Fluid-Fluid Collision for Particle-Based Immiscible Fluid Animation

Hai Mao and Yee-Hong Yang
Department of Computing Science
University of Alberta

Abstract

In this paper, we propose a new fluid-fluid collision model for particle-based immiscible fluid animation. Our model consists of two components, namely, collision detection and collision response. An interesting feature is that our model not only can prevent immiscible fluids from mixing with each other, but also can allow one fluid to run through or to wrap around another fluid. The model is very flexible and can work with many existing particle-based fluid models. The animation results are presented and show that the proposed model can produce many interesting fluid animations.

Key words: Fluid animation, fluid-fluid collision, non-Newtonian fluids, convex hull, collision detection, collision response.

1 Introduction

Fluid animation is a popular topic in computer graphics. It enriches a computer-generated virtual world and has found many applications, for instance, in motion pictures and computer games. Fluids exhibit a wide range of interesting motions during and after their interactions with rigid or deformable objects. As well, many interesting fluid motions result from fluid-fluid collision, which is the focus of this paper.

In computer graphics, there are many grid-based fluid models in addition to particle-based fluid models. In comparison, these two types of the fluid models have their own advantages and disadvantages. More discussions of the comparison can be found in [18] [20] [21] In this paper, we concentrate on the fluid-fluid collision using the particle-based fluid models.

Keiser et al. [13] propose a contact handling model which uses a penalty force to prevent the mixture of deformable point-based objects. This model requires a two-layer representation: “the volumes of the objects are discretized into a set of points (or phyxels) on which external forces can be applied, and a set of surface elements (or surfels) which are animated along with the phyxels”[13] Therefore, it would be difficult for this model to work with many existing particle-based fluid models, such as [4] [5] [15] [17] [21] where a fluid is not represented by those two sets of points.

Muller et al. [19] introduce an interface body force, which acts perpendicular to the interface of two fluids and always points from one fluid to the other fluid. Thus, such a force is not appropriate to simulate the mutual interaction in a fluid-fluid collision.

In this paper, we propose a new particle-based fluid-fluid collision model for immiscible fluid animation. Our model consists of collision detection and collision response. And, during immiscible fluid collisions, our model not only can prevent the fluids from mixing with each other, but also can allow one fluid to run through or to wrap around another fluid. Meanwhile, our model can easily work with a particle-based fluid model. It requires: (1) a particle neighbor search and (2) particles with the two common attributes, mass and velocity. Many existing particle-based fluid models, such as [4] [5] [15] [17] [21] can satisfy these two requirements. Note that the particles have no explicit mass in [4] [21] but it is trivial to assign masses to the particles.

This paper is organized as follows. The related works in fluid animation are discussed in Section 2. The particle-based fluid model that works with the proposed collision model is briefly described in Section 3. Collision detection and collision response are explained in details in Sections 4 and 5, respectively. Then, the animation results are presented in Section 6. Finally, the conclusion and future work are given in Section 7.

2 Related Work

In computer graphics, it is widely accepted that the full three-dimensional Navier-Stokes (NS) equation is the most comprehensive dynamics model for fluid animation. Based on the way to solve the NS equation, the fluid models can be categorized as the particle-based models and the grid-based models.

In grid-based fluid models, the NS equation is solved in a three-dimensional grid structure. The structure covers the animation space in which the fluid exists or may move into. Stam [22] introduces a semi-Lagrangian method to solve the NS equation in the grid structure. The solution is stable even at large time steps. In order to evolve free fluid surface in the grid structure, a hybrid method of combining marker particles and the level set method is introduced by Foster and Fedkiw [7] and later is improved by Enright et al. [6] In addition, more fluid phenomena are animated. For examples, fluid melting and flowing is animated by Carlson et al. [2] bubbles in liquid by Hong and Kim [11] [12] water drops on surfaces by Wang et al. [26] fluid with splash and foam by Takahashi et al. [24] and viscoelastic fluids by Goktekin et al. [9] To animate fluid-solid interaction, Genevaux et al. [8] propose an interface between the fluid and the solid, while Carlson et al. [3] treat rigid solids as special fluids with rigid motions. Guendelman et al. [10] propose a coupling method for water to interact with thin deformable and rigid shells.

In particle-based fluid models, a particle represents a piece of the animated fluid. The NS equation is solved with the fluid particles. A popular particle-based fluid formulism is the *Smoothed Particle Hydrodynamics* (SPH). Desbrun and Cani [5] use particles to animate highly deformable bodies based on the SPH. Muller et al. [17] propose a SPH-based model that can interactively animate fluid splashing and swirling. The SPH is also utilized to animate lava flows by Stora et al. [23] point-based elastic, plastic, and melting objects by Muller et al. [18] non-Newtonian fluids by Mao and Yang [15] fluid-solid interaction by Muller et al. [20] and fluid-fluid interaction by Muller et al. [19] Other than the SPH, the *Moving-Particle Semi-Implicit* (MPS) is another particle-based fluid formulism. Premoze et al. [21] propose a MPS-based fluid model which can produce appealing animations of fluid splashing and swirling. Based on the basic idea of using particles to represent fluids, Clavet et al. [4] propose a particle-based fluid model which simulates viscoelastic fluid behaviors by insertion and removal of springs between pairs of particles.

3 Fluid Modeling

Our model can work with many particle-based fluid models. For completeness, we briefly describe the fundamentals of a SPH-based fluid model, which we utilize to work with our model. For a detailed description of the SPH-based fluid models, the reader is referred to the previous works in [5] [15] [17] [19] [20]

The SPH is a Lagrangian formulism for modeling fluids and is based on the interpolation theory. The SPH divides a fluid into a set of elements called particles, which are used to carry fluid attributes. A scalar attribute value $A(r)$ at position r is interpolated by the values of particles within a local neighborhood:

$$A(r) = \sum_{j=1}^{n} A_j \frac{m_j}{\rho_j} W(r - r_j, h)$$

where n is the number of neighboring particles, j the particle index, A_j the particle attribute value, m_j the particle mass, ρ_j the particle density, r_j the particle position, h the neighborhood radius, and W the interpolation weighting function called kernel. In this paper, the traditional spline kernel from [16] is used:

$$W(r - r_j, h) = \frac{1}{\pi h^3} \begin{cases} 1 - 1.5q^2 + 0.75q^3 & 0 \le q \le 1 \\ 0.25(2-q)^3 & 1 \le q \le 2 \\ 0 & otherwise \end{cases}$$

where $q = 2|r\text{-}r_j|/h$. The density ρ_i of particle i at position r_i is evaluated as:

$$\rho_i = \sum_{j=1}^{n} m_j W(r_i - r_j, h) \, .$$

A SPH-based fluid model solves fluid motions using Lagrangian version of the Navier-Stokes equation:

$$\frac{dv}{dt} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot T + \frac{1}{\rho} f$$

where v is the velocity, t the time, ρ the density, p the pressure, T the stress tensor, μ the viscosity constant, and f the summation of external forces such as gravity. This Navier-Stokes equation is solved for Newtonian fluids in [17] and for non-Newtonian fluids in [15] In our implementation, we adopt the latter because non-Newtonian fluids are often immiscible and can better demonstrate our fluid-fluid collision model.

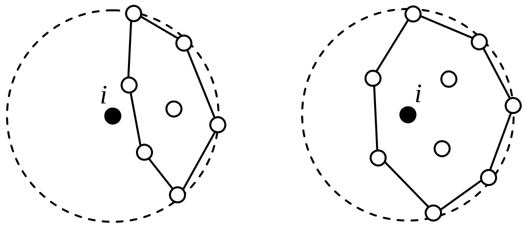
In our model, a fluid is modeled by the particles of the same type with a unique type id. The different types of the particles may have different fluid attribute values. The Navier-Stokes equation is solved for each fluid independently. At every time step during a fluid animation, our model is called to handle possible fluid-fluid collisions.

The fluid rendering follows the common practice as in the previous SPH-based fluid models [15] [17] [19] The fluid surface is represented by an iso-surface which is triangulated using the Marching Cube algorithm [14] Then, the triangular mesh is rendered with POV-Ray, which is publicly available at www.povray.org.

4 Collision Detection

Collision detections among solids and deformable objects are extensively studied in computer graphics. However, these collision detection models are not applicable to our fluid-fluid collision model. This is because solids and deformable objects are usually modeled by fixed or flexible skeletons, geometric meshes, or regular geometries such as spheres and cubes whereas our animated fluids are modeled by particles.

In our model, two fluids are in collision if a particle of one fluid is found inside the other fluid. Assume that particle i belongs to fluid A and is checked against fluid B for the collision between the two fluids. If particle i has no particles of fluid B in its neighborhood, then it is not inside of fluid B; otherwise, a convex hull is constructed from the particles of fluid B that are in the neighborhood of particle i. The convex hull is illustrated in Figure 1, where the black dots indicate particle i, the big dash circles the neighborhood, the white dots the neighboring particles of fluid B, and the segments the convex hulls. In Figure 1(a), particle i is outside the convex hull and thus it is outside fluid B, whereas in Figure 1(b), particle i is inside the convex hull and thus inside fluid *B*. In our implementation, the convex hull is constructed with the algorithm proposed by Barber et al. [1]



(a) Outside convex hull *(b) Inside convex hull*

Figure 1: Use convex hull to check if a particle of one fluid is inside another fluid.

In the collision detection between fluids *A* and *B*, all the particles of fluid *A* are checked against fluid *B*, and vice versa.

5 Collision Response

When a collision between two fluids is detected as in Figure 1(b), a particle of one fluid is mixing with the particles of the other fluid. Since fluids are modeled by particle aggregations, particle mixture is equivalent to fluid mixture. In order to animate immiscible fluids, our collision response model directly modifies the velocities and positions of the particles involved in the collision. This modification strategy is known as geometric collision response which is already used in previous collision response models such as [25] The modifications are explained in details in the following two subsections. The illustration is based on the detected collision example in Figure 1(b).

5.1 Particle Velocity Modification

The particle velocity modifications are based on a particle-particle collision model. In the example in Figure 1(b), particle *i* of fluid *A* collides with fluid *B*. It is very rare that particle *i* would collide with any particles of fluid *B*. Therefore, we create a virtual particle *j* of fluid *B* that collides with particle *i*. Particle *j* has the same position as particle *i*. Its mass and velocity are interpolated from the particles of fluid *B* in the neighborhood of particle *i*. Since all fluid particles have no physical shape, they are all represented as points. The velocities of particles *i* and *j* after the collision can be analytically solved using classical mechanics.

Assume that v_i and v_j are the velocities of particles *i* and *j*, respectively. A frame of reference moving at velocity v_j is defined. (Note that it also works similarly if the frame is moving at velocity v_i.) In such a frame, the velocities of particles *i* and *j* become $v'_i = v_i - v_j$ and $v'_j = v_j - v_j = 0$, respectively, as illustrated in Figure 2(a). Here, we assume that the collision is elastic and thus the kinetic energy and the momentum are conserved for the particles in the collision. The two conservation equations are:

$$m_i v'_i = m_i v''_i + m_j v''_j$$

$$\frac{1}{2} m_i v'^2_i = \frac{1}{2} m_i v''^2_i + \frac{1}{2} m_j v''^2_j$$

where m_i and v''_i are the mass and the after-collision velocity for particle *i*, and m_j and v''_j for particles *j*. Since the two particles have no shape, v''_i and v''_j are pointing along the line determined by v'_i, as illustrated in Figure 2(b), and they can be expressed in terms of v'_i:

$$v''_i = a_i v'_i$$

$$v''_j = a_j v'_i \ .$$

With the new expressions for v''_i and v''_j, the two conservation equations become:

$$m_i = m_i a_i + m_j a_j$$

$$m_i = m_i a_i^2 + m_j a_j^2 \ .$$

They can be easily solved for a_i and a_j:

$$a_i = \frac{m_i - m_j}{m_i + m_j}$$

$$a_j = \frac{2m_i}{m_i + m_j}$$

and, in turn, for v''_i and v''_j:

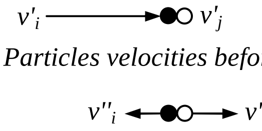
$$v''_i = \frac{m_i - m_j}{m_i + m_j} v'_i$$

$$v''_j = \frac{2m_i}{m_i + m_j} v'_i \ .$$

v''_i and v''_j are not the final after-collision velocities for particles *i* and *j* because they are solved in the moving frame at velocity v_j. In the inertial frame, the final after-collision velocities v'''_i and v'''_j for particles *i* and *j* are:

$$v'''_i = \frac{(m_i - m_j)v_i + 2m_j v_j}{m_i + m_j}$$

$$v'''_j = \frac{(m_j - m_i)v_j + 2m_i v_i}{m_i + m_j} \ .$$



(a) Particles velocities before collision.

(b) Particle velocities after collision.

Figure 2: Particle collision in the moving frame. The black and the white dots indicate the particles, and the arrow segments the velocities.

In the above collision, the momentum is conserved for particles *i* and *j*, that is,

$$\Delta M_i + \Delta M_j = 0$$

where ΔM_i and ΔM_j are the momentum changes on particles *i* and *j*, respectively, and

$$\Delta M_j = m_j(v_j''' - v_j) = \frac{2m_i m_j (v_i - v_j)}{m_i + m_j}.$$

However, the virtual particle *j* will be discarded after the collision response and thus the momentum on the whole particle system would not be conserved. To address this problem, the momentum change of particle *j* is distributed to the neighboring particles, from which particle *j*'s properties are interpolated. The weights for the distribution to the neighboring particles are the same as the weights for the interpolation. The weighted momentum change ΔM_k is added to neighboring particle *k*:

$$\Delta M_k = \frac{m_k W_k (p_j - p_k, h)}{\rho_k} \Delta M_j .$$

where $k = 1,\ldots, m$, and *m* is the number of the neighboring particles. After the distribution, the momentum is conserved for the whole particle system. Unfortunately, it would not be easy to conserve the kinetic energy at the same time when particle velocities are modified in the collision response. In our experiments, we observe the kinetic energy damping after the collision responses. This is, in terms of visual effects, consistent with the damping effect in real fluid-fluid collision phenomena.

The above velocity modifications are only for one collision as in Figure 1(b). The particle velocities modified in one collision may affect another collision. The different orders of processing the collisions may produce different results. However, such differences appear visually insignificant according to our experiments.

5.2 Particle Position Modification

In Figure 1(b), particle *i* of fluid *A* is immersed in the particles of fluid *B*. The goal of the particle position modification is to move particle *i* towards the inside of fluid *A* and out of fluid *B*. The moving direction is the normal on the interface between the two fluids and is pointing from fluid *B* to fluid *A*. The normal is computed at the position of particle *i*, and the computation is adopted from [17] Because particle *i* is inside the convex hull, the normal shooting from particle *i* must intersect with the convex hull. As a result, particle *i* is moved to the intersection and then is not immersed in the particles of fluid *B* anymore. Based on the example in Figure 1(b) with more neighboring particles, the movement of particle *i* is illustrated in Figure 3, where the dashed line indicates the fluid interface, the arrow segment the moving direction, and the arrow tip the intersection.

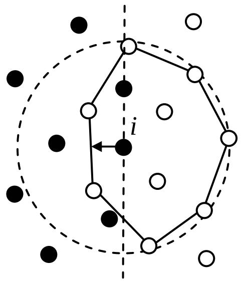


Figure 3: Move particle i onto the convex hull.

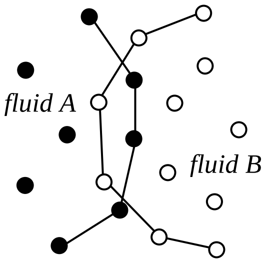


Figure 4: Some particles of fluids A and B are immersed into each other.

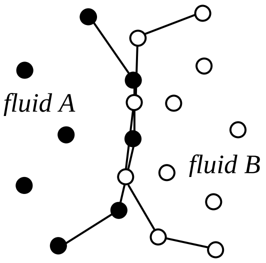


Figure 5: Fluid B is more deformed than fluid A.

When fluids *A* and *B* are in collision, their particles are immersed into each other, as illustrated in Figure 4 where the segments indicate the fluid boundaries, and the black and the white dots the particles of fluids *A* and *B*, respectively. If fluid *B*'s immersed particles are all moved out of fluid *A* first, then fluid *A*'s immersed particles are less immersed than before or not immersed into fluid *B* anymore. After all the collision responses between fluids *A* and *B* are handled, fluid *B* is more deformed than fluid *A* at the collision regions (see Figure 5 which is based on Figure 4). Such biased collision deformations on fluids *A* and *B* are not visually significant after one time step. However, if fluid *B* is always more deformed than fluid *A* at all time steps during collision, then the biased collision deformations may become quite noticeable.

An iterative process of moving immersed particles simultaneously at each time step appears to alleviate the biased deformation problem, but it is more computationally expensive. A cheaper approach is to move fluid *B*'s immersed particles first at odd time steps and to move fluid *A*'s first at even time steps, that is, to switch the order between fluids *A* and *B* for moving immersed particles. With this approach, both fluids have an equal chance to be the first, without the bias, to deform during collision.

6 Results

The animation results are produced on a 3GHz Pentium PC with 1GB of memory. Several select frames of the animations are presented in Figures 6 to 9 at the end of this paper. Figures 6 to 8 shows the artificial fluid animations and Figure 9 some animations of egg white with yolk. By default, the colliding fluids in each animation have the same initial conditions except those indicated explicitly. More animation descriptions are given in the corresponding figure captions. The animation video files are also submitted along with this paper.

As mentioned earlier, we implement the particle-based non-Newtonian fluid model [15] to work with our fluid-fluid collision model. The non-Newtonian model has a physical parameter, shear modulus M_S , to control the fluid elasticity. The physical meaning of the shear modulus is that the higher the shear modulus, the more elastically the fluid behaves. In our animations, the shear modulus is varied in order to show different fluid behaviors under the fluid-fluid collision.

Animations in Figures	*N*	*T*(seconds)
Figure 6	1200	2.3
Figure 7	1800	3.5
Figure 8	1800	3.5
Figure 9(a)	3000	5.7
Figure 9(b)	2500	4.6
Figure 9(c)	3000	5.7

Table 1: Statistics of the animations.

The statistics of the animations are summarized in Table 1, where *N* is the number of particles in each animation, and *T* the average motion computational time for each frame, excluding the fluid surface generation time and the rendering time. *T* is significantly larger than the corresponding time in another SPH-based Newtonian fluid model [17] This is because smaller time steps have to be used for non-Newtonian fluids with high shear modulus values M_S. Our fluid-fluid collision model only takes a small fraction of *T*, while the non-Newtonian fluid model [15] takes the rest. More specifically, about 20-30% of the time is spent on our fluid-fluid collision model.

7 Conclusion

In this paper, we present a new fluid-fluid collision model for particle-based immiscible fluid animation. Our model simulates the fluid-fluid collision with the two components: collision detection and collision response. The animation results demonstrate that our model not only prevents the colliding fluids from mixing with each other, but also allows one fluid to wrap around or to run through another fluid. Furthermore, our model can easily work with a generic particle-based fluid model because it requires only: (1) a particle neighbor search and (2) particles with mass and velocity. Many existing particle-based fluid models can trivially satisfy these two requirements.

The previous fluid-fluid interaction models [11] [12] [19] have been used to produce the animations of air bubbles in liquid and two immiscible fluids in a lava lamp. In these animations, one fluid is mainly immersed in the other fluid. In comparison, the implementation of our model with the non-Newtonian fluid model [15] not only animates the similar immersed fluid phenomena (Figure 9), but also animates the collisions of fluids with free surfaces (Figures 6 to 8). Thus, our model extends the modeling scope of the particle-based fluid models.

Besides immiscible fluid collision, many other interesting fluid phenomena come from mixable fluid interaction, such as pouring milk into coffee and dripping ink into water. In the future, we would like to investigate for mixable fluid animations.

Acknowledgments

This section is temporarily skipped for anonymous review. Appropriate parties will be acknowledged here if the paper is accepted.

References

[1] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa. The Quickhull Algorithm for Convex Hulls. In *ACM Transactions on Mathematical Software*, 22(4):469-483, 1996.

[2] M. Carlson, P. J. Mucha, III R. B. Van Horn, and G. Turk. Melting and Flowing. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* 2002, 167–174.

[3] M. Carlson, P. J. Mucha, and G. Turk. Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid. In *Proceedings of ACM SIGGRAPH* 2004, Los Angeles, California, August 8-12.

[4] S. Clavet, P. Beaudoin, and P. Poulin. Particle-based Viscoelastic Fluid Simulation. In *Proceed-*

ings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2005.

[5] M. Desbrun and M.-P. Cani. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Computer Animation and Simulation* 1996, 61–76.

[6] D. P. Enright, S. R. Marschner, and R. P. Fedkiw. Animation and rendering of complex water surfaces. In *Proceedings of ACM SIGGRAPH* 2002, 736–744.

[7] N. Foster and R. P. Fedkiw. Practical animation of liquids. In *Proceedings of ACM SIGGRAPH* 2001, 23-30.

[8] O. Genevaux, A. Habibi, and J.-M. Dischler. Simulating fluid-solid interaction. *Graphics Interface* 2003, 31-38.

[9] T. G. Goktekin, A. W. Bargteil, and J. F. O'Brien. A Method for Animating Viscoelastic Fluids. In *Proceedings of ACM SIGGRAPH* 2004, Los Angeles, California, August 8-12, 2004.

[10] E. Guendelman, A. Selle, F. Losasso, and R. Fedkiw. Coupling Water and Smoke to Thin Deformable and Rigid Shells. In *Proceedings of ACM SIGGRAPH* 2005, Los Angeles, California, July 2005.

[11] J.-M. Hong and C.-H. Kim. Animation of Bubbles in Liquid. *Computer Graphics Forum* 22, 3 (September 2003).

[12] J.-M. Hong and C.-H. Kim. Discontinuous Fluids. In *Proceedings of ACM SIGGRAPH* 2005, Los Angeles, California, July 2005.

[13] R. Keiser, M. Muller, B. Heidelberger, M. Teschner, and M. Gross. Contact Handling for Deformable Point-Based Objects. *Proc. Vision, Modeling, Visualization VMV'04*, Stanford, USA, pp. 315-322, Nov. 16-18, 2004.

[14] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface reconstruction algorithm. In *Computer Graphics (Proceedings of ACM SIGGRAPH 87)*, Vol. 21, No. 4, 163-169, 1987.

[15] H. Mao and Y-H Yang, A particle-based model for non-Newtonian fluid animation. Technical report TR05-21, Department of Computing Science, University of Alberta, August, 2005.

[16] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30:543–574, 1992.

[17] M. Müller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of Eurographics/SIGGRAPH Symposium on Computer Animation* 2003, 154–159.

[18] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa. Point based animation of elastic, plastic and melting objects. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* 2004, p141-151, 2004.

[19] M. Müller, B. Solenthaler, R. Keiser, and M. Gross. Particle-Based Fluid-Fluid Interaction. In *Proceedings of ACM SIGGRAPH Symposium on Computer Animation* 2005.

[20] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross. Interaction of Fluids with Deformable Solids. In *Proceedings of Computer Animation & Social Agents* CASA'04, Geneva, Switzerland, pp. 159-171, July 7-9, 2004.

[21] S. Premoze, T. Tasdizen, J. Bigler, A. Lefohn, and R. Whitaker. Particle-based simulation of fluids. *Computer Graphics Forum* 22, 3 (September 2003), 401–410.

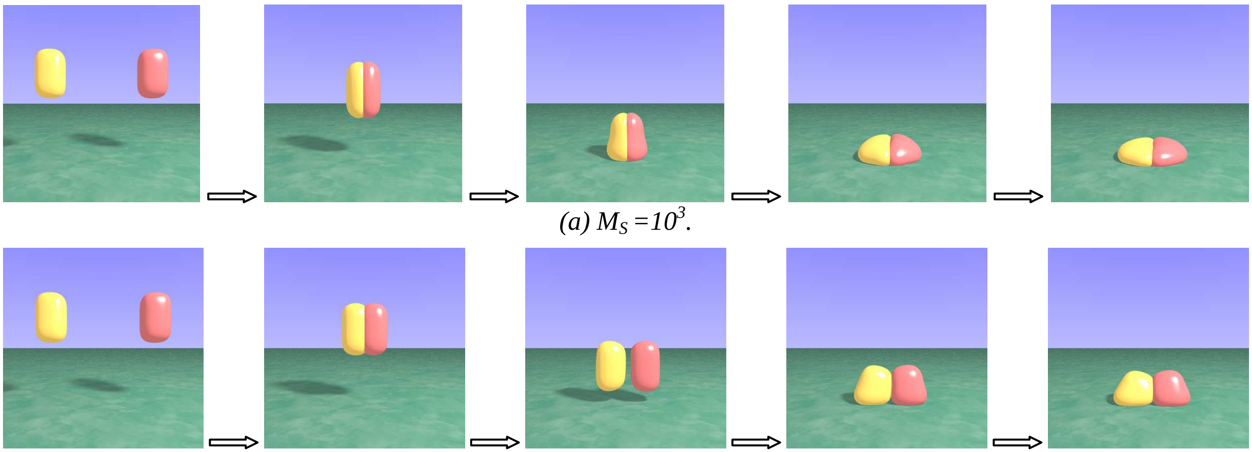
[22] J. Stam. Stable fluids. In *Proceedings of ACM SIGGRAPH* 1999, 121–128.

[23] D. Stora, P.-O. Agliati, M.-P. Cani, F. Neyret, and J.-D. Gascuel. Animating lava flows. *Graphics Interface* 99, 203–210.

[24] T. Takahashi, H. Fujii, A. Kunimatsu, K. Hiwada, T. Saito, K. Tanaka, and H. Ueki. Realistic Animation of Fluid with Splash and Foam. *Computer Graphics Forum* 22, 3 (September 2003)

[25] P. Volino and N. Magnenat-Thalmann. *Virtual Clothing Theory and Practice,* Springer-Verlag, 2000.

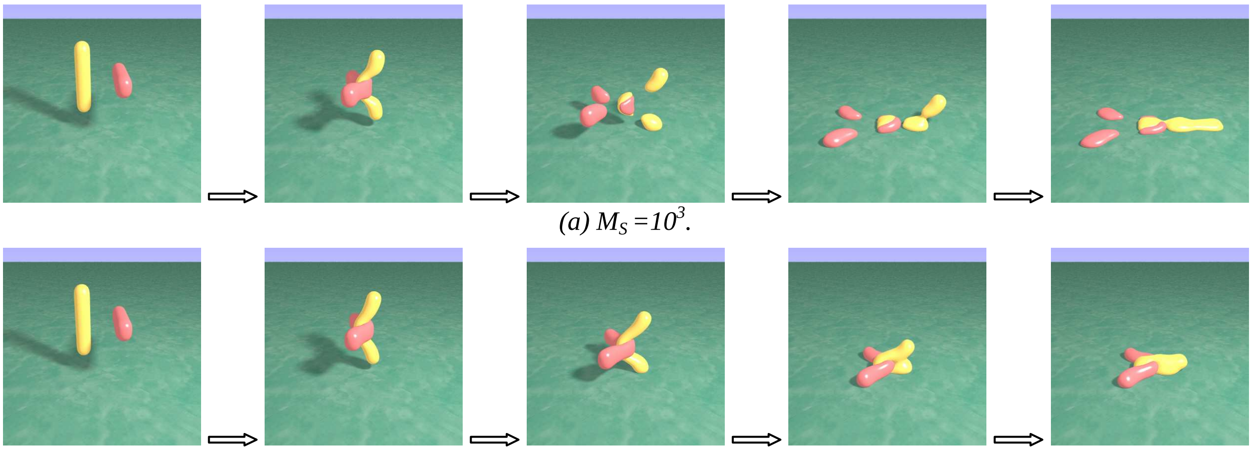
[26] H. Wang, P. J. Mucha, and G. Turk. Water Drops on Surfaces. In *Proceedings of ACM SIGGRAPH* 2005, Los Angeles, California, July 2005.



(a) $M_S = 10^3$.

(b) $M_S = 10^4$. Because of the high shear modulus, the two fluids even bounce away a little from each other.

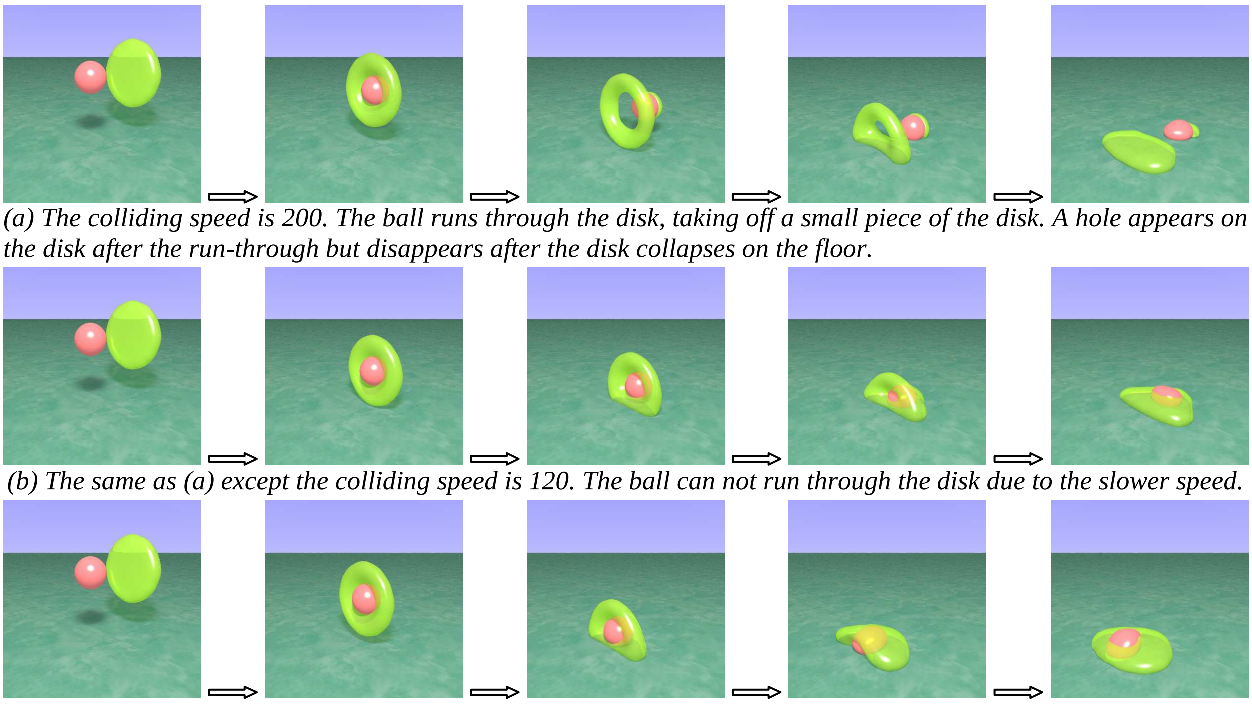
Figure 6: Two fluids are colliding with each other, and their motions are altered by the collision.



(a) $M_S = 10^3$.

(b) $M_S = 10^4$. Because of the higher shear modulus, the fluids do not split into as many pieces as those in (a).

Figure 7: Two fluid bars are cross-colliding with each other. The motions of the colliding parts are altered more by the collision than the motions of the other parts.

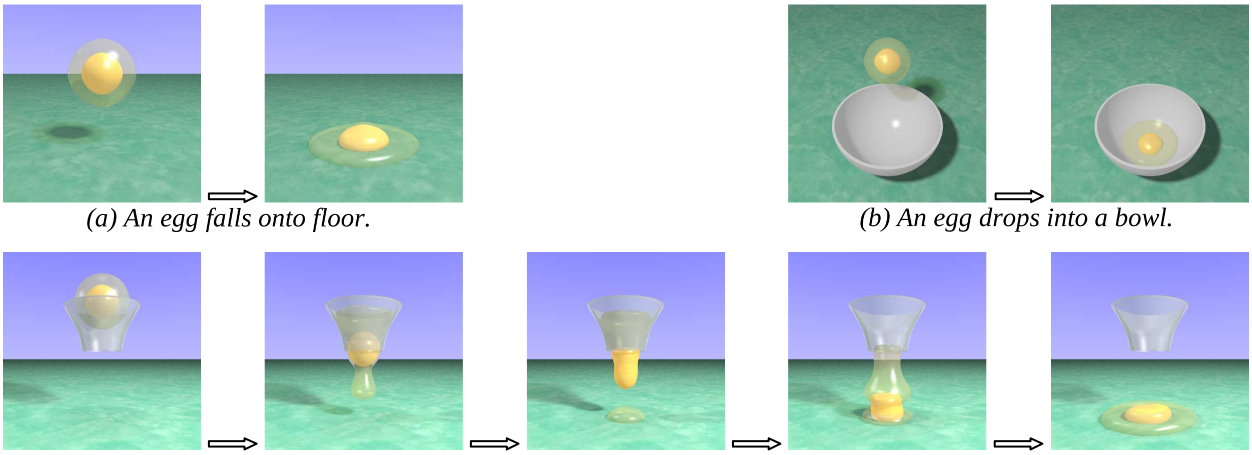


(a) The colliding speed is 200. The ball runs through the disk, taking off a small piece of the disk. A hole appears on the disk after the run-through but disappears after the disk collapses on the floor.

(b) The same as (a) except the colliding speed is 120. The ball can not run through the disk due to the slower speed.

(c) The colliding speed is 200. The ball can not run through the disk due to the reason given in Figure 8's caption.

Figure 8: A fluid ball is running into a fluid disk. $M_S = 10^4$ for the ball and $M_S = 10^3$ for the disk. In (a) and (b), the particles for the ball have 10 times the mass as the particles for the disk. In (c), the particles for the ball and the disk have the same mass. The initial conditions in (c) are the same as those in (a) except the difference about the particle mass. As a result, in (c), the disk is heavy enough to prevent the ball from running through.



(a) An egg falls onto floor.

(b) An egg drops into a bowl.

(c) An egg falls through a funnel onto floor.

Figure 9: An egg is modeled as two fluids: egg white and yolk. Yolk is mainly immersed in egg white. M_S = 500 for egg white and M_S = 10^4 for yolk. The particles for the yolk have 10 times the mass as the particles for the egg white.