

PHYSICS-BASED SIMULATIONS FOR FLUID MIXTURES

by

Dongwoon Lee

A Research Paper submitted in conformity with the requirements  
for the degree of Doctor of Philosophy  
Graduate Department of Computer Science  
University of Toronto

Copyright © 2007 by Dongwoon Lee

# Abstract

Physics-Based Simulations for Fluid Mixtures

Dongwoon Lee

Doctor of Philosophy

Graduate Department of Computer Science

University of Toronto

2007

We develop a physics-based particle method to simulate various fluid mixture effects. A particle method is chosen over a grid-based method because we believe a particle method is more intuitive and better suited for handling physical and chemical interactions between fluids. Since a particle carries the physical attributes of the fluid (e.g, density and velocity), transfer of these attributes between particles can be computed easily using a particle method. Also, arbitrary shapes of the fluid mixture can be modeled based on the particles' spatial configuration, which is updated throughout the simulation.

# Acknowledgements

First and foremost, praise and thanks goes to my savior, Jesus Christ for everything in my life.

I cannot overstate my gratitude to my supervisor, Dr. Kenneth Jackson for his guidance, encouragement and tremendous patience during the course of this research project. He has helped me to complete this research paper and prepare for future challenges. Without him, it would have not been possible at all. I would also like to thank my co-supervisor, Dr. Eugene Fiume, who motivated me throughout this project and provided helpful comments and suggestions.

I wish to express thanks to Dr. David Steinman for teaching his inspiring course, MIE418 Fluid Mechanics II. It was a great opportunity for me to obtain the fundamental knowledge of fluids. He helped me to understand dynamics of fluids through his lectures and experiments. I also wish to thank my brother-in-law, Dr. Jaegyun Park for introducing me to many interesting topics in physics and mechanics, and discussing them with me.

I would like to thank my colleagues, Joseph Laszlo, Jingrui Zhang, Po-Feng Paul Yang and Irene Fung for helping me to get through the difficult times and giving helpful advice concerning graduate studies in Toronto. I am also very grateful to Sooam Lee, Sangrok Yu and Seungwook Ma and all of my friends for their emotional support and spiritual friendship.

Lastly, I wish to thank my entire family, especially my parents, for all their support and confidence in me. I would like to thank my lovely wife, Susan and my newborn daughter, Chloe, for providing a loving environments and their encouragement.

Thank you.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Contributions . . . . .	2
1.2.1	Diffusion . . . . .	2
1.2.2	Chemical Reaction . . . . .	2
1.3	Overview of Paper . . . . .	3
<b>2</b>	<b>Related Work</b>	<b>4</b>
2.1	Particle-Based Simulation . . . . .	4
2.1.1	Deformable Objects . . . . .	4
2.1.2	Newtonian Fluid . . . . .	5
2.1.3	Non-Newtonian Fluid . . . . .	6
2.1.4	Granular Materials . . . . .	7
2.2	Grid-Based Simulation . . . . .	7
2.2.1	Incompressible Fluids . . . . .	7
2.2.2	Fluid Interaction . . . . .	8
<b>3</b>	<b>Fluid Models</b>	<b>10</b>
3.1	Fluid Dynamics . . . . .	10
3.1.1	Conservation of Mass . . . . .	12
3.1.2	Conservation of Momentum . . . . .	12

3.1.3	Eulerian versus Lagrangian Methods . . . . .	13
3.2	Multifluid Systems . . . . .	14
3.2.1	Diffusion . . . . .	15
3.2.2	Chemical Reactions . . . . .	16
3.2.3	Non-Isothermal Conditions . . . . .	16
<b>4</b>	<b>SPH Based Fluid Simulation</b>	<b>18</b>
4.1	Smoothed Particle Hydrodynamics . . . . .	19
4.2	SPH Method for Fluid Dynamics . . . . .	21
4.2.1	Pressure . . . . .	22
4.2.2	Viscosity . . . . .	23
4.2.3	External Force . . . . .	24
4.3	Smoothing Kernels . . . . .	24
4.4	Boundary Treatments . . . . .	26
<b>5</b>	<b>Simulation for the Fluid Mixture</b>	<b>28</b>
5.1	Fluid Mixture . . . . .	28
5.2	Diffusion . . . . .	29
5.2.1	Concentration . . . . .	29
5.2.2	Diffusive Force . . . . .	29
5.3	Chemical Reaction . . . . .	30
5.4	Non-Isothermal Conditions . . . . .	32
5.4.1	Temperature Field . . . . .	32
5.4.2	Thermal Viscosity . . . . .	33
5.5	Immiscible Fluids . . . . .	33
<b>6</b>	<b>Implementation Issues</b>	<b>35</b>
6.1	Neighbor Search . . . . .	35
6.2	Simulation Steps . . . . .	36

6.3	Surface Reconstruction . . . . .	37
<b>7</b>	<b>Results</b>	<b>39</b>
7.1	Animation . . . . .	39
7.1.1	Diffusion . . . . .	40
7.1.2	Chemical Reaction . . . . .	40
7.2	Integration Methods . . . . .	40
<b>8</b>	<b>Conclusion and Future Work</b>	<b>46</b>
	<b>Bibliography</b>	<b>48</b>

# List of Figures

4.1	Particle approximation . . . . .	20
4.2	The smoothing kernels . . . . .	26
4.3	Particle approximation on the boundary . . . . .	26
4.4	Ghost particles . . . . .	27
6.1	Surface reconstruction . . . . .	37
7.1	Mixture of a single fluid in the water (side view) . . . . .	41
7.2	Mixture of a single fluid in the water without mixing (top view) . . . . .	42
7.3	Mixture of a single fluid in the water with mixing (top view) . . . . .	42
7.4	Mixture of two fluids in the water (top view) . . . . .	43
7.5	Chemical reaction (top view) . . . . .	43
7.6	Time integration . . . . .	44

# Chapter 1

## Introduction

### 1.1 Motivation

Fluids are very common substances that we often observe and are familiar with in our daily lives. Thus, simulating fluids is an important topic in graphics research which focuses on simulating the real world. However, since fluids are too complicated to model realistically using any traditional key-frame animation techniques, many researchers have turned to physics-based models for visually convincing animation results. Physics-based models include various physical parameters and depend on a set of natural physical principles. Specifically, fluid models are based on fluid dynamics principles (e.g., Navier-Stokes equations) that have been well established in the CFD (Computational Fluid Dynamics) community.

In recent years, many techniques have been proposed for a wide range of fluid effect problems: tracking free surfaces, interplay between fluids and solid objects and modeling various types of fluids. However, relatively little attention has been paid to multiple fluids problems. In such problems, two or more fluids interact with each other and their complex behaviors are simulated using physics laws to model mixing effects. For example, when black ink is poured into a glass of water, both the ink and the water interact and

mix in complex patterns. On the other hand, oil and water never mix at all. Thus, if different types of fluids are introduced to mix, many different phenomena can occur depending on the physical properties of these fluids.

In this thesis, we develop a physics-based particle method to simulate these fluid effects. A particle method is chosen over a grid-based method because it is more intuitive and appropriate for handling physical and chemical interactions between fluids. Since a particle carries the physical attributes of the fluid (e.g., density and velocity), interchanges and transfers of these attributes between particles can be easily computed using a particle method. Also, arbitrary shapes of the fluid mixture can be recovered based on the particles' spatial configuration, which is updated throughout the simulation.

## 1.2 Contributions

For modeling the fluid mixture, we designed the following two procedures to be incorporated into an existing particle-based fluid solver: diffusion and chemical reaction.

### 1.2.1 Diffusion

To model diffusion in the fluid mixture, we introduce mass concentrations, which are computed together with the fluid equations. They are represented by an additional scalar field to model the spatial proportions that fluids occupy in certain areas. These concentrations guide the direction of the diffusion and determine local shapes of the mixture where multiple fluids meet.

### 1.2.2 Chemical Reaction

Diffusion plays a role only in distributing and spreading fluids in appropriate directions. Chemical reactions determine what kind of effects can happen in accordance with the fluid properties while diffusion proceeds. For chemical reactions, we consider the following

effects: dissolution, fluid generation or transition, heat generation or absorption. When heat is generated or absorbed during the simulation, heat transfer must be modeled accurately to handle non-isothermal conditions properly. We simulate this process using the diffusion equation, in a manner similar to that described in [22, 28].

### 1.3 Overview of Paper

In Chapter 2, we review background and previous work related to our research problem. In Chapter 3, we describe basic mathematics to represent the physics and dynamics of fluids. In Chapter 4, we explain how the fluid problem can be solved numerically by the SPH method, a particle method. In Chapter 5, we extend our approach to handle complex mixture problems occurring in the fluid mixtures. In Chapter 6, we provide a brief overview of the implementation of our method. In Chapter 7, we discuss experimental results and efficiency issues. In Chapter 9, we conclude with a brief summary of the key results in the thesis and a discussion of future work.

# Chapter 2

## Related Work

In this chapter, we briefly review previous work relevant to our research. As there are many fluid simulation problems with different characteristics, many numerical techniques have been developed to model them. There are two main distinct approaches for modeling fluid flow problems: particle based and grid based methods. Thus we classify the related work outlined below for fluid flow problems according to whether they are modeled by particle based or grid based methods.

### 2.1 Particle-Based Simulation

Since a particle system was first introduced in graphics by Reeves [25] to animate irregular fuzzy objects, particle based methods have been widely used to solve many graphics problems thanks to the flexibility and computational simplicity of this approach.

#### 2.1.1 Deformable Objects

Tonnesen [29] used a particle system to construct a deformable surface. Particle interactions are based on the Lennard-Jones potential force. Their interactions are then augmented with a set of geometrical potentials to simulate deformation. Also particle

based methods can be extended to handle thermal dynamics problems, such as melting, by including heat transfer. Desbrun et al. [5] introduced the Smoothed Particle Hydrodynamics (SPH) method to the graphics community for simulating highly deformable bodies. The SPH method allows a particle system to represent deformation with more physical accuracy than other particle based methods that were used earlier. They demonstrated various behaviors of inelastic deformable bodies.

### 2.1.2 Newtonian Fluid

The SPH method was adopted by Stora et al. [28] to simulate fluids, particularly, lava flows. The mechanical features of lava are modeled by a temperature dependent viscosity that evolves with the heat transfer during the simulation. Müller et al. [20] developed a viscous water simulation using the SPH method, in which the basic fluid motion is governed by the Navier-Stokes equations. They showed that their water model performs well for a realtime application. Their model was further extended in [22] to a multiple fluids simulation in which fluid-fluid interactions are taken into account. They simulated interactions between fluids by including interface tension forces. They also considered temperature diffusion to visualize the phase changes of the fluid. Since the last two references are most relevant to our work, we discuss them in more detail in Chapter 4. Müller et al. proposed in [21] a solid-fluid interaction model, which couples a particle based fluid solver with an elastic solid simulator formulated via the finite element method (FEM). Solid objects represented by a polygonal mesh are re-sampled by a number of particles on their surfaces. Since both computational domains are represented by particles, this type of interaction problem can be solved by particle-particle interactions. Since incompressibility is inherently difficult to enforce in a particle method, a fluid modeled by the SPH method behaves like a compressible fluid. This deficiency was addressed by Premoze et al. [23] who used a semi-implicit method, the Moving-Particle Semi-Implicit (MPS) method. In the MPS method, incompressibility is enforced to

achieve constant density. Also, particle convection is computed via an Eulerian scheme for handling inflow and outflow problems.

### 2.1.3 Non-Newtonian Fluid

While the aforementioned fluid models are based on a Newtonian fluid, other researchers have studied non-Newtonian fluid simulations. Steele et al. [26] presented a particle based viscous liquid simulation. They modeled a liquid motion by approximating three forces: adhesion, viscosity and friction. Adhesion forces are expressed in the form of a matrix, the entries of which contain the interaction functions of the materials and liquid types. Volume preservation is obtained by correcting density errors iteratively. However, their model is restricted to highly viscous liquids because it uses a linear density kernel and no-penetration is strictly constrained. Clavet et al. [4] relaxed this restriction by introducing viscoelastic behaviors into the fluid simulation. They proposed a novel method, double density relaxation, to ensure the incompressibility and anti-clustering in the fluid simulation. With this procedure, the smooth liquid surface can be animated and realistic surface tension effects can be obtained. Viscoelastic behavior is modeled by inserting virtual springs between particles and different viscoelastic fluid properties are modeled by adjusting the spring rest lengths. They also incorporated into their fluid model a rigid body simulator to enable a two-way interaction between the fluid and other objects. Mao et al. [19] introduced a nonlinear function of the stress tensor, a co-rotational Maxwell model, for simulating a non-Newtonian fluid. Since the rotational frame indifference is used in the stress tensor computation, their model produces more accurate rotational fluid motions. They used a semi-implicit scheme to compute pressure to enforce incompressibility. Also they reduced the particle clustering problem by re-sampling a distribution of particles over the simulation.

### 2.1.4 Granular Materials

Bell et al. [1] simulated granular materials, such as sand and grains, using molecular dynamics. They considered each grain as a rigid object and computed their interactions based on the molecular forces. Interactions are modeled by handling contact forces between particles, which have been neglected in most particle-based fluid models. They also extended their model to handle the interaction between particles and rigid bodies, by placing particles onto the rigid bodies and solving with a rigid body simulator. Zhu et al. [30] showed that only small modifications are needed to convert an existing fluid solver to a sand simulator. They combined a particle based method with a grid based one for the simulation. While particles are used for accurate surface tracking and advection, the incompressibility and boundary conditions are enforced by the grid method.

## 2.2 Grid-Based Simulation

As an alternative to particle based methods, grid based methods have been very successful for simulating incompressible fluids.

### 2.2.1 Incompressible Fluids

Foster et al. [8] first utilized work done in the computational fluid dynamics community to achieve realistic animation of liquids. Physical quantities associated with fluids are directly derived from the solution of the full Navier-Stokes equations. To compute the fluid motion, they used a staggered grid, on which the velocity is computed at each face of a grid cell while the pressure is computed at the center of a cell. Additional particles (i.e., marker particles) are introduced to track the free surface of the liquid. They also proposed in [9] a gas model in which the turbulent rotational motions are animated and the gas interacts with the surrounding environment. The heat equation is included to simulate the thermal buoyancy effect. However, their fluid simulator is numerically limited by

its small time-steps because it is based on an explicit integration method. Stam [27] improved this approach by using a semi-Lagrangian scheme, which allows significantly larger time-steps without losing numerical stability. Since a first-order integrator is used, this method suffers from numerical dissipation which makes small vortices vanish rapidly. Fedkiw et al. [7] developed a gas model using the inviscid Euler equations rather than the viscous full Navier-Stokes equations. They reduced the numerical dissipation inherent in semi-Lagrangian schemes by adding vorticity confinement in the simulation step. This additional vorticity force enables their method to represent small scale rolling features of smoke and it helps animate visually realistic smoke. Foster et al. [10] presented a practical method for simulating and animating liquids. They observed that numerical dissipation could result in visually unacceptable surface features in the liquid simulation. They corrected this problem by introducing a hybrid method that uses inertialess particles and an implicit surface. These particles allow liquid to splash freely. Also they proposed a control method for the liquid motion by treating boundary conditions between moving objects and liquid. Later, Enright et al. [6] made further enhancements for animating and rendering the liquid surface. They introduced the particle level set method, which uses massless marker particles and a dynamic implicit surface. This method focuses on modeling the surface, not the liquid volume, as was done in previous work [10]. By extrapolating the velocities across the surface, more accurate computation and better control is obtained. Carlson et al. [2] simulated material dynamics, such as melting and solidifying, by using the existing fluid model. The heat diffusion equation is coupled with the viscosity of the fluid. They used an implicit Euler method to compute diffusion. This allows for the stable integration of highly viscous systems.

### 2.2.2 Fluid Interaction

Along with the fluid models, which have proved successful for animating fluid motions, many researchers have considered other problems, such as how to deal with fluid-fluid or

fluid-solid interactions. Génevaux et al. [11] integrated existing grid based fluid simulators with Lagrangian solid dynamics simulators. They coupled two separate simulators: a grid based fluid simulator and a particle based deformable solid simulator. To do so, they formulated interaction rules between marker particles (from the fluid) and mass particles (from the solid). Their method is capable of modeling interacting motions between the fluid and the solid, such as floating, sinking and splashing. Carson et al. [3] presented a better coupling method, which can allow the fluid to interact with a rigid object and permitted more complex object shapes. They first rasterize rigid objects and treat their velocities as if they were made of the fluid. Those velocities are updated by using the fluid solver and then modified by incorporating any collision and buoyancy forces. Finally they are constrained to be a rigid motion by enforcing rigidity conditions. Losasso et al. [17] simulated interactions between multiple liquids. They use a separate particle level set method to represent each region from multiple liquid interactions. Those regions can be any type of liquids with differing viscosities, densities and viscoelastic properties. They solved the numerical difficulties associated with using multiple level sets by incorporating physical jump conditions to model discontinuities across region boundaries. Ihm et al. [13] exploited the theory of chemical kinetics to animate chemical reactions of a gaseous fluid. While the gas mixture's velocities are computed using an existing fluid solver, its density and temperature are evolved using an advection-diffusion scheme. The densities of each substance in the mixture are updated according to the defined reaction functions. They demonstrated various chemical reaction effects, such as explosions and catalytic effects.

# Chapter 3

## Fluid Models

In this chapter, we describe basic mathematics to represent the dynamics and physical properties of fluids. The governing equations for the fluid are derived from Newton's physics laws: conservation of mass, momentum and energy. The Navier-Stokes equations are a popular approach for modeling various fluid problems and applications. However, they are often insufficient to model more complex problems, such as those that arise in multifluid systems. Thus, it is necessary to supplement the Navier-Stokes equations with constitutive relations describing physical interactions or chemical reactions.

We begin with a brief introduction to the Navier-Stokes equations and then extend them to solve multifluid problems.

### 3.1 Fluid Dynamics

We begin by considering a simple fluid problem. To start, we assume that the fluid is

- continuous
- Newtonian
- incompressible

- isothermal
- uniform and homogeneous

A fluid is a continuous object whose physical properties can be described by certain continuous functions of space and time. This is a necessary condition to formulate conservation laws for physical quantities such as mass and momentum. If the viscosity is always constant at all shear rates, a fluid is called Newtonian. For example, water is a Newtonian fluid. On the other hand, a non-Newtonian fluid is a fluid for which the viscosity changes under an applied strain rate, like toothpaste or ketchup. Also we only consider fluid systems for which the density and the temperature remain constant. To begin, we consider a single fluid only; multifluid systems are discussed in Section 3.2.

Given these assumptions, the basic equations of the fluid dynamics are derived based on the following fundamental physics laws:

- conservation of mass
- conservation of momentum
- conservation of energy

The conservation of mass law states that no mass is created or lost during the simulation. The conservation of momentum law implies that a fluid particle's acceleration is proportional to all the forces acting on it. These two conservation laws are sufficient to determine the fluid motion by specifying the trajectory of each particle. The conservation of energy law states that the sum of the kinetic energy and the internal potential energy remains constant within the system. Since in an isothermal fluid system no heat is introduced or generated, the internal energy depends solely on the kinetic energy. Thus, if we neglect any volume compression (or expansion) and energy dissipation, the potential energy remains unchanged. That is, conservation of energy is guaranteed under these assumptions.

### 3.1.1 Conservation of Mass

The equation for conservation of mass is derived by considering a fixed control surface through which the fluid flows. The flux of mass through the control surface must equal the decrease of mass within the volume:

$$\int_S \rho \mathbf{u} \cdot \mathbf{n} dA = - \int_V \frac{\partial \rho}{\partial t} dV \quad (3.1)$$

where  $S$  is the control surface,  $V$  is the control volume,  $\mathbf{n}$  is the surface normal,  $\mathbf{u}$  is the velocity and  $\rho$  is the density. By Gauss' theorem,

$$\int_S \rho \mathbf{u} \cdot \mathbf{n} dA = \int_V \nabla \cdot \rho \mathbf{u} dV \quad (3.2)$$

Substituting (3.2) into (3.1) and rearranging terms, we get

$$\int_V \left( \frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} \right) dV = 0 \quad (3.3)$$

Since this is true for every control volume,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0 \quad (3.4)$$

For an incompressible fluid whose density  $\rho$  is constant, (3.4) reduces to

$$\nabla \cdot \mathbf{u} = 0 \quad (3.5)$$

This is the continuity equation for an incompressible fluid.

### 3.1.2 Conservation of Momentum

Using Newton's second law, we can formulate the equation for momentum conservation as

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} = \mathbf{f}^{body} + \mathbf{f}^{surface} \quad (3.6)$$

The net force consists of the body force and the surface force which act on the fluid particle. The body force is proportional to the volume of the fluid particle. Examples

include the gravitational force and the electrical force. They exert an external force on the fluid particle. Here, we consider the gravitational force only,

$$\mathbf{f}^{body} = \rho \mathbf{g} \quad (3.7)$$

where  $\mathbf{g}$  is the gravitational acceleration. The surface force is the force which is exerted on the fluid surface, and is physically associated with pressure and external stresses. Since the surface force is derived from the stress tensor, by expanding (3.6) using the stress tensor formula described in [24], we get

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g} \quad (3.8)$$

where the density is constant and  $\mu$  is the viscosity coefficient of a Newtonian fluid. This equation together with (3.4) are called the Navier-Stokes equations for the incompressible fluid.

### 3.1.3 Eulerian versus Lagrangian Methods

There exist two main categories of computational methods for fluid dynamics: Eulerian methods and Lagrangian methods. Eulerian methods are based on a fixed mesh grid on which the fluid equations are discretized. Every physical property (e.g., density and velocity) are sampled and computed at these fixed points. On the other hand, Lagrangian methods usually discretize the fluid using a moving mesh grid. In contrast to Eulerian methods, by directly following the trajectory of the fluid particles, Lagrangian methods track the fluid. Thus, Eulerian methods are based on the partial derivative  $\frac{\partial}{\partial t}$  and Lagrangian methods are based on the total derivative  $\frac{d}{dt}$ . Equation (3.8) (which is given in Eulerian form) can be converted to Lagrangian form using the material derivative

$$\frac{d}{dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla \quad (3.9)$$

Using (3.9), we rewrite (3.8) without a convective term as

$$\rho \frac{d\mathbf{u}}{dt} = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g} \quad (3.10)$$

In an Eulerian method, it is difficult to track the fluid motion without numerical errors, which may result in numerical dissipation and damping. Furthermore, the numerical accuracy depends on the grid resolution. Higher resolution of the grid gives better accuracy, but increases the computational time and memory requirements. However, the incompressibility condition is easily enforced using an Eulerian method. In a Lagrangian method, tracking the fluid particles comes naturally. Thus, as long as the number of particles and their mass remain constant, conservation of mass and the continuity condition are easily enforced. However, it is hard to obtain incompressibility in the fluid motion and to solve large deformation and turbulence modeling problems.

## 3.2 Multifluid Systems

So far, we have considered only a simple fluid system in which there exists one fluid without any physical interactions or chemical reactions. Now we extend our simple model to a more complex system by reducing the assumptions we made in the previous section. Our new fluid system may have

- different types of fluids
- physical interactions
- chemical reactions
- non-isothermal properties

If different types of fluids having different physical properties, such as density and viscosity, are considered, they can interact in various ways based on their properties. Mixing can occur at the interface between the fluids, such as occurs with coffee and milk. In some cases, such as oil and water, the fluids never mix at all. Sometimes fluids can experience a chemical reaction which produces a new type of fluid or generates heat energy. When heat energy is added or internally generated, there may be temperature gradients in the

fluid system. The temperature gradients may be converted to kinetic energy which may lead to some dynamic motion in the fluid.

### 3.2.1 Diffusion

Only diffusive mixing is considered in our system. This diffusion is a physical phenomena associated with mass transport. The diffusion occurs when there is a spatial gradient associated with some proportions of the fluid mixture. This is termed a concentration gradient. For example, consider black ink drops in water. When ink is injected into water, from its injection point where ink concentration is highest, it is diffused outward to regions where there is less or no concentration of ink . This process continues until an equilibrium state is reached in terms of the mixture proportions. Besides mass diffusion, many gradients in other physical properties, such as temperature, pressure and electric fields, can cause mixing diffusion, but they are not discussed here.

There are many ways of defining concentration in a fluid mixture. The two basic approaches are mass concentration (density) and molar concentration, which are defined, respectively, by

$$\begin{aligned}\rho^i &= \frac{m^i}{V} = \frac{\text{mass of species } i}{\text{volume of solution}} \quad (kg \ m^{-3}) \\ c^i &= \frac{n^i}{V} = \frac{\text{number of moles of species } i}{\text{volume of solution}} \quad (mol \ m^{-3})\end{aligned}$$

Two related dimensionless concentrations are the mass fraction and molar fraction, which are defined, respectively, by

$$\begin{aligned}w^i &= \frac{\rho^i}{\rho} = \frac{\text{mass concentration of species } i}{\text{density of solution}} \\ x^i &= \frac{c^i}{c} = \frac{\text{molar concentration of species } i}{\text{molar concentration}}\end{aligned}$$

We use mass fraction concentration  $w^i$  to model the mixing diffusion. The mixing is driven by the diffusive force, which is strongly dependent on the concentration gradient

and increases with temperature. The diffusive force is

$$\mathbf{f}^i = -RT\nabla w^i \quad (3.11)$$

where  $R$  is the gas constant and  $T$  is the temperature. The minus sign indicates that the diffusion proceeds in the direction from higher to lower concentration. This force can be applied to the fluid equations as an additional body force which acts on the fluid particle in much the same way as the gravitational force does.

### 3.2.2 Chemical Reactions

A chemical reaction is a process that results in the conversion of chemical substances or the changes of chemical structure. This process involves the motion of electrons, which cause chemical bonds to break or form. Generation of new substances or heat energy are often associated with chemical reactions. We consider only simple chemical reactions here, such as



where  $A, B, E, F$  are chemical substances and  $a, b, e, f$  are their stoichiometric coefficients. Such a chemical reaction can be described by a differential equation:

$$\frac{d[A]}{dt} = -a \cdot r, \quad \frac{d[B]}{dt} = -b \cdot r, \quad \frac{d[E]}{dt} = e \cdot r, \quad \frac{d[F]}{dt} = f \cdot r \quad (3.13)$$

where  $[ ]$  denotes the molar concentration and  $r$  is the rate function which depends on time and temperature. A heat energy term can be added on either side of (3.12), which indicates that heat is released (exothermic) or heat is absorbed (endothermic) during the chemical reaction. The heat energy change is associated with a temperature change which can influence the fluid motion. The reverse reaction is ignored here.

### 3.2.3 Non-Isothermal Conditions

A non-isothermal condition prevails when the temperature changes due to heat generation or addition. For example, in a chemical reaction, the heat transfer or viscous dissipation

can lead to heat generation within the fluid. In such cases, the fluid is no longer isothermal and the fluid equations must be supplemented by another equation enforcing conservation of energy. The energy equation, which is associated with temperature, is

$$\rho c_p \frac{dT}{dt} = T\beta \frac{dp}{dt} + k\nabla^2 T + \Phi + \rho r \quad (3.14)$$

where  $c_p$  is the specific heat at constant pressure,  $T$  is the temperature,  $\beta$  is the thermal expansivity,  $k$  is the thermal conductivity,  $\Phi$  is the viscous dissipation and  $r$  is the heat source. If the fluid is incompressible, the pressure change can be negligibly small even if the density varies with the temperature. Also the dissipation term can be neglected so long as the viscous heating is small compared with the conductive heat flow from the temperature gradient. Under these assumptions and taking other fluid properties to be constant, the heat equation (3.14) reduces to

$$\frac{dT}{dt} = k'\nabla^2 T + \text{heat source} \quad (3.15)$$

where  $k'$  is the modified thermal conductivity. For a low density fluid, such as an ideal gas, the thermodynamics can be completed by adding physical properties, such as

$$pV = nRT \quad (3.16)$$

where  $n$  is number of moles and  $R$  is the gas constant. Thus the density can be approximated by

$$\rho = \rho_0[1 - \alpha(T - T_0) + \beta(p - p_0)] \quad (3.17)$$

where  $\rho_0$ ,  $T_0$  and  $p_0$  are the reference values of the density, temperature and pressure, respectively.

# Chapter 4

## SPH Based Fluid Simulation

In this chapter, we briefly review the Smoothed Particle Hydrodynamics (SPH) method, the Lagrangian particle method that we use as our numerical method for approximating fluid flows. A more detailed description of the SPH method can be found in [15].

As discussed in Chapter 3, although both Eulerian and Lagrangian methods generally work well for simulating fluids, one method often outperforms the other for a particular problem or application. For fluid mixtures, where multiple fluids coexist and interact, their mixing interaction (e.g., diffusion and reaction) must be physically modeled and mixture boundaries must be tracked accordingly. We emphasize modeling irregular surface geometry coming from simulating fluid mixtures. Lagrangian particles carry their own physical quantities, such as mass and velocity, which are directly related to physical interactions. Thus, using Lagrangian methods, these interactions can be easily modeled. This is the main reason we choose to base our simulation on a Lagrangian particle method rather than an Eulerian grid method.

We begin by describing the SPH method and our approach to solving the Navier-Stokes equation described in Section 3.1 using the SPH formulation. We then address boundary problems which are inherent in particle methods and discuss our approach to handling them.

## 4.1 Smoothed Particle Hydrodynamics

Smoothed particle hydrodynamics (SPH) was originally developed for modeling astrophysical phenomena by Lucy [18] and Gingold and Monaghan [12]. Later it was extended to solve many dynamic material problems and fluid dynamics problems. Thus, it has been successfully used in a wide range of areas, including soft object deformation, free-surface tracking, shockwave, underwater explosion and hydrodynamic problems.

We introduce the fundamentals for the SPH method and discuss how it is applied to fluid problems. In contrast to the Grid based Eulerian method, the SPH method is a purely meshless Lagrangian method. The SPH method consists of the following key steps. First, the SPH method discretizes the Partial Differential Equations (PDEs) for the field variables such as density, velocity and energy, associated with the problem. This produces Ordinary Differential Equations (ODEs) that approximate the field variables with the integral representation. Second, this integral representation of the field variables is further approximated with a set of particles (i.e, the particle approximation) because the problem domain of the SPH method is discretized by randomly distributed particles.

A function of a field variable is based on the following integral representation:

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}')\delta(\mathbf{x} - \mathbf{x}')d\mathbf{x}'$$

where  $\mathbf{x}$  is a three dimensional position vector and  $\delta(\mathbf{x} - \mathbf{x}')$  is the Dirac delta function given by

$$\delta(\mathbf{x} - \mathbf{x}') = \begin{cases} \infty & \mathbf{x} = \mathbf{x}' \\ 0 & \mathbf{x} \neq \mathbf{x}' \end{cases}$$

$$\int_{\Omega} \delta(\mathbf{x} - \mathbf{x}')d\mathbf{x}' = 1$$

If the Dirac delta function  $\delta(\mathbf{x} - \mathbf{x}')$  is replaced by a smoothing function  $W(\mathbf{x} - \mathbf{x}', h)$  of radius  $h$ , the integral representation of  $f(\mathbf{x})$  is approximated over a support domain with radius  $h$  by

$$f(\mathbf{x}) \approx \int_{\Omega} f(\mathbf{x}')W(\mathbf{x} - \mathbf{x}', h)d\mathbf{x}' \quad (4.1)$$

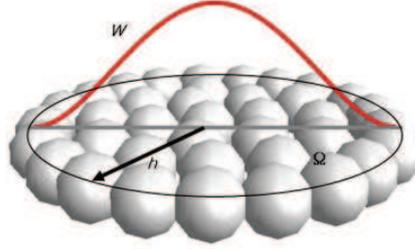


Figure 4.1: Particle approximation over support domain of radius  $h$ . The support domain is spherical but only its cylindrical cross-section is shown in this simple illustration.

The smoothing function  $W$  is usually chosen to be an even function and it should satisfy the following properties:

$$\int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 1 \quad (4.2)$$

$$W(\mathbf{x} - \mathbf{x}', h) = 0 \quad \text{when } |\mathbf{x} - \mathbf{x}'| > h \quad (4.3)$$

$$\lim_{h \rightarrow 0} W(\mathbf{x} - \mathbf{x}', h) = \delta(\mathbf{x} - \mathbf{x}') \quad (4.4)$$

In the SPH method, since the entire system is represented by a finite number of particles, the kernel approximation must be further approximated. This is done by replacing the infinitesimal volume  $d\mathbf{x}'$  in (4.1) by the finite volume of the particle,  $\Delta V_j = m_j/\rho_j$ , where  $m_j$  is the mass of the particle  $j$  and  $\rho_j$  is its density, and summing over the corresponding values of all particles contained in the local support domain of the particle at location  $\mathbf{x}$  (see Figure 4.1)

$$f(\mathbf{x}) \approx \sum_j m_j \frac{f(\mathbf{x}_j)}{\rho_j} W(\mathbf{x} - \mathbf{x}_j, h) \quad (4.5)$$

This process is termed the particle approximation in the SPH method. From here on, we do not distinguish between a continuous function and its approximation. The density approximation  $\rho(\mathbf{x})$  is obtained by substituting  $\rho(\mathbf{x})$  for the function  $f(\mathbf{x})$  in (4.5):

$$\rho(\mathbf{x}) \approx \sum_j m_j \frac{\rho(\mathbf{x}_j)}{\rho_j} W(\mathbf{x} - \mathbf{x}_j, h) = \sum_j m_j W(\mathbf{x} - \mathbf{x}_j, h) \quad (4.6)$$

where  $\rho_j = \rho(\mathbf{x}_j)$ . Any other field function can be approximated in a similar way. Note that the choice of smoothing kernel function  $W$  determines the characteristics of the field function.

Most fluid dynamics problems are modeled by systems of PDEs, in which derivatives of field variables appear. These derivatives must be approximated in the SPH method. By choosing derivative forms of the kernel function in (4.5), the gradient and Laplacian respectively of  $f(\mathbf{x})$  can be approximated by

$$\nabla f(\mathbf{x}) \approx \sum_j m_j \frac{f(\mathbf{x}_j)}{\rho_j} \nabla W(\mathbf{x} - \mathbf{x}_j, h) \quad (4.7)$$

$$\nabla^2 f(\mathbf{x}) \approx \sum_j m_j \frac{f(\mathbf{x}_j)}{\rho_j} \nabla^2 W(\mathbf{x} - \mathbf{x}_j, h) \quad (4.8)$$

In following sections, we describe how to approximate fluid dynamics equations using the SPH method.

## 4.2 SPH Method for Fluid Dynamics

Recall that the SPH method is a meshfree Lagrangian particle method. Thus, in the SPH method the continuity equation is not necessary and the momentum equation is formulated without a convection term (see the Section 3.1.3). We begin with a simple version of the Navier-Stokes equation for modeling fluids:

$$\rho \frac{d\mathbf{u}}{dt} = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}^{\text{external}} \quad (4.9)$$

In fluid dynamics, the force density has the physical dimensions of force per unit volume. Force density is a vector field representing the flux density of any physical force within the bulk of a fluid. The net force on a differential volume element  $dV$  of the fluid is

$$d\mathbf{F} = \mathbf{f} dV = \mathbf{f} \frac{m}{\rho}$$

Thus, by Newton's second law of motion, the force density at a point (i.e., a particle) divided by the density is the acceleration of the fluid at that point:

$$\mathbf{a}_i = \frac{d\mathbf{F}_i}{m_i} = \frac{\mathbf{f}_i}{\rho_i}$$

The sum of the three terms of the right side of (4.9) can be viewed as the total force density exerted on a particle. These three terms are associated with pressure, viscosity and external body force density respectively. Since particles continue to move throughout the simulation, force density fields must vary according to the particles' spatial configuration. Thus, they must be updated at every time step to accelerate particles properly. The next sections describe SPH-based techniques to model these force density fields.

### 4.2.1 Pressure

The pressure term  $-\nabla p$  in (4.9) is approximated by substituting  $p(x)$  for  $f(\mathbf{x})$  in (4.7):

$$\mathbf{f}_i^{\text{pressure}} = -\nabla p(\mathbf{x}_i) \approx -\sum_j m_j \frac{p_j}{\rho_j} \nabla W(\mathbf{x}_i - \mathbf{x}_j, h) \quad (4.10)$$

where  $p_j = p(\mathbf{x}_j)$ . The pressure force plays an important role in maintaining evenly-spaced particles by causing them to attract and repel each other. However, since (4.10) is not symmetric, it produces an unbalanced pressure force between particles. For example, the acting and reacting forces can be unequal in magnitude between a pair of particles  $i$  and  $j$ . This problem can cause numerical instability during the simulation. Desbrun et al. [5] suggested a symmetrization technique to rectify this deficiency:

$$\mathbf{f}_i^{\text{pressure}} \approx -\rho_i \sum_j m_j \left( \frac{p_j}{\rho_j^2} + \frac{p_i}{\rho_i^2} \right) \nabla W(\mathbf{x}_i - \mathbf{x}_j, h) \quad (4.11)$$

Müller et al. [20] proposed a simpler method that is fast and stable enough for many simulations:

$$\mathbf{f}_i^{\text{pressure}} \approx -\sum_j m_j \frac{p_j + p_i}{2\rho_j} \nabla W(\mathbf{x}_i - \mathbf{x}_j, h) \quad (4.12)$$

We use this simple model (4.12) for approximating the pressure force. The pressure field function is formulated by the ideal gas state equation

$$p = k\rho \quad (4.13)$$

where  $k$  is a temperature-dependent gas constant. To maintain rest density,  $\rho_0$ , we use a modified version based on a density offset originally suggested by Desbrun et al. [5]:

$$p = k(\rho - \rho_0) \quad (4.14)$$

This pressure model works well with (4.12) to give appropriate attraction-repulsion forces, as the Lennard-Jones potentials does. Since the pressure force fields encourage particles to be relaxed spatially, this procedure can be seen as a density relaxation.

## 4.2.2 Viscosity

The SPH formulation approximates the viscosity term  $\mu\nabla^2\mathbf{u}$  as

$$\mathbf{f}_i^{\text{viscosity}} = \mu\nabla^2\mathbf{u}(\mathbf{x}_i) \approx \mu \sum_j m_j \frac{\mathbf{u}_j}{\rho_j} \nabla^2 W(\mathbf{x}_i - \mathbf{x}_j, h) \quad (4.15)$$

Viscosity makes a particle's velocity converge to the average velocity of its surrounding particles. The viscosity force depends on the velocity difference between particles, not the absolute velocity of particles. Thus, (4.15) should be modified by substituting the relative velocity for the actual velocity (as suggested in [20]):

$$\mathbf{f}_i^{\text{viscosity}} \approx \mu \sum_j m_j \frac{\mathbf{u}_j - \mathbf{u}_i}{\rho_j} \nabla^2 W(\mathbf{x}_i - \mathbf{x}_j, h) \quad (4.16)$$

If several different viscosities are considered in a multiple fluids system, each viscosity should be computed individually by averaging viscosity coefficients between particles as

$$\mathbf{f}_i^{\text{viscosity}} \approx \sum_j \frac{\mu_i + \mu_j}{2} m_j \frac{\mathbf{u}_j - \mathbf{u}_i}{\rho_j} \nabla^2 W(\mathbf{x}_i - \mathbf{x}_j, h) \quad (4.17)$$

### 4.2.3 External Force

Gravity and collision responses are all considered to be external body forces which appear as last term on the right side of (4.9). While gravity acts on fluid particles all the time, collision response occurs only when a particle collides against a solid boundary. Note that a particle-particle collision normally does not occur because of the pressure force. In this simple case, the external force is expressed as the sum of the gravitational force and the collision responses:

$$\mathbf{f}^{\text{external}} = \mathbf{f}^{\text{gravity}} + \mathbf{f}^{\text{collision}} \quad (4.18)$$

If a mixing interaction is included in the fluid simulation, the interaction force (e.g., diffusive force) is also considered as an additional external force:

$$\mathbf{f}^{\text{external}} = \mathbf{f}^{\text{gravity}} + \mathbf{f}^{\text{collision}} + \mathbf{f}^{\text{interaction}} \quad (4.19)$$

## 4.3 Smoothing Kernels

The smoothing kernel function,  $W$ , plays a very important role in the SPH method, as it determines both the accuracy of the field function representation and the efficiency of computation. The smoothing length  $h$  of the kernel defines the radius of influence of interactions created by a particle. If  $h$  is too small, the particle's interaction is very localized. As a result, the computation may be inaccurate. On the other hand, if  $h$  is too large, the computational complexity increases because the number of interactions grows proportional to  $h$ . Also, since  $W$  no longer approximates  $\delta$  well, the approximation error may increase in (4.1).

Many different kernels have been introduced and constructed for various purposes for SPH methods. Instead of the spline Gaussian kernel which has been used by most researchers [5, 15], we follow Müller's approach suggested in [20] for choosing kernels for the density, pressure and viscosity, because it has been shown to be computationally

efficient and numerically stable enough for fluid simulation. All these kernels have a second order interpolation error and are normalized symmetric functions. Moreover, the function and its derivatives vanish at the boundary of its support (see Figure 4.2). At each particle location, its density is given by

$$W_{poly6}(\mathbf{x}, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - x^2)^3 & 0 \leq x \leq h \\ 0 & \text{otherwise} \end{cases} \quad (4.20)$$

where  $x = |\mathbf{x}|$ . Although density fields are well-approximated with this kernel, pressure force fields are not evaluated correctly. Since the gradient of the kernel function vanishes around the center, the pressure force diminishes as a particle gets closer to its neighbors. This results in a clustering problem due to strong attraction and weak repulsion. To alleviate the clustering problem, different kernels are suggested in [20]. One such kernel is

$$W_{spiky}(\mathbf{x}, h) = \frac{15}{\pi h^6} \begin{cases} (h - x)^3 & 0 \leq x \leq h \\ 0 & \text{otherwise} \end{cases} \quad (4.21)$$

This kernel has non-vanishing gradient around the center and vanishes near the boundary. Thus it can approximate repulsive forces properly. It is shown in Section 4.2.2 that the viscosity decreases velocity difference between particles and damps out their kinetic energy. Unfortunately, since the previous two kernels have negative Laplacians, the viscosity term diverges and makes the fluid simulation unstable by increasing relative velocity between particles. To remedy this problem, [20] introduced a new kernel

$$W_{viscosity}(\mathbf{x}, h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{x^3}{2h^3} + \frac{x^2}{h^2} + \frac{h}{2x} - 1 & 0 \leq x \leq h \\ 0 & \text{otherwise} \end{cases} \quad (4.22)$$

which has a non-negative Laplacian with the following properties:

$$W(|\mathbf{x}| = h, h) = 0 \quad (4.23)$$

$$\nabla W(|\mathbf{x}| = h, h) = 0 \quad (4.24)$$

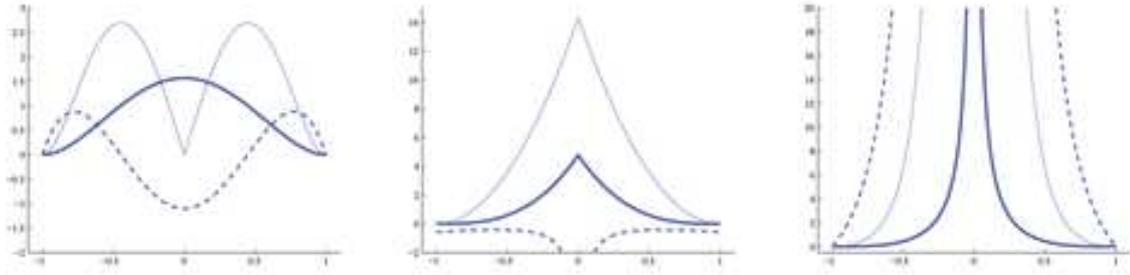


Figure 4.2: The Smoothing Kernels;  $W_{poly6}$ ,  $W_{spiky}$  and  $W_{viscosity}$  (from left to right). Thick lines represent the kernels, thin lines their gradient and dashed lines their Laplacian, respectively.

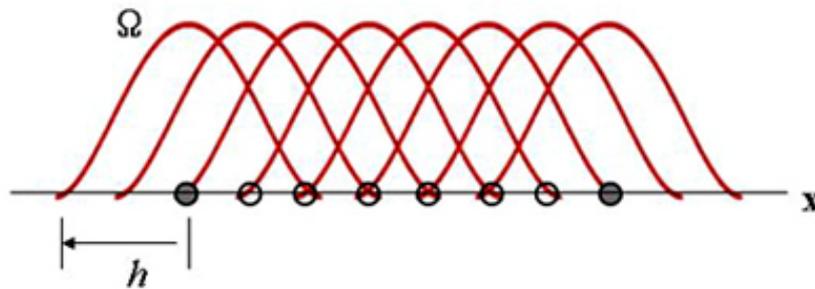


Figure 4.3: Particle approximation for boundary particles (solid circle) and interior particles (empty circle).

## 4.4 Boundary Treatments

In the SPH method, the numerical accuracy depends on the number of particles used to represent the problem domain. If particles are distributed sparsely and unevenly over the domain, their field functions may be approximated poorly. Since an interior particle usually has enough neighboring particles, an associated force is usually well-approximated by a balanced summation of all their interactions. However, for particles near the boundary, only particles inside the computational domain contribute to the particle approximation; no particle contributes from outside (see Figure 4.3). This one-sided contribution frequently produces a poor approximation. It is observed that in fluid

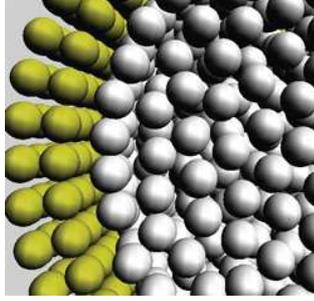


Figure 4.4: Ghost particles(yellow ones) are placed outside of the solid boundary.

simulations, fluid particles near a solid boundary tend to move outward and stick to the boundary. When multiple fluids mix and interact, these unbalanced forces do not allow particles to interact freely. Although this sticking problem has not been addressed in the literature, it could be a serious problem for our simulation of fluid mixtures. To overcome this undesired effect, we use ghost particles, originally introduced by Libersky et al. [14] to treat the generalized SPH boundary problem. We extend their approach to solve the sticking problem. Ghost particles are located just outside of the boundary and produce smooth approximations (see Figure 4.4). They do not evolve through the simulation, but are introduced only to smooth the kernel approximations near the boundary of the computational domain.

# Chapter 5

## Simulation for the Fluid Mixture

In this chapter, we describe our method for simulating a fluid mixture in which multiple fluids mix and interact with each other. We use a basic fluid solver (given in Chapter 4) to model the fluid flow, and integrate it with additional features to represent the fluid mixture. Although the fluids of the mixture can interact in many different ways, we consider only diffusion and chemical reaction for our problem. Both phenomena are modeled using the mathematical formulations explained in Chapter 3. Also, the non-isothermal condition is considered so that the temperature field can change dynamically and it can influence the fluid motions.

### 5.1 Fluid Mixture

For a fluid mixture, more than one fluid exists in the fluid system, which consists of many particles. Each fluid is represented by a disjoint subset of those particles. Thus every particle is assigned an additional attribute,  $k$ , to identify a fluid species or type. Since each fluid may have different properties, such as mass and viscosity, in the SPH method, balanced approximation must be used around the interfaces between fluids. For example, if multiple viscosities are considered in the fluid system, the viscosity force given earlier

as

$$\mathbf{f}_i^{\text{viscosity}} = \mu \sum_j m_j \frac{\mathbf{u}_j - \mathbf{u}_i}{\rho_j} \nabla^2 W(\mathbf{x}_i - \mathbf{x}_j, h) \quad (5.1)$$

should be replaced by

$$\mathbf{f}_i^{\text{viscosity}} = \sum_j \frac{\mu_i + \mu_j}{2} m_j \frac{\mathbf{u}_j - \mathbf{u}_i}{\rho_j} \nabla^2 W(\mathbf{x}_i - \mathbf{x}_j, h) \quad (5.2)$$

## 5.2 Diffusion

Recall that diffusion can occur when there is a spatial gradient of concentration in the fluid mixture (see Chapter 3). Diffusion always proceeds along the direction from higher concentration to lower concentration. Diffusion plays an important role in determining how fluid mixtures form.

### 5.2.1 Concentration

For a concentration attribute, we choose a dimensionless mass fraction notation:

$$w_i^k = \frac{\rho_i^k}{\rho_i} = \frac{\text{mass concentration of species } k \text{ at particle } i}{\text{density at particle } i} \quad (5.3)$$

In the SPH method, this can be written as

$$w^k(\mathbf{x}_i) = \frac{\rho^k(\mathbf{x}_i)}{\rho(\mathbf{x}_i)} = \frac{\sum_{j \in N_k} m_j W(\mathbf{x}_i - \mathbf{x}_j, h)}{\sum_j m_j W(\mathbf{x}_i - \mathbf{x}_j, h)} \quad (5.4)$$

where  $N_k$  is a set of particles representing fluid species  $k$  and  $\sum_k w^k(\mathbf{x}_i) = 1$ .

### 5.2.2 Diffusive Force

As given in (3.11), the diffusive force is inversely proportional to the spatial gradient of concentration. The diffusive force of fluid  $k$  is computed by

$$\mathbf{f}_i^{\text{diffusion},k} = -d \nabla w_i^k \quad (5.5)$$

where  $d$  is a temperature dependent diffusion coefficient. Equation (5.5) can be approximated using the SPH method as

$$\mathbf{f}_i^{\text{diffusion,k}} \approx -d \sum_j m_j \frac{w_j^k}{\rho_j} \nabla W(\mathbf{x}_i - \mathbf{x}_j, h) \quad (5.6)$$

Equation (5.6) models the diffusion well over interior particles, but it does not deal with boundary particles correctly. On one side of a boundary particle, there are neighboring particles, but, on the other side, there are no particles. Thus, the SPH method regards the boundary side as a region of relatively low concentration of the fluid. Thus, near the boundary, the diffusion may proceed towards the boundary and it may produce an unrealistic numerical solution. To avoid this problem, we introduce the new scalar function for the concentration offset:

$$f(w_j^k) = \begin{cases} c(w_j^k - w_i^k) & w_i^k > w_j^k \\ 0 & \textit{otherwise} \end{cases} \quad (5.7)$$

where  $c$  is a user-defined positive constant. Substituting (5.7) for  $w_j^k$  in (5.6), the diffusive force becomes

$$\mathbf{f}_i^{\text{diffusion,k}} \approx -d \sum_j m_j \frac{f(w_j^k)}{\rho_j} \nabla W(\mathbf{x}_i - \mathbf{x}_j, h) \quad (5.8)$$

Equation (5.8) constrains the particles' movements within the fluid domain when the diffusion proceeds in the fluid mixture. If there exists more than one species of solutes, all of their concentrations can be individually computed at every particle. However, for the diffusion, a particle is accelerated only by the diffusive force associated with its species.

### 5.3 Chemical Reaction

Only the simple chemical reaction



is considered. In a non-isothermal system, an extra heat source term may be added on either side of (5.9). As described in Chapter 3, the chemical reaction can occur when reactants (e.g.,  $A$  and  $B$ ) join together to create new products (e.g.,  $E$  and  $F$ ) which have different chemical properties. This process involves changes in the configuration of electrons and energy around reactant atoms or molecules. When atoms become ionized, some of them lose electrons while others gain them. Thus, each atom may have an imbalance between the number of electrons and protons. If the atom gains electrons, it becomes a negatively charged ion; if it loses electrons, it becomes a positively charged ion. When these ionized reactants join together, new chemical bonds can be created between them and new products can be created.

The simulated chemical reactions in our fluid system are not based on real physical and chemical laws. Instead, we take a simpler approach. Every particle is assigned an additional chemical attribute: electric charge,  $e$ , to represent an ionic state. When a reactant is charged negatively, its particles have negative  $e$ . When it is charged positively, they have positive  $e$ . Even though electron migration occurs from electrostatic potential energy and its electric force, we simply simulate electron flow in the same way as we do heat transfer using the diffusion equation

$$\frac{\partial e}{\partial t} = \alpha \nabla^2 e \quad (5.10)$$

$$e_{i,t} = e_{i,t-\Delta t} + \Delta t \frac{\partial e_i}{\partial t}$$

Equation (5.10) is approximated using the SPH method:

$$\frac{\partial e}{\partial t} \approx \alpha \sum_j m_j \frac{e_j - e_i}{\rho_j} \nabla^2 W(\mathbf{x}_i - \mathbf{x}_j, h)$$

When differently charged reactants form new bonds, their resulting charges may become neutralized. Using (5.10), electric charge  $e$  eventually converges to 0. Once a particle's electric charge reaches 0, it can be regarded as neutralized and new products can be created.

When a new product is created, contributing reactants should be consumed according to chemical reaction equations and the mass conservation law. To define a mapping between reactants and products, we introduce the following transition function

$$k = k(k_{initial}, e, r) \quad (5.11)$$

where  $k_{initial}$  is the species present in the initial state,  $k$  is the species in the current state and  $r$  is the reaction rate. Here,  $k$  is a piecewise constant function of  $e$  and  $r$ . Since we do not consider any intermediate states between each species,  $k(\cdot)$  is a discrete function.

## 5.4 Non-Isothermal Conditions

For modeling a non-isothermal fluid system, we consider the following features:

- temperature field
- temperature dependent viscosity
- heat from the chemical reactions or external sources

### 5.4.1 Temperature Field

The temperature attribute is introduced to model the heat transfer. The temperature field evolves during the simulation due to heat diffusion. This evolution is modeled in the same way as the electric charge  $e$  is:

$$\begin{aligned} \frac{\partial T}{\partial t} &= \beta \nabla^2 T + \sigma \\ T_{i,t} &= T_{i,t-\Delta t} + \Delta t \frac{\partial T_i}{\partial t} \end{aligned} \quad (5.12)$$

where  $\beta$  is the thermal diffusivity and  $\sigma$  is the heat source term. Equation (5.12) is approximated using the SPH method:

$$\frac{\partial T}{\partial t} \approx \beta \sum_j m_j \frac{T_j - T_i}{\rho_j} \nabla^2 W(\mathbf{x}_i - \mathbf{x}_j, h)$$

Heat can be generated from chemical reactions or supplied from external sources. Since temperature influences the rest density of the fluid, the rest density,  $\rho_0$ , is modeled as

$$\rho_0 = \rho_{0,initial}[1 - \gamma(T - T_0)] \quad (5.13)$$

where  $\rho_{0,initial}$  and  $T_0$  are the initial value of the rest density and the temperature, respectively, and  $\gamma$  is a user-defined constant.

### 5.4.2 Thermal Viscosity

In an isothermal fluid system, the fluid viscosity remains constant during the simulation. However, in a non-isothermal system, it may vary with the temperature. For temperature dependent viscosity, we choose the following viscosity model suggested in [28]:

$$\mu_i = be^{-aT_i}, \quad a > 0, b > 0$$

Using this model, viscosity increases exponentially as the temperature decreases.

## 5.5 Immiscible Fluids

So far, we have considered the interactions between miscible fluids only. Simulating immiscible fluids, such as oil and water, is not a trivial problem, because preventing blending may not be sufficient to produce a realistic numerical solution. In multiple fluids system, there is more than one set of particles; each set represents a specific fluid. When these fluids flow, the interfaces between them move accordingly. However, when using particle methods, different types of particles may penetrate across the interfaces. This may produce a subtle mixing around the interfaces. To avoid this problem, we introduce a penalty force, given by the molecular force of the Lennard-Jones potential:

$$\mathbf{f}^{\text{penalty}} = \begin{cases} q[(\frac{h}{r_{ij}})^n - (\frac{h}{r_{ij}})^m] \frac{\mathbf{x}_{ij}}{r_{ij}^2} & h \geq r_{ij} \\ 0 & h < r_{ij} \end{cases} \quad (5.14)$$

where  $q$  is a user-defined constant and  $r_{ij} = |\mathbf{x}_{ij}| = |\mathbf{x}_i - \mathbf{x}_j|$ . This penalty force helps prevent different types of particles from moving across the interface that separates them.

# Chapter 6

## Implementation Issues

In this chapter, we provide a brief overview of the implementation issues of our method. In the SPH method, considerable computation is needed to evaluate various physical forces between particles. If many particles are considered, this computation can be a major bottleneck, making the simulation slow. We partially avoid this problem by using a grid-based technique to accelerate neighbor search. Also, we present details about the simulation steps of our method, and show our approach for rendering surfaces to be natural for modeling fluid mixtures.

### 6.1 Neighbor Search

Recall that the SPH method performs the evaluation of various forces using the smoothing kernel having the localized and finite support. Thus, since only nearby neighboring particles contribute to this computation, it is necessary to identify them first using a nearest neighbor search. A naive approach is to examine all particles and check if they are located within  $h$  of particle  $i$ . Since we must perform this calculation for all  $n$  particles, this approach has a running time proportional to  $n^2$  and becomes intractable for a simulation with many particles. Thus, we use a 3D grid of cells of size  $h$ , where  $h$  is the smoothing length of the kernel function  $W$ . Only particle  $i$ 's own cell and its

adjacent cells are examined when searching for the neighbors of particle  $i$ . As particles are assumed to be evenly distributed, the time complexity of the force computation is reduced to  $O(n)$ .

## 6.2 Simulation Steps

For a typical simulation, various forces are computed at the current position of a particle and then are accumulated to accelerate it. In our fluid model, all these forces are expressed as

$$\mathbf{f}_i = \mathbf{f}_i^{\text{pressure}} + \mathbf{f}_i^{\text{viscosity}} + \mathbf{f}_i^{\text{gravity}} + \mathbf{f}_i^{\text{diffusion}} + \mathbf{f}_i^{\text{penalty}} \quad (6.1)$$

It is observed that the pressure force,  $\mathbf{f}_i^{\text{pressure}}$  is dominant for accelerating particle  $i$  in (6.1). Thus, we modify this explicit scheme by splitting into two sub-time-steps:

$$\begin{aligned} \mathbf{f}'_i &= \mathbf{f}_i^{\text{viscosity}} + \mathbf{f}_i^{\text{gravity}} + \mathbf{f}_i^{\text{diffusion}} + \mathbf{f}_i^{\text{penalty}} \\ \mathbf{f}_i &= \mathbf{f}'_i + \mathbf{f}_i^{\text{pressure}} \end{aligned}$$

In the first step, all forces except the pressure force are computed to accelerate the particle. The immiscibility condition is included in the penalty force. In the second step, the pressure force redistributes the particle. Our simulation scheme proceeds as follows:

- (a) Update density and concentrations  $(\rho_i, w_i)$
- (b) Compute viscosity, diffusion, penalty and gravitational forces  
 $(\mathbf{f}_i^{\text{viscosity}}, \mathbf{f}_i^{\text{diffuse}}, \mathbf{f}_i^{\text{penalty}}, \mathbf{f}_i^{\text{gravity}})$
- (c) Update temperature and electric field for chemical reaction  $(T_i, e_i)$
- (d) Update velocity and position with collision detection  $(\mathbf{v}_i, \mathbf{x}_i)$
- (e) Update density  $(\rho_i)$
- (f) Compute pressure force  $(\mathbf{f}_i^{\text{pressure}})$

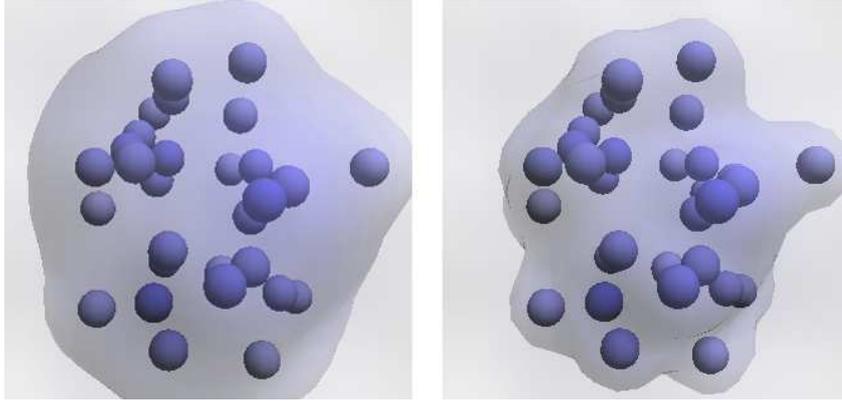


Figure 6.1: Surface Reconstruction: the surface is created by the implicit function of the location of the particles (left) and the implicit function is scaled by the particles' concentrations (right).

(g) Update velocity and position with collision detection ( $\mathbf{v}_i, \mathbf{x}_i$ )

### 6.3 Surface Reconstruction

Once the particles' properties are updated through the simulation steps, we extract the fluid surface from the particles using the marching cube algorithm [16]. This method extracts an implicit surface based on a scalar function of the particles' locations. For a single fluid, only the free surface must be tracked. However, for fluid mixtures, various interface surfaces (e.g., surfaces between fluids) must be tracked as well. Each fluid surface is represented by its associated particles: surface  $k$  is generated by tracking the particles associated with species  $k$ . For the implicit value  $\phi_i$ , we use a spherical function of the distance from a particle  $i$ . We recompute all particles' implicit values through the simulation and update fluid surfaces using these new values. Since surfaces are re-created at each frame, they might lose the temporal coherence in the video frames. Also, it is observed that the surface changes abruptly because subtle features in the interface between fluids cannot be modeled properly (left image in Figure 6.1). Thus, we improve upon the straightforward surface rendering approach described above using

concentrations which are already computed to perform the diffusion. The concentration field associated with all the particles changes smoothly through the simulation. When we compute the implicit values of the marching cube at the particle  $i$ , we scale them by particle  $i$ 's concentration  $w_i$ . If  $w_i$  is small, particle  $i$ 's influence on the implicit values is small. Conversely, if  $w_i$  is large, particle  $i$ 's influence on the implicit values is large. This allows our method to evolve the surface more smoothly and naturally around the boundary between fluids (right image in Figure 6.1).

# Chapter 7

## Results

In this chapter, we present some experimental results. We focus on visualizing effects associated with diffusion and chemical reactions. Also, we discuss some issues associated with the integration methods used in our simulations.

### 7.1 Animation

Using the techniques described in Chapter 5, we simulate some fluid mixtures for which diffusion and chemical reactions are significant. Our simulation is initialized with water in a cubic container. We model two cases: a single fluid added to the water and two fluids added to the water. The former is modeled by 1000 particles, while the latter is modeled by 1800 particles. The animations were run on a 1.7 GHz Pentium 4 PC with 1G memory and an ATI Radeon 8500 graphics card. For the mixture of a single fluid with the water, the animation runs at approximately 17 *frames/sec* without surface rendering. If we render the surface as well, the animation speed drops to approximately 0.9 *frames/sec*. For the mixture of two fluids with the water, the animation runs at approximately 15 *frames/sec* without surface rendering and at about 0.3 *frames/sec* with surface rendering.

### 7.1.1 Diffusion

Figures 7.1, 7.2 and 7.3 illustrate the mixture of a single fluid with the water; more specifically, a blue liquid is dropped into the water and it mixes with the water. When the blue liquid meets the water, the two fluids immediately start to mix. The nature of the mixture depends on whether the two fluids are immiscible or miscible. If both are immiscible, no blending occurs between them. Otherwise, diffusion causes them to blend. In this simulation, a water particle weighs  $0.01 \text{ kg}$ , while a blue liquid particle weighs  $0.02 \text{ kg}$ . Thus, the blue liquid tends to sink in the water. Figure 7.4 illustrates the mixture of two colored liquids in the water. The simulation begins with the two colored liquids in the water. The colored liquids are assigned different diffusion rates; the red liquid blends slower than the blue liquid does.

### 7.1.2 Chemical Reaction

We also simulate a simple chemical reaction. In the mixture of the two colored liquids in the water, we allow the colored liquids to react to produce a new liquid of a different color. More specifically, in Figure 7.5, the blue and red liquids react to produce a new purple liquid.

## 7.2 Integration Methods

For the time integration of our simulation, we tested three explicit integration schemes: the Euler, improved Euler and leap-frog methods. For an indication of the error associated with these methods, we measured the kinetic energy of the entire particle system throughout the numerical integration. The energy should decay with time because of damping due to the viscosity in the system. The kinetic energy at time  $t$  is expressed as

$$E^k(t) = \sum_i \frac{1}{2} m_i \|\mathbf{v}_i(t)\|^2 \quad (7.1)$$

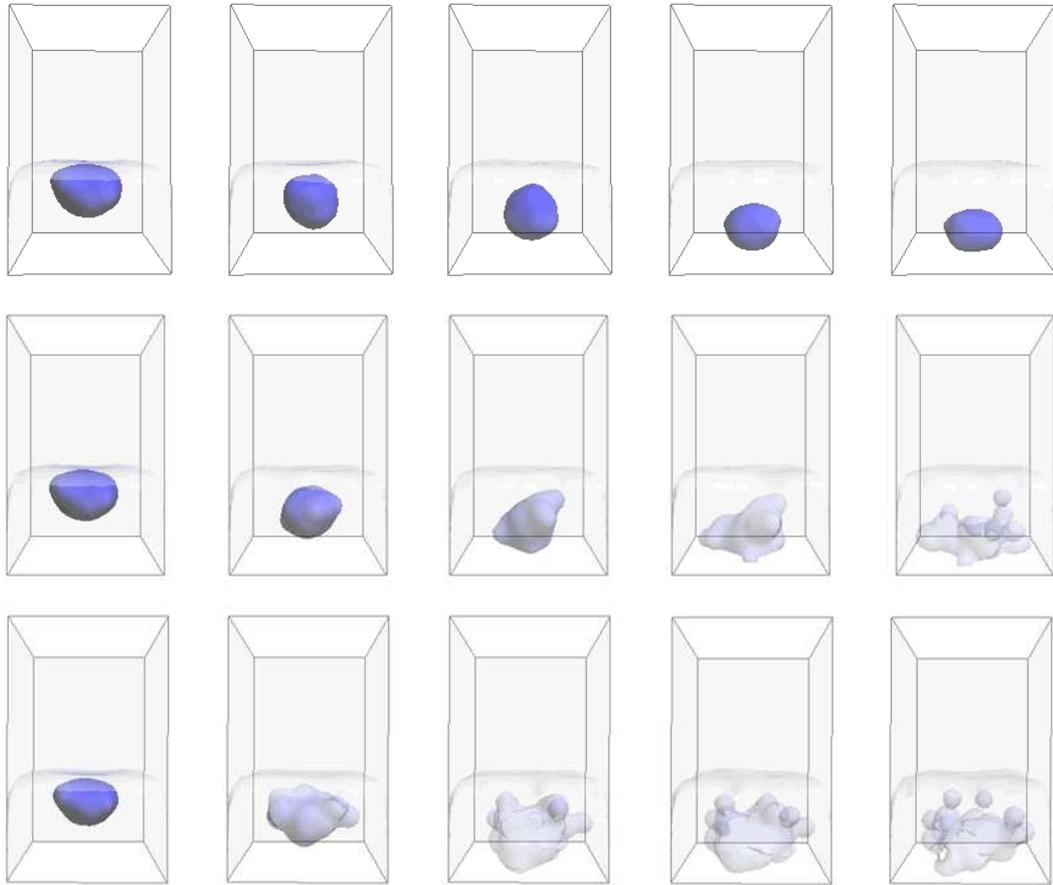


Figure 7.1: Mixture of a single fluid in the water (side view). The top row shows a blue liquid dropped into the water. The liquids are immiscible and therefore do not mix. In the middle row, the blue liquid changes from immiscible to miscible during the simulation. At first, the blue liquid sinks with no mixing, then it starts to diffuse in the water. The bottom row simulates two miscible fluids. The diffusion causes the blue liquid and the water to mix well.

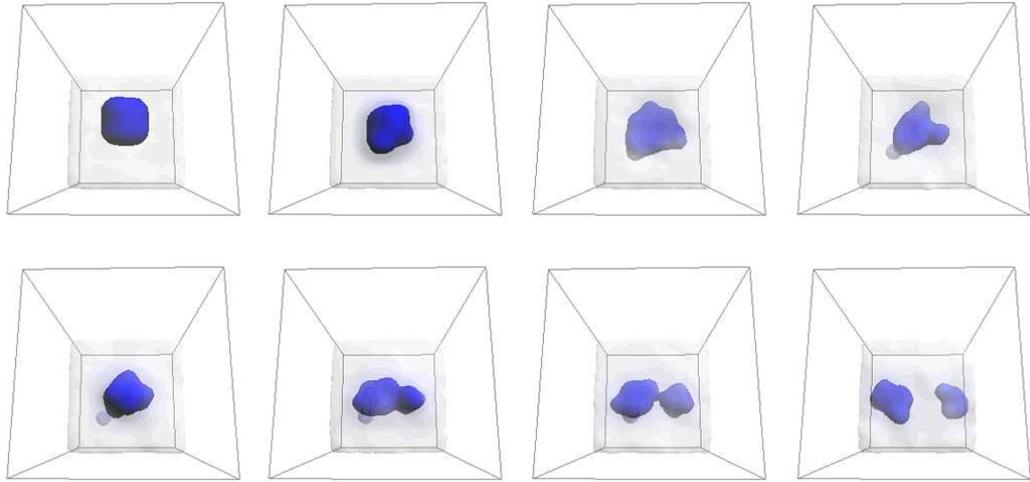


Figure 7.2: Mixture of a single fluid in the water (top view). The blue liquid drops into the water but the immiscibility of the two liquids prevents them from mixing.

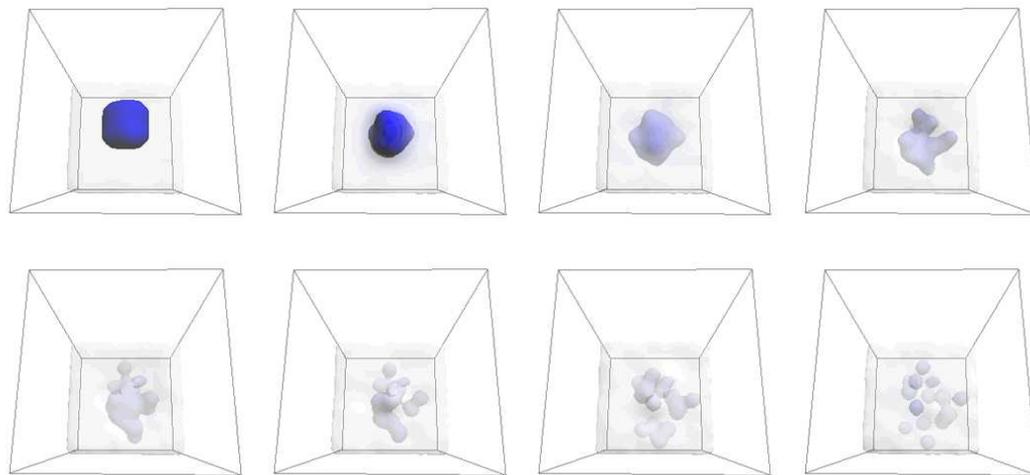


Figure 7.3: Mixture of a single fluid in the water (top view). The blue liquid drops into the water. Since the two liquids are miscible in this case, they mix well due to the diffusion.

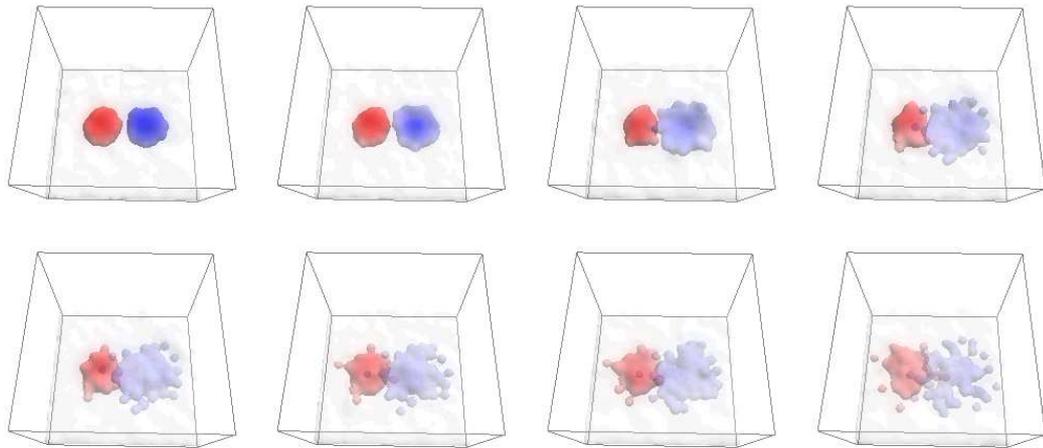


Figure 7.4: Mixture of two fluids in the water (top view). The red liquid mixes slower than the blue liquid does due to different diffusion rates.

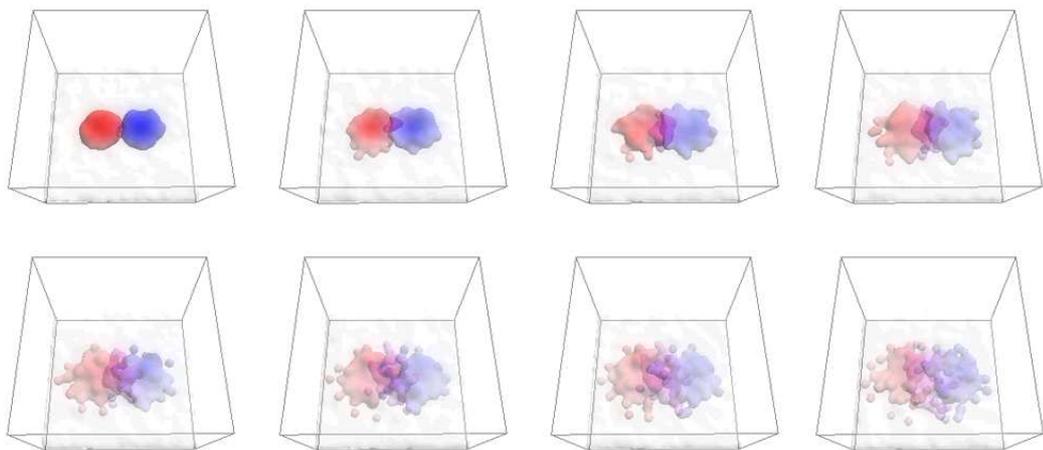


Figure 7.5: Chemical reaction (top view). When the red liquid and the blue liquid meet, a chemical reaction occurs that produces a purple liquid.

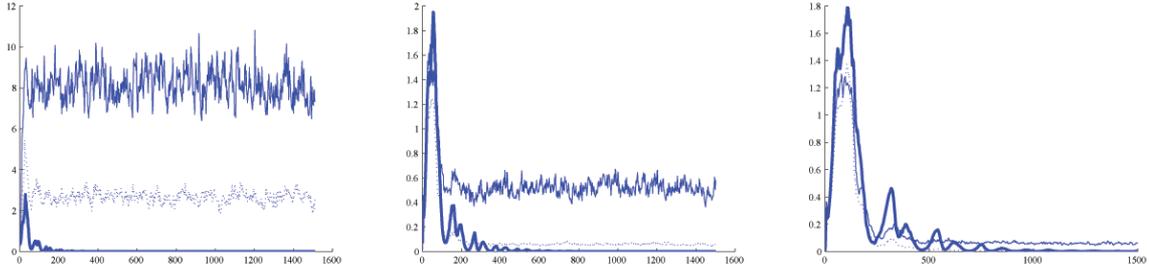


Figure 7.6: Time integration with  $\Delta t = 0.008$ ,  $0.004$  and  $0.002$  (from left to right). The solid line represents the results for the explicit Euler method, the dashed line represents the results for the improved Euler method and the thick line represents the results for the leap-frog method. The vertical axis is the energy of the particle system and the horizontal axis is the time step number.

The Euler method is

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \mathbf{v}(t)$$

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \Delta t \cdot \mathbf{a}(t)$$

The improved Euler method is

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \Delta t \cdot \mathbf{a}(t)$$

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \frac{1}{2}(\mathbf{v}(t) + \mathbf{v}(t + \Delta t))$$

The leap-frog method is

$$\mathbf{v}(t + \frac{1}{2}\Delta t) = \mathbf{v}(t - \frac{1}{2}\Delta t) + \Delta t \cdot \mathbf{a}(t)$$

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \mathbf{v}(t + \frac{1}{2}\Delta t)$$

Our simulation is initialized with 800 particles which are randomly distributed in the cubic container. Once the simulation starts, particles fall, interact with others and bounce against the solid walls of the container until they reach the rest state. For each time step-size ( $\Delta t = 0.008$ ,  $0.004$  and  $0.002$ ), we collected data on the total kinetic energy and

plotted them in Figure 7.6. Note that there are many peaks in the energy plots. When particles happen to stay within the collision distance of others, they are immediately pushed away by repulsive forces, which often cause abrupt increase in the kinetic energy. Also, since particles are randomly positioned at  $t = 0$ , they experience many collisions at the earlier time steps.

For  $\Delta t = 0.008$ , both the Euler and improved Euler methods exhibit instability, but the leap-frog method appears to be stable. In particular, the kinetic energy associated with the Euler method has large oscillations. For  $\Delta t = 0.004$ , both the improved Euler and the leap-frog scheme appear to be stable, but the Euler scheme remains unstable. When time step-size is reduced to  $\Delta t = 0.002$ , all methods appear to be stable.

In summary, the three explicit methods tested perform well only for small time step-sizes. Among them, the leap-frog method seems to be best. This method allows a time step-size up to 0.02 for this simulation.

# Chapter 8

## Conclusion and Future Work

We have presented our method for simulating fluid mixture. Our fluid model uses a particle-based method, the SPH method. As constitutive models to represent the fluid mixture, we developed physical diffusion and chemical reaction procedures. These procedures determine an interaction between fluids and produce various fluid mixture effects. They perform well with a particle-based fluid governed by the Navier-Stokes equations. Also we improved some difficulties for particle methods associated with the boundary problem by using virtual particles. We provided a new method for reconstructing implicit surfaces based on both iso-values and concentration attributes of particles. Our method is capable of capturing subtle features of mixing fluids.

There are many possible extensions for our method and future work. We hope to extend our fluid simulation to cope with fluid-solid interactions and user-interactions. All of these interactions can be implemented as additional procedures, such as stir, shake and splash, to affect behaviors and shapes of the fluid mixtures. To this end, two separate domains of the fluid and solid must be coupled and simulated simultaneously. Since the solid dynamics model is usually solved by the Lagrangian method, it can be easily coupled to our particle-based fluid model. We hope to explore different types of mixtures by applying our diffusion model to various materials, such as fabric and paper. For

example, when we spill ink onto clothes, it blots and soaks into them. We believe we can simulate this in a way similar to that developed in this thesis for the fluid mixtures, but also taking into account surface and material properties of clothes. Physical diffusion may proceed in an anisotropic manner. Our fluid mixture is based on Newtonian fluid dynamics. We hope to include non-Newtonian fluid effects to enhance the richness of our mixtures.

Visualization of fluids could be improved by optimizing surface generation and using high quality renderers (e.g., pixie and renderman). We use the marching cube algorithm to create surfaces and we observe that its quality depends noticeably on the spatial distribution of particles. It often produces poor results for visualizing sharp features or boundaries which are often represented by a few particles. A particle level set or a hybrid level set method could improve the quality of surfaces extracted from moving particles.

We use the explicit Euler method for the time integration. Thus, our simulation is stable only if a small time step is chosen. Implicit methods could improve the effectiveness of our time integration scheme. Since the performance of the SPH method depends on the particles' spatial configuration, it does not work well for nonuniform particle spacing. Also, the SPH method has difficulty modeling incompressibility and keeping pressures constant. These weaknesses could be reduced by augmenting our schemes with grid-based approaches. Parallel methods could be employed to speedup the computation.

# Bibliography

- [1] N. Bell, Y. Yu, P. Mucha. Particle-based Simulation of Granular Materials. *Proceedings of 2005 ACM SIGGRAPH Symposium on Computer Animation*, pages 77-86, 2005.
- [2] M. Carlson, P. J. Mucha, R. Horn III, G. Turk. Melting and Flowing. *Proceedings of 2002 ACM SIGGRAPH Symposium on Computer Animation*, pages 167-174, 2002.
- [3] M. Carlson, P. J. Mucha, G. Turk. Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid. *Proceedings of 2004 ACM SIGGRAPH*, pages 377-384, 2004.
- [4] S. Clavet, P. Beaudoin, P. Poulin. Particle-based Viscoelastic Fluid Simulation. *Proceedings of 2005 ACM SIGGRAPH Symposium on Computer Animation*, pages 219-228, 2005.
- [5] M. Desbrun, M. P. Cani. Smoothed particles: A new paradigm for animating highly deformable bodies. *6th Eurographics Workshop on Computer Animation and Simulation '96*, pages 61-76, 1996.
- [6] D. Enright, S. Marschner, R. Fedkiw. Animation and rendering of complex water surfaces. *Proceedings of 2002 ACM SIGGRAPH*, pages 736-744, 2002.
- [7] R. Fedkiw, J. Stam, H. W. Jensen. Visual Simulation of Smoke. *Proceedings of 2001 ACM SIGGRAPH*, pages 15-22, 2001.

- [8] N. Foster, D. Metaxas. Realistic Animation of Liquids. *Graphical Models and Image Processing 58*, pages 471-483, 1996.
- [9] N. Foster, D. Metaxas. Modeling the motion of a hot, turbulent gas. *Proceedings of 1997 ACM SIGGRAPH*, pages 181-188, 1997.
- [10] N. Foster, R. Fedkiw. Practical Animation of Liquids. *Proceedings of 2001 ACM SIGGRAPH*, pages 23-30, 2001.
- [11] O. G enevaux, A. Habibi, J-M. Dischler. Simulating Fluid-Solid Interaction. *Graphics Interface 2003*, pages 31-38, 2003.
- [12] R. A. Gingold, J. J. Monaghan. Smoothed Particle Hydrodynamics: Theory and Application to Non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, pages 181:375-389, 1977.
- [13] I. Ihm, B. Kang, D. Cha. Animation of Reactive Gaseous Fluids through Chemical Kinetics. *Proceedings of 2004 ACM SIGGRAPH Symposium on Computer Animation*, pages 203-212, 2004.
- [14] L. D. Libersky, A. G. Petscheck, T. C. Carney, J. R. Hipp, F. A. Allahdadi. High strain Lagrangian hydrodynamics—a three dimensional SPH code for dynamic material response. *Journal of Computational Physics*, pages 109:67-75, 1993.
- [15] G. R. Liu, M. B. Liu. Smoothed Particle Hydrodynamics: A Meshfree Particle Method. *World Scientific Publishing*, 2003.
- [16] W. E. Lorensen, H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Proceedings of 1987 ACM SIGGRAPH*, pages 163-169, 1987.
- [17] F. Losasso, T. Shinar, A. Selle, R. Fedkiw. Multiple Interacting Liquids. *Proceedings of 2006 ACM SIGGRAPH*, pages 812-819, 2006.

- [18] L. B. Lucy. Numerical approach to testing the fission hypothesis. *Astronomical Journal*, pages 82:1013-1024, 1977.
- [19] H. Mao, Y. Yang. A particle-based model for non-Newtonian fluid animation. *Technical report TR05-21*, Department of Computer Science, University of Alberta, 2005.
- [20] M. Müller, D. Charypar, M. Gross. Particle-Based Fluid Simulation for Interactive Application. *Proceedings of 2003 ACM SIGGRAPH Symposium on Computer Animation*, pages 154-159, 2003.
- [21] M. Müller, S. Schirm, M. Teschner, B. Heidelburger, M. Gross. Interaction of Fluids with Deformable Solids. *Journal of Computer Animation and Virtual Worlds*, pages 159-171, 2004.
- [22] M. Müller, B. Solenthaler, R. Keiser, M. Gross. Particle-Based Fluid-Fluid Interaction. *Proceedings of 2005 ACM SIGGRAPH Symposium on Computer Animation*, pages 237-244, 2005.
- [23] S. Premoze, T. Tasdizen, J. Bigler, A. Lefohn, R. T. Whitaker. Particle-Based Simulation of Fluids. *Computer Graphics Forum*, pages 401-410, 2003.
- [24] R. Probstein. Physicochemical Hydrodynamics: An Introduction. *John Wiley & Sons*, 2003.
- [25] W.T. Reeves. Particle systems — a technique for modeling a class of fuzzy objects. *Proceedings of 1983 ACM SIGGRAPH*, pages 359-375, 1983.
- [26] K. Steele, D. Cline, P. K. Egbert, J. Dinerstein. Modeling and rendering viscous liquids. *Journal of Computer Animation and Virtual Worlds 15*, pages 183-192, 2004.
- [27] J. Stam. Stable fluids. *Proceedings of 1999 ACM SIGGRAPH*, pages 121-128, 1999.
- [28] D. Stora, P-O. Agliati, M-P. Cani, F. Neyret, J-D. Gascuel. Animating Lava Flows. *Graphics Interface 99*, pages 203-210, 1999.

- [29] D. Tonnesen. Modeling Liquids and Solids using Thermal Particles. *Graphics Interface 91*, pages 255-262, 1991.
- [30] Y. Zhu, R. Bridson. Animating Sand as a Fluid. *Proceedings of 2005 ACM SIGGRAPH*, pages 965-972, 2005.