from $\phi = 0.05$ to $\phi = 0.95$ should satisfy $W = 3\sqrt{2}w$ when using our controllable interface model, according to [Sun and Beckermann 2007]. So we know:

$$w = \frac{2\sqrt{2}}{3}D\sqrt{t}\cdot\mathrm{erf}^{-1}(0.9). \tag{10}$$

Equation 10 shows that $w$ should be a linear function of $\sqrt{t}$, to produce ideal diffusive effects. Figure 5 verifies the correctness of our model, when we apply it to simulate this 1D example.

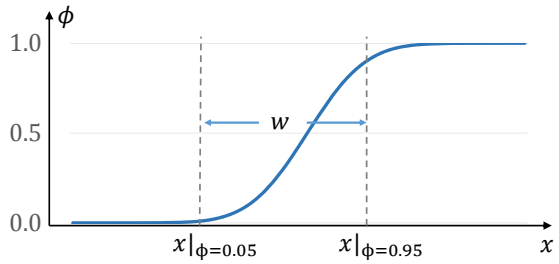



**Figure 4:** *A 1D diffusive flow example. We define the interface between two fluids as the interval from $\phi = 0.05$ to $\phi = 0.95$.*

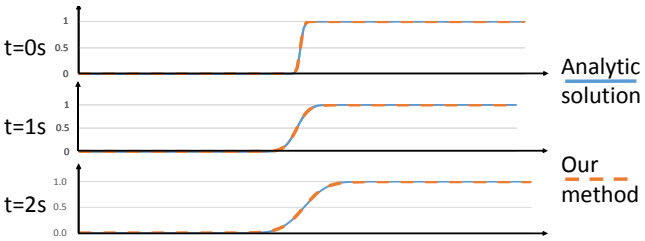



**Figure 5:** *A comparison example. This example shows that by defining w using Equation 10, the simulation result of our controllable interface model matches well with the analytic solution.*

In reality, the interface profile can be different from the curve defined in Equation 8. To represent arbitrary interfaces, we assume $w$ is not only temporally varying, but also spatially varying. Intuitively speaking, any complex interface is regarded as a combination of the simple curves defined in Equation 8. In other words, for any infinitesimal part of a complex interface, we can always find an equivalent part of a certain curve defined in Equation 8, where the diffusion constant is computed by inverting Equation 10. So given $w^t(x)$ at $x$ and time $t$, we first use Equation 10 to estimate the amount of time the interface has already been diffused:

$$\hat{t}(x) = \left( \frac{w^t(x)}{\frac{2\sqrt{2}}{3} D \cdot \mathrm{erf}^{-1}(0.9)} \right)^2. \tag{11}$$

We then update $w$ by:

$$w^{t+1}(x) = \frac{2\sqrt{2}}{3} D \sqrt{\hat{t}(x) + \Delta t} \cdot \mathrm{erf}^{-1}(0.9) \tag{12}$$

When extending this method to 3D cases, we consider the thickness in the normal direction $\mathbf{n}$ and update $w$ over time in the same way. However, it should be pointed out that Equation 12 only holds for diffusive flows that strictly follow the Fick's law of diffusion, in which we also neglects all other factors that can cause sharpening/widening of the interface, e.g., fluid motion. We will explore those more complex interface control methods in our future work.

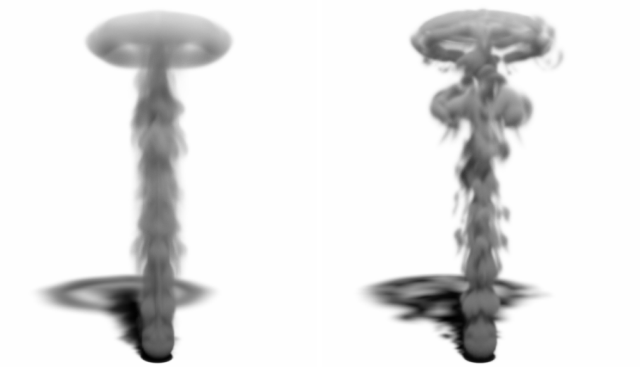
## 4.2 Implementation

We use method of characteristics to split Equation 5 into three steps:

$$\phi' = \phi^t - \Delta t \left( \mathbf{u} \cdot \nabla \phi^t \right), \tag{13}$$

$$\phi'' = \phi' - \Delta t \alpha \nabla \cdot \left( \frac{\phi'(1-\phi')}{w^{t+1}} \mathbf{n} \right), \tag{14}$$

$$\phi^{t+1} = \phi'' + \alpha \Delta t \nabla^2 \phi^{t+1}, \tag{15}$$



(a) Without interface control (b) With interface control

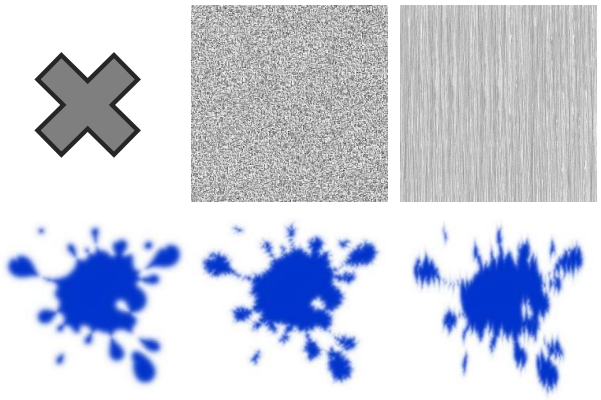
**Figure 6:** *Smoke with a low diffusion speed. Without interface control, numerical dissipation causes the simulation result in (a) more blurred than our result with interface control in (b).*

which are corresponding to explicit advection, explicit sharpening, and implicit diffusion respectively. In our experiment, we test three advection methods, including semi-Lagrangian advection [Stam 1999], conservative semi-Lagrangian advection [Lentine et al. 2011a] and upwind advection [Mullen et al. 2007]. Figure 2 and 6 show their results without and with using our interface control method. To make the sharpening step mass-preserving, we calculate $\frac{\phi'(1-\phi')}{w^{t+1}} \frac{\nabla \tilde{\phi}}{|\nabla \tilde{\phi}| + \epsilon}$ at each grid cell, obtain its value at each face by taking the average, and then compute its divergence. Here $\epsilon$ is a small positive number and $\tilde{\phi}$ is a slightly blurred version of $\phi'$ by Laplacian smoothing. They are used to prevent the sharpening step from failures, when $|\nabla \phi'|$ becomes too close to zero.

To calculate the spatially varying variable $w^{t+1}$, we first perform semi-Lagrangian advection to get an intermediate variable $w'$ at each grid cell. When using semi-Lagrangian, the interpolation should be weighted by volume fraction as well. After that, we calculate $w^{t+1}$ from $w'$ using Equation 12, and we solve the linear system in Equation 15 in the end.

**Volume fraction correction.** Although the advection step can be mass-preserving as Lentine and collaborators [2011a] showed, the sharpening step and the diffusion step can still cause $\phi$ to go beyond the range $[0, 1]$. To solve this issue, we propose a simple correction step to explicitly project wrong $\phi$s back. Our idea is to propagate the additional volume to neighboring cells, if they can accept more volume. For simplicity, here we discuss the $\phi > 1$ case only. The $\phi < 0$ case can be handled in the same fashion.

We first add all of the grid cells satisfying $\phi > 1$ into a heap. In each iteration, we remove the grid cell $\mathbf{c}$ with the largest $\phi$ from the heap, label it as *visited*, and distribute $\phi_\mathbf{c} - 1$ uniformly to its unvisited neighbors. If $\phi$ of any neighbor changes from $\phi \leq 1$ to $\phi > 1$, we add it into the heap as well. The whole process ends when the heap becomes empty. Although it is rare, we may notice an issue when all of $\mathbf{c}$'s neighbors are visited and $\phi_\mathbf{c} - 1$ cannot be distributed. To solve this problem, we sum all of these undistributed volume into a global volume and redistribute it uniformly to every unvisited cells, after the original process ends. We note that this redistribution may cause more grid cells to be $\phi > 1$, so we mark them as visited, collect their surplus volumes and distribute them again. Since the total volume does not change according to the discretization in Subsection 4.2, this method is guaranteed to converge.



(a) (b) (c)

**Figure 7:** *Diffusion Control. Different diffusion effects (bottom) can be created after using different textures (top) to modify the sharpening direction. (a) Isotropic diffusion; (b)(c) Anisotropic diffusion.*

## 5 Two-Phase Advective Flow

To make this paper self-contained, we will discuss the advective flow (i.e., the bulk flow) in this section. Let $\phi$ be the volume fraction of fluid A, and $1-\phi$ be the volume fraction of fluid B. According to the convection equation in Equation 1, we have:

$$\begin{cases} \frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \mathbf{u}) - \nabla \cdot \mathbf{j}_A = 0, \\ \frac{\partial (1-\phi)}{\partial t} + \nabla \cdot ((1-\phi)\mathbf{u}) - \nabla \cdot \mathbf{j}_B = 0. \end{cases} \tag{16}$$

in which $\mathbf{j}_A$ and $\mathbf{j}_B$ are the diffusive fluxes of the two fluids. Combing the two formulae in Equation 16 together, we get:

$$\nabla \cdot \mathbf{u} = \nabla \cdot (\mathbf{j}_A + \mathbf{j}_B). \tag{17}$$

Since A and B are the only two fluids within the domain, we must have $\mathbf{j}_A = -\mathbf{j}_B$ everywhere, and we can formulate the governing equation of the advective flow using the Navier-Stokes equations:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0, \\ \rho \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} &= -\nabla p + \nabla \cdot \mu \left( \nabla \mathbf{u} + \nabla \mathbf{u}^T \right) + \mathbf{f}_s + \mathbf{f}_b, \end{aligned} \tag{18}$$

in which $p$ is the pressure, $\mathbf{f}_e$ is the body force, and $\mathbf{f}_s$ is the surface tension force. Since we handle diffuse interfaces rather than sharp interfaces (such as the level sets), there is no jump condition in our system. Instead, we define the density and viscosity as a smooth function: $\rho = \phi\rho_A + (1-\phi)\rho_B$ and $\mu = \mu_A\phi + \mu_B(1-\phi)$. We solve Equation 18 in a typical way [Stam 1999], by applying velocity advection, viscosity, external forces, and pressure projection.

One tricky issue here is how to handle the non-uniform viscosity. Similar to [Rasmussen et al. 2004], we split the asymmetric system into an asymmetric component and a symmetric component. To improve the system performance, we further divide the symmetric component into a non-uniform component and a uniform component. We then solve the asymmetric component and the non-uniform component explicitly, and the uniform component implicitly. We note that such an explicit-implicit viscosity scheme can be unconditionally stable, by choosing a proper magnitude in the uniform component. More details about this can be found in [Douglas Jr and Dupont 1971].

(a) Before anisotropic diffusion (b) After anisotropic diffusion

**Figure 8:** *Tiger. Anisotropic diffusion in Figure 7(c) is performed on a tiger's body to make it look flurry.*

## 6 Results and Discussions

(Please see the supplemental video for more animation examples.) We implemented our system and tested it on a quad-core Intel Xeon w3550 3.07GHz workstation with 10GB memory. We use different time steps for the simulation of the two flows, as long as they are synchronized within a certain interval. Currently, we set the time step for the diffusive flow to be $\Delta t = \min(\Delta t, Ch/\max\|\mathbf{u}\|)$, in which $C = 0.5$ is the CFL number and $h$ is the sampling distance.

**Diffusivity comparison.** Figure 1 demonstrates the ability of our approach in handling a spectrum of liquid diffusivity, from immiscible to highly miscible. The liquid diffusivity can be easily adjusted by changing the diffusion speed $D$. Thanks to interface control, our result is free of numerical dissipation artifacts.

**Anisotropic Diffusion.** Figure 7 demonstrates the ability of our approach in producing a variety of interesting diffusion effects. For the anisotropic diffusion, we use a texture to modify the sharpening direction in Equation 4 as

$$\hat{\mathbf{n}} = \gamma \mathbf{M}\mathbf{n}. \tag{19}$$

Here $\gamma$ is a constant that controls the magnitude of sharpening and we set it to 1.2 in our experiments. $\mathbf{M}$ is a matrix that controls the sharpening direction, which is constructed from the texture as $\mathbf{M} = \mathbf{t}\mathbf{t}^T$ with $\mathbf{t}$ being the normalized gradient of the texture. By using different textures, we can easily create various diffusion effects. We also perform the anisotropic diffusion to a tiger's body to make it look furry as shown in Figure 8.

**Phase change.** This example (in Figure 11) shows the phase change effect simulated by our system. When steam is injected into air, small water drops gradually form as the temperature decreases. In this simulation, we use no gravity and we create the condensation effect by sharpening the interface between stream and air. So we update the control parameter $w$ in an opposite way as:

$$w^{t+1} = \frac{2\sqrt{2}}{3} D \sqrt{\hat{t} - \Delta t} \cdot \mathrm{erf}^{-1}(0.9). \tag{20}$$

We clamp $w$ to $h/2$, if the result from Equation 20 is below that.

**Ink.** Figure 9 demonstrates the Rayleigh-Taylor instability effect, when ink is poured into water. In this example, the ink density is 1050kg/m3, which is slightly higher than the water density 1000kg/m3. The diffusion speed $D$ is set to 0.01. Under the gravity, ink sinks and blurs gradually, which forms rich details.

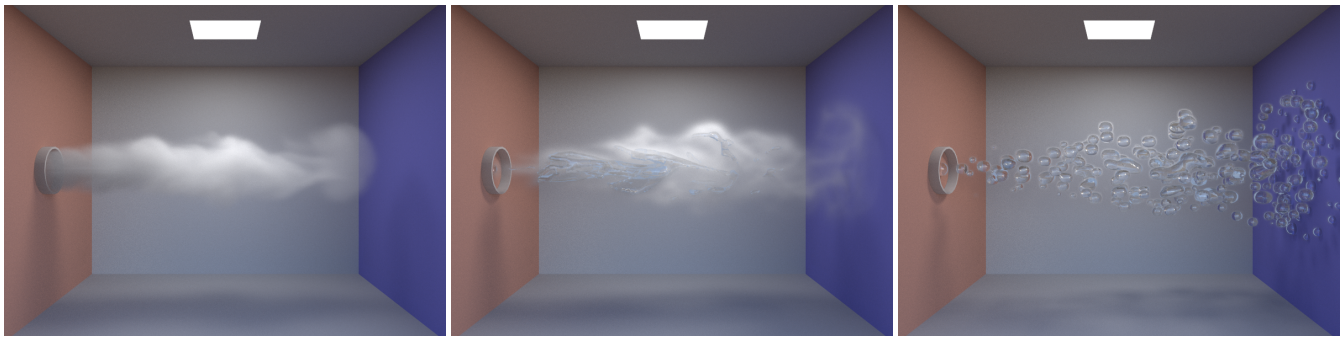


**Figure 11:** *Phase transition. This example simulates phase transition between steam and water.*

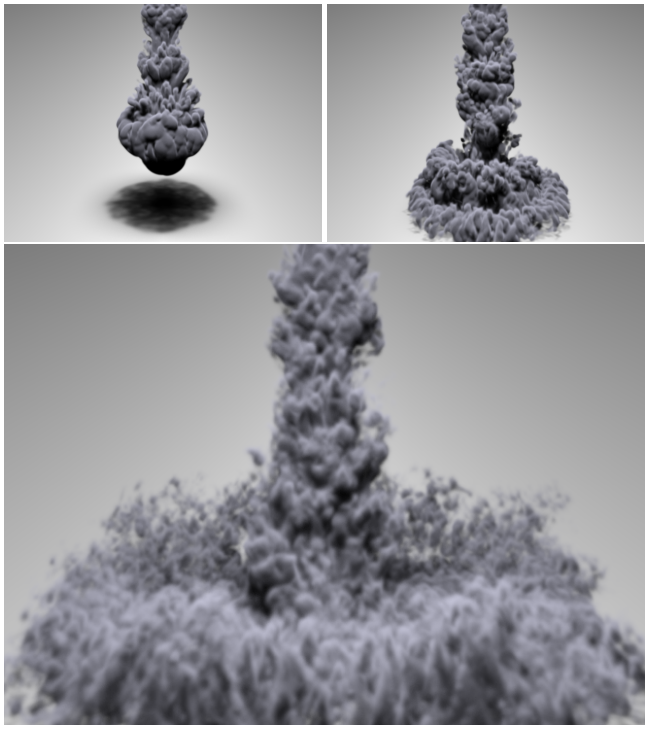


**Figure 9:** *Ink. Interesting fluid details are formed when heavy ink is injected into the water container.*

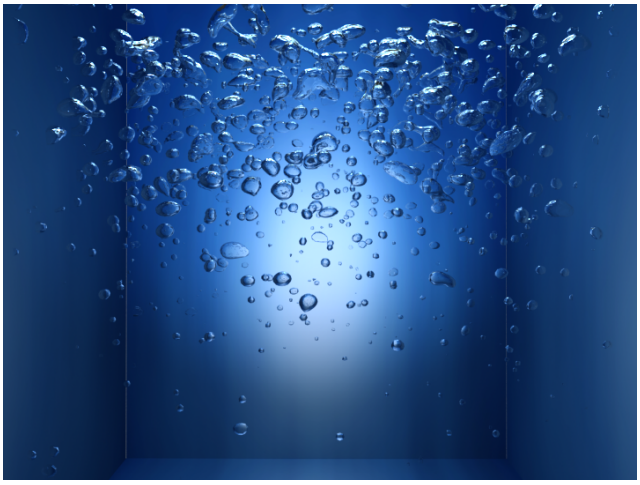


**Figure 10:** *Bubbles. This example simulates both large and small air bubbles, and their transitions in a water container.*

**Bubbles.** This example in Figure 10 shows that our system can simulate bubbles in water. For rendering, we use the marching cube method to extract water-bubble surfaces for large bubbles. The system can naturally handle merging or splitting transitions between large and small bubbles without using particles.

**Computational cost.** In our current implementation, the most time-consuming part is solving the fluid incompressibility with a preconditioned conjugate gradient method. For the interface control, we find a simple Jacobi iteration method with 20 iterations is enough to get good results and the total computational cost of the convection scheme takes approximately 10% of the whole cost.

**Limitations.** The thickness of the controlled interface needs to be at least twice the grid cell size to prevent numerical instability, so it cannot simulate highly immiscible fluids and preserve sharp surface features during convection. In that case, a sharp interface model, such as the level set method, should be used instead. Since the whole system is formulated based on volume fraction, it is not clear how we can handle more than two fluids. Finally, we derive the parameters and the interface thickness over time based on theoretical analysis under the Fick's law of diffusion. In the real world, however, the mixing process of two fluids can be much more complicated due to many other factors.

## 7 Conclusions and Future Work

We present an Eulerian-based approach to simulate mixing and unmixing effects of two fluids, by directly controlling their interface. Since it no longer suffers from numerical dissipation, it can produce more rich details in animation results. It can also be used as a convenient tool for users to control specific mixing and unmixing effects in animation production.

In the future, we plan to study the simulation of more than two fluids. We are also interested in investigating the actual physics causing phase transitions of two fluids for simulation purposes. Another interesting problem we will study is whether the use of Lagrangian particles can improve the convection quality for preserving sharp features, similar to the particle level set method. Finally, we will explore the use of our approach in real-time simulation by GPU.

## 8 Acknowledgements

The authors would like to thank the anonymous reviewers for their constructive comments and Ke Ding for modeling, rendering and video making. The research was supported by the National Basic Research Program of China (No. 2013CB329305) and the National Natural Science Foundation of China (No. 61232013, 61173059, 6140051239, 61232014, 61272326, 61421062). Enhua Wu was also supported by the research grant of University of Macau (MYRG202(Y3-L4)-FST11-WEH). Xiaowei He was also supported by the Open Project Program of the State Key Lab of CAD&CG(No. A1424), Zhejiang University.

## References

Allen, S. M., and Cahn, J. W. 1976. Mechanisms of phase transformations within the miscibility gap of Fe-rich Fe-Al alloys. *Acta Metallurgica 24*, 5, 425–437.

Badalassi, V., Ceniceros, H., 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, 401–410.

Boettinger, W., Warren, J., Beckermann, C., and Karma, A. 2002. Phase-field simulation of solidification. *Annual review of materials research 32*, 1, 163–194.

Boyd, L., and Bridson, R. 2012. MultiFLIP for energetic two-phase fluid simulation. *ACM Trans. Graph. (SIGGRAPH) 31*, 2, 16.

Chentanez, N., and Müller, M. 2012. Mass-conserving Eulerian liquid simulation. In *Proceedings of SCA*, Eurographics Association, 245–254.

Ding, H., Spelt, P. D., and Shu, C. 2007. Diffuse interface model for incompressible two-phase flows with large density ratios. *Journal of Computational Physics 226*, 2, 2078–2095.

Douglas Jr, J., and Dupont, T. 1971. Alternating-direction Galerkin methods on rectangles. *Numerical Solution of Partial Differential Equations, II (SYNSPADE 1970)*, 133–214.

Enright, D., Marschner, S., and Fedkiw, R. 2002. Animation and rendering of complex water surfaces. *ACM Trans. Graph. (SIGGRAPH) 21*, 3, 736–744.

Foster, N., and Fedkiw, R. 2001. Practical animation of liquids. In *Proceedings of SIGGRAPH 2001*, E. Fiume, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 23–30.

Hirt, C., and Nichols, B. 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics 39*, 1, 201 – 225.

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

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

Kim, B., Liu, Y., Llamas, I., and Rossignac, J. 2005. Flowfixer: Using BFECC for fluid simulation. In *Proceedings of NPH*, 51–56.

Lentine, M., Aanjaneya, M., and Fedkiw, R. 2011. Mass and momentum conservation for fluid simulation. In *Proceedings of SCA*, 91–100.

Lentine, M., Grétarsson, J. T., and Fedkiw, R. 2011. An unconditionally stable fully conservative semi-Lagrangian method. *Journal of Computational Physics 230*, 8, 2857–2879.

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

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

Nielsen, M. B., and Østerby, O. 2013. A two-continua approach to Eulerian simulation of water spray. *ACM Trans. Graph. (SIGGRAPH) 32*, 4.

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, 455–467.

Rasmussen, N., Enright, D., Nguyen, D., Marino, S., Sumner, N., Geiger, W., Hoon, S., and Fedkiw, R. 2004. Directable photorealistic liquids. In *Proc. of SCA*, 193–202.

Song, O.-Y., Shin, H., and Ko, H.-S. 2005. Stable but nondissipative water. *ACM Trans. Graph. 24*, 1 (Jan.), 81–97.

Stam, J. 1999. Stable fluids. In *Proc. of SIGGRAPH '99*, Computer Graphics Proceedings, Annual Conference Series, 121–128.

Sun, Y., and Beckermann, C. 2007. Sharp interface tracking using the phase-field equation. *Journal of Computational Physics 220*, 2, 626–653.

Zhu, Y., and Bridson, R. 2005. Animating sand as a fluid. *ACM Trans. Graph. (SIGGRAPH) 24*, 3 (July), 965–972.

Zhu, H., Liu, X., Liu, Y., and Wu, E. 2006. Simulation of miscible binary mixtures based on Lattice Boltzmann method. *Computer Animation and Virtual Worlds 17*, 3-4, 403–410.

Zhu, H., Bao, K., Wu, E., and Liu, X. 2007. Stable and efficient miscible liquid-liquid interactions. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, 55–64.