Hybrid Simulation of Miscible Mixing with Viscous Fingering

Seung-Ho Shin†, Hyeong Ryeol Kam‡, and Chang-Hun Kim§
Korea University

Abstract

By modeling mass transfer phenomena, we simulate solids and liquids dissolving or changing to other substances. We also deal with the very small-scale phenomena that occur when a fluid spreads out at the interface of another fluid. We model the pressure at the interfaces between fluids with Darcy’s Law and represent the viscous fingering phenomenon in which a fluid interface spreads out with a fractal-like shape. We use hybrid grid-based simulation and smoothed particle hydrodynamics (SPH) to simulate intermolecular diffusion and attraction using particles at a computable scale. We have produced animations showing fluids mixing and objects dissolving.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Physically based modeling

1. Introduction

In computer graphics, many fluid simulation techniques have been developed and used to create realistic animations. However most of those techniques focus on immiscible fluids such as water, air, and bubbles. Losasso et al. [LSSF06] simulated fire and more than two liquids in the same scene, but did not deal with miscible fluids. Recently, Zhu et al. [ZLLW06], Mullen et al [MMTD07], and Park et al. [PKW∗08] have presented miscible fluid simulations. However, they excluded the physical and chemical phenomena in which fluids are mixed and react with each other. When two different fluids meet, they spread out in a fractal shape because of physical pressure differences and diffusion laws. We can see this happen when ink is dropped into water. Substances can also melt and be dissolved by mass transfer caused by chemical reaction, and then change into other substances. Molecules of solute float about in the flowing fluid and spread out in a complicated fashion. We propose methods of simulating complicated fluid phenomena like those described above, and present animations of the interaction of miscible fluids such as ink, water, bubbles, and melting solids.

† winstorm@korea.ac.kr
‡ neary@korea.ac.kr
§ chkim@korea.ac.kr
implement since it is modelled with pressure jump. We also model the mass transfer phenomena caused by chemical reactions using the equation of heat-dependent mass transfer proposed by Mihalef et al. [MUM∗06] and Son et al. [SDR99]. These techniques allow us to animate substances that change phase and melt to form other substances.

Hong et al. [HLYK08] and Losasso et al. [LTKF08] simulated detailed splashing and bubble motion by combining a grid-based version of Euler’s method with a particle-based Lagrangian approach. We simulate the motion of molecule-like particles that represent a concentration using this hybrid method. These concentration particles experience forces that include diffusion and quasi-intermolecular attraction and repulsion. We control and simulate these forces simply using smoothed particle hydrodynamics (SPH).

2. Related Work

Numerical simulation of the Navier-Stokes equation has become a standard technique for the realistic animation of fluids. Foster and Metaxas [FM96] introduced a fully three-dimensional Navier-Stokes solver into computer graphics, and an effective and robust solution to the Navier-Stokes equation that includes semi-Lagrangian advection was reported by Stam [Sta99]. Foster and Fedkiw [FF01] used a conjugate gradient method to solve the Poisson equation and an implicit level-set surface to represent the interface area effectively. Their method smoothes the fluid interface, and changes of topology are represented robustly. This method has been extended to a particle level-set method by Enright et al. [EMF02], with the addition of conservation of mass. Losasso et al. [LGF04] used an adaptive octree data structure to show detailed fluid effects such as the crown phenomenon. Losasso et al. [LSSF06] then simulated various immiscible fluids such as oil, water, or fire using multiple level-sets. Hong and Kim [HK05] dealt with interface discontinuities using the GFM [FAMO99, KFL00]. They considered surface tension at the interface between two immiscible fluids at the projection step and introduced a discontinuous viscosity condition. Shin and Kim [SK07] modeled the force that drives liquids to a target shape using a pressure jump within the GFM. We model the viscous fingering phenomenon in a similar way.

There has been a spate of recent research in the computer graphics community, on the animation of phase transitions. Mihalef et al. [MUM∗06] animated air bubbles in boiling water. They controlled the number and the volume of the bubbles produced by applying the equation of mass transfer by heat. Kim et al. [KLL∗07] conserved the volume of air bubbles by revising the local value of divergence. Losasso et al. [LIGF06] simulated phase transitions such as ice melting and paper burning. Wojtan et al. [WCMT07] animated corrosion and erosion of solid. In our scenario, phase changes, liquid to liquid as well, result from chemical reactions.

Desbrun and Cani [DC96] modified the SPH method to handle viscous fluids, and Müller et al. [MCG03] proposed an interactive method in which SPH underpins the simulation of water; these authors also developed a multiphase SPH method to describe fluids of different compositions [MSKG05]. Cleary et al. [CPPK07] have improved the realism with which the collision of foam and bubbles on a complicated surface can be modeled.

Recently, much of effort has been applied to improving the efficiency of fluid simulations by modeling the details of fluid behavior on an underlying subgrid, which can be achieved by combining a Eulerian grid with Lagrangian particles. Kim et al. [KCC∗06] modeled splashing by combining the SPH method with escaped particles from a particle level-set. Losasso et al. [LTKF08] developed a two-way coupling between a particle level-set and SPH to simulate diffuse regions such as splashing. Hong et al. [HLYK08] used SPH to model lively air bubbles on a coarse grid while retaining small-scale features of the flow. Lee et al. [LHK08] proposed a way of making particle and level-set representations more interchangeable. We use the techniques mentioned above to simulate floating particles representing concentration factors.

3. Modeling Miscible Fluids with Multiple Level-Sets

To simulate fluids mixing, it is necessary to simulate more than two fluids and track the interfaces between them. Losasso et al. [LSSF06] tracked numerous interfaces of immiscible fluids using multiple level-sets, and this is the approach that we use here. We use multiple level-sets in miscible fluids to track mixing surfaces in which we make a pres-
sure jump according to Darcy’s Law and simulate pressure term. We then trace the new mixing surfaces that viscous fingering creates. Chemical mass transfer also occurs at mixing surfaces.

We use fields containing the velocity of each substance for each multilevel-set. When using a single velocity field, it is difficult to represent the mixing surfaces’ characteristics described above, because the velocity field of each fluid is scattered by the effect of the diffusion and pressure terms. We therefore must simulate scenarios that satisfy the divergence-free condition for each fluid while considering that fluid’s changes of phase. We calculate the pressure term for each fluid incorporating the effect of Darcy’s Law and mass transfer. Each velocity field is extrapolated and then advected by the semi-Lagrangian advection method. We create a single velocity field for the fluid by combining the calculated velocity fields. Then we calculate the pressure term for the velocity field of the entire fluid in an additional step. The velocity field must be adjusted so that it fulfills the divergence-free condition, and finally we calculate the diffusion term. We perform this simulation repeatedly while dividing the velocity field to match the multiple multilevel-sets. This method is expensive, but it is able to control the changes to the multiple level-sets that occur after reactions and interactions, so it is necessary. Figure 1 shows a schematic outline of our method.

In this paper, we create and use combined level-sets to describe the mixing surfaces of entire fluids, as shown in Figure 2. This method allows us to choose what data we assign to the combined level-set of the entire fluid, and to the level-set of each fluid when calculating surface tensions. When using numerous level-sets, it is possible for the direction of the surface tension to be miscalculated at the intersection between level-sets, as shown in Figure 2 (right). We therefore consider all the level-sets to represent a single fluid, and calculate the surface tension using the combined level-set data. This allows us to determine the correct direction for the surface tension across the whole mixed fluid, as shown in Figure 2 (left).

Figure 2: When using a combined level-set, surface tension (left) is more accurate than when using individual level-sets (right).

4. Basic Fluids Simulation

The Navier-Stokes equation, which is the basis of our simulation, preserves mass and momentum:

\[ u_t = -(u \cdot \nabla)u + \frac{\nabla \cdot \tau}{\rho} - \frac{\nabla p}{\rho} + f, \]  
(1)

\[ \nabla \cdot u = 0, \]  
(2)

where \( u \) is the velocity, \( \tau \) is the viscous stress tensor, and \( \rho \) is the density. The term \( f \) can be used to add external forces such as gravity and buoyancy. The numerical simulation of Eqs. (1) and (2) requires the value of \( u \) to be updated from \( u^n \) to \( u^{n+1} \) at the \( n^{th} \) time step. We discretize Eq. (1) by splitting it into two equations by introducing an intermediate velocity \( u^* \):

\[ \frac{u^* - u^n}{\Delta t} = -(u \cdot \nabla)u^n + \frac{\nabla \cdot \tau}{\rho} + f, \]  
(3)

\[ \frac{u^{n+1} - u^*}{\Delta t} = -\frac{\nabla p}{\rho}. \]  
(4)

The variable \( u^* \) can be used to compute the advection term using the semi-Lagrangian method of Stam [Sta99]. We can write the divergence of Eq. (4) as a form of Poisson’s equation:

\[ \nabla^2 p = \frac{\rho}{\Delta t} \nabla \cdot u^*. \]  
(5)

Once the pressure profile has been determined by solving this equation, we can obtain the final velocity profile:

\[ u^{n+1} = u^* - \frac{\Delta t}{\rho} \nabla p. \]  
(6)

There is a discontinuous pressure profile at the interface \( \Gamma \) between two different fluids. It is possible to take the discontinuous pressure at the interface into account, using GFM [HK05, KFL00]. The pressure at node \( i \), which is \( p_i \), and the pressure at node \( i+1 \), which is \( p_{i+1} \), are extrapolated across \( \Gamma \) to determine the ghost values, \( p_i^G \) and \( p_{i+1}^G \):

\[ p_i^G = p_i + J; \]  
(7)

\[ p_{i+1}^G = p_{i+1} - J. \]  
(8)

Using these equations, Eq. (5) can be expanded as follows [HK05]:

\[ p_{i+1}^G - p_i^G = \frac{\rho}{\Delta t} \nabla \cdot u_i. \]  
(9)
to Darcy’s Law, the pressure gradient vector is:

$$\nabla p_{\text{Darcy}} = \frac{\mu U}{k} + \rho g.$$  \hspace{1cm} (11)

where $\mu$ is viscosity, $U$ is velocity, $k$ is permeability, $\rho$ is density, and $g$ is gravity.

There is a discontinuity of density and viscosity at an interface between two fluids. To calculate the density at a fluid interface exactly, we use an equation proposed by Losasso et al. [LSSF06]:

$$\hat{\beta} = (\beta^- + \beta^+)/\left\{ \beta^+ + (1 - \theta)\beta^- \right\},$$  \hspace{1cm} (12)

where $\theta = |\phi(x_i)|/(|\phi(x_i)| + |\phi(x_{i+1})|)$, and the $-$ and $+$ superscripts refer to values from different sides of the interface. The viscosity at the interface can be calculated in a similar way. As mentioned in Section 3, each substance has its own velocity field, and $U$ is the velocity of the solute.

The porosity equation is formulated using the curvature of the interface of the solute that is infiltrating into a fluid. As shown in Figure 3, if the curvature is high, the fluid surface is convex and the probability of infiltration is high, but if this shape is flat or concave, the probability of infiltration is low. Thus, we express porosity as a constant that we set to 0.1, and $k$ is a constant that we set to 0.1, and $k$ is a constant that we set to 0.1. The viscosity at the interface can be calculated in a similar way. The viscosity at the interface can be calculated in a similar way. As mentioned in Section 3, each substance has its own velocity field, and $U$ is the velocity of the solute.

$$\nabla \cdot \mathbf{U} = 0.$$  \hspace{1cm} (13)

This formulation allows us to demonstrate viscous fingering at a mixing surface, as shown in 2D in Figure 4. This is a similar structure to that shown in a real photograph of mixing fluids, due to Habermann [Hom87], also reproduced in Figure 4.

6. Chemical Mass Transfer

When fluids mix, chemical reactions can occur. Then, the volume of fluid may decrease or increase and some of the...
fluid may change into another type of fluid. We model and simulate this kind of mass transfer. Mihalef et al. \cite{MUM*06} simulated the mass transfer that occurs when water boils. The mass transfer because of chemical reaction is similar to the phase transition that can be caused by heat. The rate of mass transfer resulting from a chemical reaction can be expressed as follows:

\[ \dot{m}_c = - \frac{c_f_{\text{solute}} \cdot N - c_f_{\text{solvent}} \cdot N}{H}, \]  

(14)

where \( c_f \) is the chemical flux defined by \( c_f = -D \nabla C_{\text{cell}} \cdot N \) is the outward facing normal from \( c_f_{\text{solute}} \) to \( c_f_{\text{solvent}} \), \( H \) is the heat of reaction, \( C_{\text{cell}} \) is the concentration of each cell, and \( D \) is the coefficient of diffusivity for the concentration.

The rate of mass transfer by chemical reaction depends on the concentration of the substance. Depending on the rate of mass transfer, the level-set of the fluid is updated as follows:

\[ \phi^* = \phi_n - \Delta t \frac{\dot{m}_c}{\hat{\rho}} |\nabla \phi|, \]  

(15)

\( \phi^* \) is advected to \( \phi_{n+1} \) using level-set method after updated. Because the volume of the fluid changes, we must revise the divergence value of the pressure term by using the rate of mass transfer to control the volume of fluid, as follows:

\[ \nabla \cdot \mathbf{u}^{n+1} = \frac{\dot{m}_c}{\rho} |\nabla \phi|, \]  

(16)

\[ \nabla^2 p = \frac{\hat{\rho}}{\Delta} \nabla \cdot \mathbf{u}^{n+1} = \frac{\dot{m}_c}{\Delta} |\nabla \phi|, \]  

(17)

This method is similar to those explained in \cite{MUM*06, KLL*07}. This allows us to simulate phenomena in which the volume of the substance changes, such as the melting of a solid because of a chemical reaction. Figure 5 shows a solid teapot melting in a transparent liquid. Because of mass transfer, the volume of the teapot decreases. The solid teapot shape is defined by an implicit surface using level-set data, which makes it easy to implement this scenario.

7. Hybrid Method

It is hard to model intermolecular forces using only grid-based advection of concentration. We therefore simulate the effect of molecular diffusion and interaction on concentration using what we call concentration particles. The advection of concentration is simulated using both a grid-based model of semi-Lagrangian advection and particle-based advection. We can assume that the concentration particles are not completely absorbed in the solvent in which they dissolve, but float about like oversize molecules. Figure 6 shows a 2D simulation of the mixing of two substances using this hybrid method.

7.1. Concentration Particles and Absorption

Concentration particles are generated in the grid cells corresponding to a mixing surface. They are defined by position, velocity, radius, and concentration. The number of particles that are reseeded depends on the maximum that has been set for each cell. That is, we simulate more particles with reseeding to have \( \text{max}_p \) particles in a mixing surface cell if the number of particles is less than \( \text{max}_p \). \( \text{max}_p \) is the maximum number of particles that a cell can have. In our experiment, we set \( \text{max}_p \) to four or eight. When a particle is seeded, its concentration is initialized to \( C_{\text{equilibrium}}/\text{max}_p \).

When one or more concentration particles exist in a cell, absorption occurs from the particles to the grid. The reaction rate is proportional to the concentration of the substance. In general, the reaction rate of two substances is \( k_{\text{reaction}} |A| |B| \), where \(|A|\) is the sum of the particle concentrations existing...
in each cell ($\sum_{\text{particle}} C_{\text{particle}}$), and $[B]$ is the concentration of solvent in the cell.

In our simulation, we limit the maximum number of cell concentrations to $C_{\text{equilibrium}}$, and we assume that the concentration of solvent $[B]$ is initialized to $C_{\text{equilibrium}}$. Because the concentration of substance $[B]$ is proportionally reduced as reaction product is generated, we can model $[B]$ as $(C_{\text{equilibrium}} - C_{\text{cell}})$. Thus, the cell concentration of reaction product absorbed from a particle for the next time step, $C_{\text{cell}}^{n+1}$ is:

$$C_{\text{cell}}^{n+1} = C_{\text{cell}}^{n} + \sum_{\text{particle}} k_{\text{reaction}} C_{\text{particle}}^{n} (C_{\text{equilibrium}} - C_{\text{cell}}^{n}),$$

(18)

where $C_{\text{cell}}$ is the concentration of product in the cell, $C_{\text{particle}}$ is the $i$th particle’s concentration in the cell, $np$ is the number of particles in the cell, and $k_{\text{reaction}}$ is a coefficient that defines the rate of the chemical reaction. In our experiment, we use a small number, typically 0.001, for $k_{\text{reaction}}$ and set $C_{\text{equilibrium}}$ to 1.0.

When a concentration particle is absorbed into the cell, the particle concentration of the solute is reduced:

$$C_{\text{particle}}^{n+1} = C_{\text{particle}}^{n} - k_{\text{reaction}} C_{\text{particle}}^{n} (C_{\text{equilibrium}} - C_{\text{cell}}^{n}).$$

(19)

Depending on Eq. (19), if a particle’s concentration is less than a fixed threshold, which is 0.01, the particle is deleted. If $C_{\text{cell}}$ is larger than $C_{\text{equilibrium}}$, the concentration of product decreases and the particle concentration of solute increases in the next step. The process is similar to maintaining a state of chemical equilibrium. This allows the computation to be simplified without a significant effect on accuracy.

### 7.2. Concentration Particle Advection

Concentration particles are advected by their own velocity according to a Lagrangian method. Each of their positions is updated using the equation:

$$p_{\text{particle}}^{n+1} = p_{\text{particle}}^{n} + u_{\text{particle}}^{n} \cdot \Delta t.$$

The velocity of a particle $u_{\text{particle}}$ is calculated using the equation:

$$u_{\text{particle}}^{n+1} = u_{\text{particle}}^{n} + f_{\text{advection}} \cdot \Delta t.$$

The advection of concentration particles depends on the intermolecular diffusion, infiltration, and coupling forces:

$$f_{\text{advection}} = f_{\text{diffusion}} + f_{\text{capillary}} + f_{\text{molecular}} + f_{\text{coupling}}.$$

(20)

Concentration particles spread out in the direction of the concentration gradient. The diffusive flux follows Fick’s Law:

$$f_{\text{diffusion}} = -D \nabla C_{\text{cell}},$$

(21)

where $D$ is the diffusion coefficient. It is not easy to model complicated diffusion scenarios in which irregular filaments are produced and spread out, using only the above Eq. (21). Thus, we use a capillary force and intermolecular forces.

The capillary force is modeled using the gradient of curvature of the concentration field. Because regions of high curvature have a high porosity, it is easy for a fluid to infiltrate in that region. We therefore model the capillary force as follows:

$$f_{\text{capillary}} = I |\kappa_{\text{concentration}}| \nabla \kappa_{\text{concentration}},$$

(22)
where \( \kappa_{\text{concentration}} \) is the curvature of the concentration field and \( I \) is the infiltration coefficient. The value of \( \kappa_{\text{concentration}} \) can be expressed as

\[
\kappa_{\text{concentration}} = \nabla \cdot \left( \frac{\nabla C_{\text{cell}}}{\| \nabla C_{\text{cell}} \|} \right).
\]

We simulate intermolecular forces using the SPH method. The intermolecular forces themselves are modeled by the interaction forces between concentration particles. The following equation, which is typically used to calculate pressure in SPH-based fluid simulations, has been found experimentally to be effective in modeling the attractive force between concentration particles:

\[
f_{\text{molecular}} = A \sum_j m_j \left( \frac{P_i}{C_{\text{particle},i}} \chi + \frac{P_j}{C_{\text{particle},j}} \chi \right) \nabla W(x_j - x_i),
\]

(23)

where the pressure \( P_i = \alpha C_{\text{particle},i} \), with the control parameter \( \alpha \), \( A \) is the attraction coefficient and \( m \) is the mass of a particle. We assume that the mass of all the particles is the same, with a value of 1.

When a concentration particle moves between different fluids, a resistance force is provided by the fluid into which the particle is moving. This force causes the coupling between the velocity of grid and particle. The magnitude of this force is proportional to the relative velocity of the concentration particle and the receiving fluid, and can be expressed as:

\[
f_{\text{coupling}} = -R (u_{\text{particle}} - u_{\text{cell}}).
\]

(24)

where \( R \) is the resistance coefficient. The overall advection of the particles is provided by the sum of the forces in Eqs. (21),(22),(23) and (24).

8. Results

Simulations were performed on an Intel PC with a 3.0 GHz CPU. In general, simulation grids are economically implemented as octrees, but we must use a regular grid for scenarios where concentration must be modeled. In this case, the value of the level-set is less than 0. In other words, we can simulate concentrations where liquid is present without ambient air. Therefore, if \( \phi \) over the whole level-set is more than 0, we could use an octree data structure instead of the dense regular grid that we used where \( \phi \) is less than 0.

Because an octree data structure is being used, the computation cost increases in proportion to the volume of fluid that is simulated. The simulation shown in figure 5 takes at least 15 seconds per frame and at most 120 seconds. For rendering, we used Mental-Ray in Maya. Simulation data are saved as a mesh and a scalar field to be rendered in Maya.

Figure 7 shows that the Venus-shaped volume of fluid dives into an open water surface and is mixed. The two different fluids intermingle and spread out just as ink mixes with water. This experiment was performed using a 256 \( \times \) 256 \( \times \) 128 grid. The simulations took approximately 9 minutes per frame, including running the fluid solver and the file I/O.

Figure 8 shows how a solid teapot shape dissolves in a liquid. As the volume of the solid is reduced by mass transfer, the solid teapot dissolves in water. Violet material from the teapot that has been mixed in water floats on the surface and interacts with air bubbles. Figure 9 shows a somewhat
different scenario in which a liquid teapot mixes with the water that surrounds it. In contrast to Figure 8, viscous fingering now takes place because of the difference in pressure at the interface between the two fluids and a liquid core itself flows. We intentionally inserted bubbles to show the comparison between the solid and the liquid when interacting with bubbles. We can thus see the complicated phenomena of liquids mixing. The grid for the simulations shown in Figures 8 and 9 was 128$^3$. The average simulation time per frame was 120 seconds.

Figure 10 presents a dramatic scenario in which a liquid teapot and a rigid teapot are dropped into water one by one. First, we show how they react when they are of the same type of substance, clear water. Then, the insoluble object drops, and after it reaches the bottom of the water, we make that teapot blue ink. It slowly dissolves and mixes with the water. The grid for this simulation was 128$^3$, and the average simulation time per frame was 100 seconds.

9. Conclusions

We have described a technique for modeling the flow of miscible multi-phase fluids by improving the handling of interfacial properties and chemical reactions. In several experiments, we constructed naturalistic scenarios in which a solid body melts or liquids are mixed. These combinations of viscous fingering, chemical-based mass transfer, and molecular forces are relatively easy to model with techniques familiar to the computer graphics community.

In the future, we aim to simulate smaller-scale features such as the filaments that appear when ink spreads out in water. We will simulate the liquid-solid reaction, based on accurate chemical laws such as solution and solidification with a state of chemical equilibrium.

Acknowledgements

This research was supported by Basic Science Research Program WCU(World Class University) program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (No.2009-0083827, R31-2008-000-10008-0). This research was supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2009.

References


© 2009 The Author(s)
Journal compilation © 2009 The Eurographics Association and Blackwell Publishing Ltd.