A Particle-Based Model for Non-Newtonian Fluid Animation

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

Non-Newtonian fluids exist in nature, such as blood and egg white, and in man-made products, such as toothpaste and ketchup. They exhibit more interesting and complex behaviors than traditional Newtonian fluid such as water. In this paper, we propose a new particle-based model for the animation of non-Newtonian fluids. The new model has two contributions. The first contribution is a new particle dynamics method. It takes into account rotational frame indifference in stress tensor computation while previous non-Newtonian fluid models in computer graphics do not. Therefore, our model is more accurate in animating rotational non-Newtonian fluid motions. In addition, the particle dynamics method includes a new SPH-based Poisson equation for pressure to enforce fluid incompressibility. The second contribution is a new particle re-sampling method. It is observed that irregular particle distribution causes inaccurate fluid modeling and consequently particles may form unrealistic clusters. In previous Newtonian fluid models, some techniques have been proposed to deal with the particle clustering problem. However, we show that they are not effective for fluid stretching, which is a common motion involved in many non-Newtonian fluid phenomena. To address this problem, in our particle re-sampling method, particles are down-sampled and then up-sampled such that a well distribution of particles is attained. Overall, many experimental animations are produced, including immiscible fluid-fluid interaction and fluid interaction with rigid-body. These animations demonstrate our contributions and show that our model is able to animate interesting non-Newtonian fluid behaviors.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling – Physically Based Modeling; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism – Animation; I.6.8 [**Simulation and Modeling**]: Types of Simulation – Animation

General Terms: Algorithm
Additional Key Words and Phrases: particle-based model, viscoelastic fluid, non-Newtonian fluid, Smoothed Particle Hydrodynamics (SPH), particle re-sampling, Delaunay triangulation, convex hull.

1. INTRODUCTION

Non-Newtonian fluids are a rich class of fluids. In fluid mechanics, they are defined as fluids for which the stress tensor cannot be expressed as a linear, isotropic function of the velocity gradient [Owens and Phillips 2002]. In our daily lives, many fluids that we encounter are non-Newtonian, e.g. blood, egg white, toothpaste, ketchup, and so on. These fluids exhibit more interesting and complex behaviors than traditional Newtonian fluids, such as water. In computer graphics, many proposed models are based on the Newtonian fluid formulation, which can animate various Newtonian fluid behaviors and produce very appealing results. Viscoelastic fluids are examples of non-Newtonian fluids. One of the earliest viscoelastic models is proposed by Terzopoulos and Fleischer [1988]. More recently, two such models are proposed by Goktekin et al. [2004] and Clavet et al. [2005]. For clarity, viscoelastic models are referred to as non-Newtonian models hereafter.

In this paper, we propose a new particle-based fluid model for the animation of non-Newtonian fluids. The dynamics formulation of the proposed model is based on

Smoothed Particle Hydrodynamics (SPH), which is a popular fluid modeling formulation in Computational Fluid Dynamics (CFD) [Monaghan 1992 and Ellero et al. 2002] and is also widely used in computer graphics [Desbrun and Cani 1996, Desbrun and Cani 1999, Stora et al. 1999, Muller et al. 2003, Muller et al. 2004, and Clavet et al. 2005].

The new model has two contributions. The first contribution is a new particle dynamics method for non-Newtonian fluids. The previous non-Newtonian models by Terzopoulos and Fleischer [1988] and Clavet et al. [2005] are particle-based models and, in their particle dynamics methods, the stress tensor for non-Newtonian fluids is based on a linear combination of elastic spring, viscous dashpot, and plastic yield condition. Meanwhile, the grid-based non-Newtonian model by Goktekin et al. [2004] computes the stress tensor based on a linear Maxwell model and the von Mises plastic yield condition. These models violate frame indifference when the fluids are in rotational motions. In contrast, our particle dynamics method is based on a corotational Maxwell model [Ellero et al. 2002], and takes into account frame indifference for arbitrary fluid motions. An experimental result is presented later to demonstrate that our model is more accurate in simulating rotational motions. In addition, our particle dynamics method includes a new SPH-based Poisson equation for pressure to enforce fluid incompressibility. The Poisson equation for pressure has been used in the grid-based fluid model [Foster and Fedkiw 2001] and the particle-based fluid model [Premoze et al. 2003]. It is more suitable to enforce fluid incompressibility than the state equation of gas, which is used by the previous SPH-based fluid models [Desbrun and Cani 1996, Stora et al. 1999, and Muller et al. 2003]. We adopt the fluid incompressibility method from [Premoze et al. 2003] completely except that we translate the Poisson equation into the SPH formulation and propose a new SPH-based pressure Laplacian evaluation.

The second contribution is a new particle re-sampling method. In the particle-based fluid models, particles are distributed in the fluid. The flowing and deformation of the fluid can cluster particles in some regions and spread them in others. Since the fluid attributes are computed according to the particle distribution in the neighborhood, the computation is not accurate in the sparse-particle regions. Consequently, particles may form unrealistic clusters in the animation results. To deal with the particle clustering problem, Desbrun and Cani [1996] propose a repulsive pressure force between particles such that particles do not get too close to each other. In our experiments, the repulsive pressure force can not prevent the particle clustering problem in the animation of fluid stretching which is a common motion of non-Newtonian fluids. Instead, noticeable artifacts are observed in our experimental results, which are shown later in the paper. They [Desbrun and Cani 1999] also propose a particle re-sampling method in which particles are re-sampled according to the particle pressure variations. This method is applicable for adaptive space discretization and is not able to solve the particle clustering problem as discussed in this paper. Premoze et al. [2003] propose a particle position reconfiguration method. In this method, particles that belong to a fixed boundary or inlet or outlet boundary should go back to their original positions. Meanwhile, they make sure that the particles on the surface of free moving boundary are at equal distance apart. For other particles, the positions are arbitrarily determined. This method does not ensure sufficient particle concentration in the sparse-particle regions and thus irregular particle distribution still occurs. We cannot repeat their method because there is no information on how to redistribute surface particles to be at equal distance apart and how to arbitrarily determine other inside particle positions. The previous particle-based non-Newtonian models [Terzopoulos and Fleischer 1988 and Clavet et al. 2005] do not demonstrate the animations of non-Newtonian fluid stretching and thus the particle clustering problem is not a concern in these two models. To address the particle clustering problem, we propose a new particle re-sampling method, which consists of a down-sampling method and an up-sampling method. The down-sampling method merges two particles if they are

too close to each other. The up-sampling method inserts new particles in the sparse-particle regions, which are detected by the Delaunay triangulation [Watson 1981 and Bowyer 1981]. As a result, the whole re-sampling method maintains a well distribution of particles.

The paper is organized as follows. Related works are briefly discussed in Section 2. An overview of the proposed model is given in Section 3. The particle dynamics method and the particle re-sampling method are described in detail in Section 4 and 5, respectively. The animation results are presented in Section 6. Finally, conclusion and future work are given in Section 7.

2. RELATED WORKS

It is widely accepted in computer graphics that the three-dimensional Navier-Stokes (NS) equation is the most comprehensive dynamics model for fluid animation. Here, we briefly discuss previous fluid models that use the NS equation, because our model does too. Those fluid models can be categorized as grid-based models and particle-based models. The former uses the Eulerian version of the NS equation, and the latter the Lagrangian version of the NS equation.

In grid-based models, a three-dimensional grid structure is initialized to cover the animation space in which the fluid exists or may move into. The Eulerian version of the NS equation is solved on the grid structure. Early work of Foster and Metaxas [1996] demonstrates that various 3D fluid behaviors can be animated by solving the NS equation. However, the animation becomes unstable when the time step is large. To alleviate this problem, Stam [1999] introduces a semi-Largrangian method to obtain a stable solution with large time steps. As a result, the animation efficiency is improved. In order to track complex free fluid surface, a hybrid method of combining marker particles and the level set method is introduced by Foster and Fedkiw [2001] and improved by Enright et al. [2002]. The hybrid method can achieve appealing realism of fluid splashing and swirling. Carlson et al. [2002] present a model to animate fluids by varying viscosity in the NS equation. They propose an implicit integration method to deal with the stability problem that stems from high viscosities, and produce nice melting and flowing animations for highly viscous fluids. Interesting fluid animations are also produced by other grid-based fluid models. The model in [Hong and Kim 2003] produces animations of bubbles in fluid by embedding surface tension in the NS equation, while the model in [Takahashi et al. 2003] animates fluid with splash and foam by incorporating SPLASH and FOAM particles. For fluid-solid interactions, Genevaux et al. [2003] propose an interface between the fluid and the solid, while Carlson et al. [2004] treat rigid solids as special fluids with rigid motions. Goktekin et al. [2004] describe a model for animating viscoelastic fluids, which are treated as materials intermediate between elastic solids and viscous fluids. In their model, the stress tensor is embedded in the NS equation to account for the viscoelastic effects, and its computation is based on a combination of the linear Maxwell model and the von Mises plastic yield condition. By varying the yield value and the elastic decay rate, their model produces a wide range of interesting fluid behaviors.

In particle-based fluid models, a particle represents a volume of the animated fluid. The Lagrangian version of the NS equation is solved on the particles. Desbrun and Cani [1996] use particles to animate highly deformable bodies based on SPH. In order to improve the animation efficiency, an adaptive SPH model is proposed in [Desbrun and Cani 1999]. Muller et al. [2003] propose a SPH-based model that can interactively animate fluid splashing and swirling. In this model, the realism of the fluid motion is traded for the interactive speed by using a limited number of particles. SPH is also utilized to animate lava flows in [Stora et al. 1999], fluid-solid interaction in [Muller et al. 2004], and fluid-fluid interaction in [Muller et al. 2005]. The Moving-Particle Semi-implicit (MPS) is another particle-based model in CFD and is introduced into computer

graphics by Premoze et al. [2003]. This model can produce appealing animations of fluid splashing and swirling which are comparable to the current best results by the grid-based fluid models. One of the earliest viscoelastic fluid models in computer graphics is proposed by Terzopoulos and Fleischer [1988], and it is a particle-based model. In this model, the viscoelasticity is modeled as a linear combination of elastic spring, viscous dashpot, and plastic yield condition. Recently, Clavet et al. [2005] model the viscoelasticity with a similar linear combination as in [Terzopoulos and Fleischer 1988]. Their model is able to achieve interactive animation speed.

Non-Newtonian fluids have been extensively studied in CFD, and the study of them is also known as Rheology. For more background information, we refer interested readers to [Renardy 2000, and Owens and Phillips 2002].

3. MODEL OVERVIEW

During each time step in our model, the basic operations are shown in Figure 1. The boundary constraints, if any, are enforced using the method in [Muller et al. 2004]. The fluid-fluid interactions follow the model in [Muller et al. 2005]. The fluid rendering follows the common practice as in [Muller et al. 2003 and Muller et al. 2004]. The fluid surface is represented by an iso-surface which is triangulated using the Marching Cube algorithm [Lorensen and Cline 1987]. And the triangular mesh is rendered using ray-tracing software.

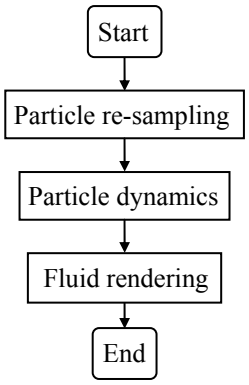


Fig. 1. Basic operations during each time step.

4. PARTICLE DYNAMICS

The proposed model is a particle-based fluid model. The particle motions are governed by the Lagrangian version of the NS equation as follows:

$$\frac{dv}{dt} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot T + \frac{\mu}{\rho}\nabla^2 v + \frac{1}{\rho} f \tag{1}$$

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. In contrast, the grid-based fluid models compute fluid motions using the Eulerian version of the NS equation and they have to compute an extra advection term. More detailed comparison between the Lagrangian and Eulerian NS equations can be found in [Muller et al. 2003].

Equation (1) is evaluated at each particle location using the SPH formulation. The evaluation of the viscosity term is the same as that given in [Muller et al. 2003] and the external force term can be computed trivially. The evaluations of the pressure and the stress tensor terms are described in detail in Section 4.2 and 4.3, respectively.

4.1 SMOOTH PARTICLE HYDRODYNAMICS (SPH)

Smoothed Particle Hydrodynamics (SPH) is a particle-based formulation for modeling fluids and is based on the interpolation theory. For completeness, a brief description of the SPH formulation is given here. The interested reader is referred to [Monaghan 1992] for a detailed introduction.

SPH divides the fluid into a set of elements called particles, which are used to carry fluid attributes. A scalar attribute value $A(r)$ at location r is interpolated by the values of particles within a local neighborhood:

$$A(r) = \sum_{j=1}^{n} A_j \frac{m_j}{\rho_j} W(r - r_j, h) \tag{2}$$

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 location, h the neighborhood radius, and W the interpolation weighting function called kernel. If not specified otherwise, the traditional spline kernel [Monaghan 1992] is used in this paper:

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

where $q = 2|r\text{-}r_j|/h$.

Using Equation (2), the density ρ_i of particle i is evaluated as:

$$\rho_i = \sum_{j=1}^{n} m_j W(r_i - r_j, h) \tag{4}$$

where r_i is the location of particle i.

4.2 PRESSURE

The pressure term in the NS equation is evaluated at each particle i as in [Monaghan 1992]:

$$-\frac{1}{\rho_i} \nabla p_i = \sum_{j=1}^{n} m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla_i W(r_i - r_j, h) \tag{5}$$

where p_i and p_j are the pressure of particle i and j, respectively. The spiky kernel in [Desbrun and Cani 1996] is adopted in order to have a repulsive pressure force:

$$W(r - r_j, h) = \frac{15}{\pi h^6} \begin{cases} (h - R)^3 & 0 \le R \le h \\ 0 & otherwise \end{cases} \tag{6}$$

where $R = |r\text{-}r_j|$.

In [Premoze et al. 2003], the Moving Particle Semi-implicit (MPS) method is presented to enforce the fluid incompressibility. This method is adopted in our model completely except the following two changes. The Poisson equation for pressure is translated from the MPS formulation to the SPH formulation as follows:

$$\frac{1}{\rho_i} \nabla^2 p_i = \frac{\rho_{org} - \rho_i}{\rho_{org} \Delta t^2} \tag{7}$$

where ρ_{org} is the original fluid density to enforce. In addition, we propose a pressure Laplacian evaluation on each particle i under the SPH formulation:

$$\frac{1}{\rho_i} \nabla^2 p_i = \sum_{j=1}^{n} (p_i - p_j) \frac{m_i + m_j}{2(\rho_i + \rho_j)^2} W(r_i - r_j, h) \,. \tag{8}$$

With these two changes, an equation system can be formed from Equation (7) evaluated on all particles and the unknown variables are the particle pressures. With Equation (8),

the system coefficient matrix is symmetric and positive-definite, if there is at least one surface particle, which can be trivially satisfied. The pressure for the surface particles is set to be the constant air pressure, and thus the equations for the surface particles can be removed from the system. The solution can be efficiently obtained using a Preconditioned Conjugate Gradient method [Press et al. 1992].

4.3 STRESS TENSOR

For non-Newtonian fluids, the stress tensor is a nonlinear function of the velocity gradient. A common method to compute the stress tensor is to integrate the tensor-rate as in [Goktekin et al. 2004 and Owens and Phillips 2002]. The tensor-rate in our model is based on a nonlinear corotational Maxwell model [Ellero et al. 2002]:

$$\frac{dT}{dt} = \Omega + \frac{\mu_e}{2} D' - \frac{1}{\lambda} T \tag{9}$$

where T is the stress tensor, μ_e the elasticity constant, λ the relaxation time. Ω is the rotational tensor which is the coordinate transformation between the global inertial frame and a coordinate frame rotating with the instantaneous fluid angular velocity at the particle. Its expression is:

$$\Omega = \frac{1}{2}(T \bullet \omega - \omega \bullet T) \tag{10}$$

where

$$\omega = \nabla v - (\nabla v)^T \tag{11}$$

and each element $\omega^{\alpha\beta}$ is

$$\omega^{\alpha\beta} = \frac{\partial v^\beta}{\partial r^\alpha} - \frac{\partial v^\alpha}{\partial r^\beta} . \tag{12}$$

D' is the traceless strain tensor:

$$D' = D - \frac{Trace(D)}{3} I \tag{13}$$

where

$$D = \nabla v + (\nabla v)^T \tag{14}$$

and each element $D^{\alpha\beta}$ is

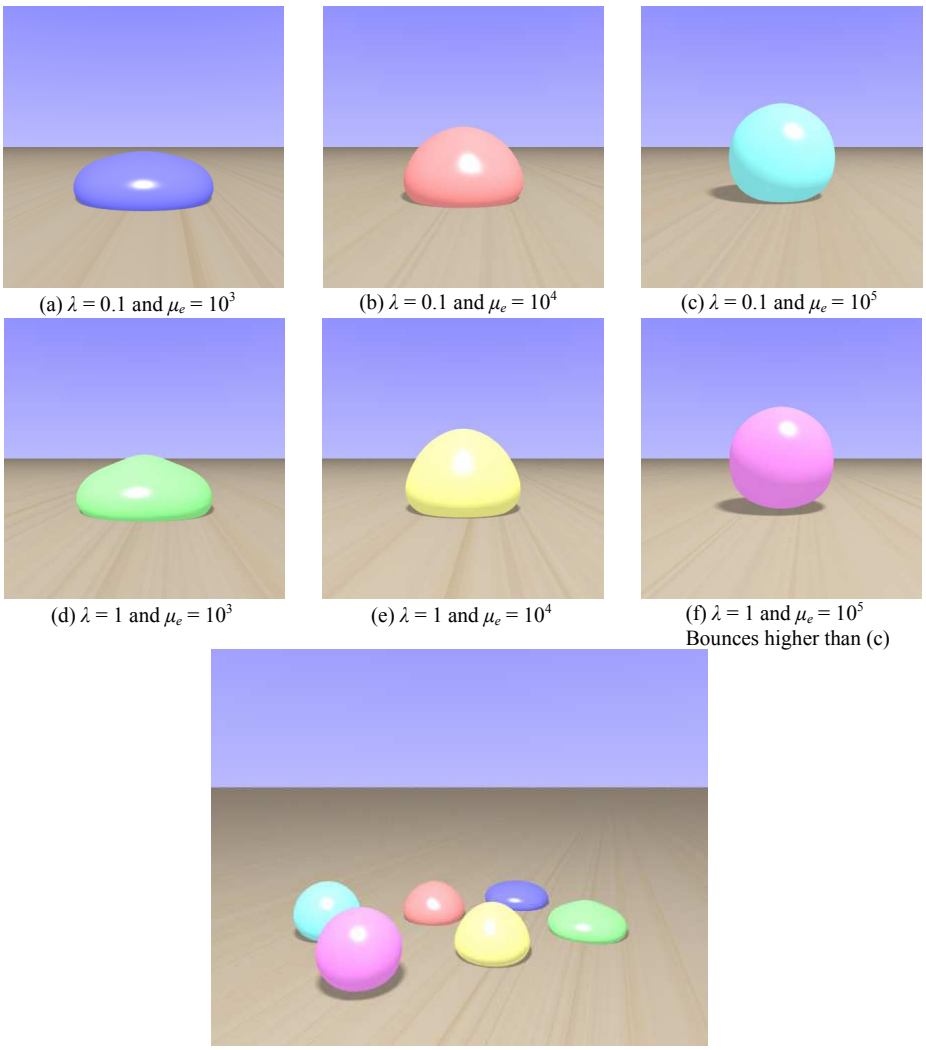
$$D^{\alpha\beta} = \frac{\partial v^\beta}{\partial r^\alpha} + \frac{\partial v^\alpha}{\partial r^\beta} . \tag{15}$$

In Equation (12) and (15), the Greek indices α and β denote 3D spatial coordinates. Under the SPH formulation, the partial velocity derivative on each particle is

$$\frac{\partial v^a}{\partial r^b} = \sum_{j=1}^{n} \frac{m_j}{\rho_j} (v_j^a - v^a) \frac{\partial W(r - r_j, h)}{\partial r^b} \tag{16}$$

where $a, b = \alpha, \beta$.

In Equation (9), the relaxation time λ is a characteristic constant for a non-Newtonian fluid. It characterizes the length of "memory" in which the non-Newtonian fluid has for its previous shape. $1/\lambda$ has the similar physical meaning as the material's decay rate in [Goktekin et al. 2004]. Basically, the larger the λ, the more strongly a non-Newtonian fluid tries to restore to its previous shape. In the normal room temperature, the relaxation time of water is about 10^{-12}s, practically no "memory" of its previous shape, and that of glass is in excess of 28 hours. The elasticity constant μ_e is a factor to characterize fluid resistance to deformation. The higher the μ_e, the stronger the resistance is. More information about λ and μ_e can be found in [Owens and Phillips 2002]. In Figure 2, six fluid balls drop onto the floor and their motions are consistent with their respective λ and μ_e.



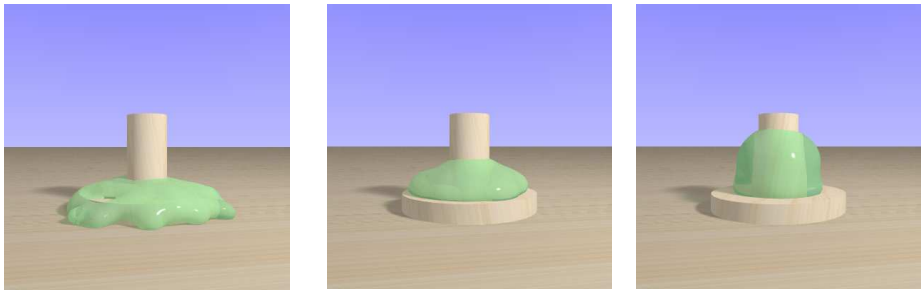
(a) $\lambda = 0.1$ and $\mu_e = 10^3$ (b) $\lambda = 0.1$ and $\mu_e = 10^4$ (c) $\lambda = 0.1$ and $\mu_e = 10^5$

(d) $\lambda = 1$ and $\mu_e = 10^3$ (e) $\lambda = 1$ and $\mu_e = 10^4$ (f) $\lambda = 1$ and $\mu_e = 10^5$
Bounces higher than (c)

(g) Complete view of six fluid balls.

Fig. 2. Six fluid balls with different values of λ and μ_e fall on floor. Shown are the pictures of the balls at frame 87. The two balls in (c) and (f) with high elasticity $\mu_e = 10^5$ even bounce up.

The tensor-rate proposed by Goktekin et al. [2004] is similar to the tensor-rate in Equation (9). The major difference is that their tensor-rate does not have the rotational tensor Ω which takes account of the rotational frame indifference. As explained in the introduction, the other two particle-based models [Terzopoulos and Fleischer 1988 and Clavet et al. 2005] do not account for the rotational tensor Ω either. The experimental results in Figure 3 demonstrate that our model with the rotational tensor Ω is more accurate in simulating the rotational motions. The rod-climbing is one of the most striking non-Newtonian fluid phenomena and involves intensive rotational motion [Renardy 2000, and Owens and Phillips 2002]. In the experiment, a rotating rod is inserted into a pool of non-Newtonian fluid. The rotating motion causes a tension along the concentric streamlines, which leads to a force pushing the fluid inward. Consequently, the free surface rises and the fluid climbs up the rod. The rod-climbing is animated using our model. Two of the animation frames are shown in Figure 3(b) and (c). A comparing animation without the rotational tensor Ω in Equation (9) is also produced, in which the rod-climbing can not be animated, and a frame is shown in Figure 3(a).



(a) Frame 400: no climbing without Ω (b) Frame 400: climbing with Ω (c) Frame 1500: climbing with Ω

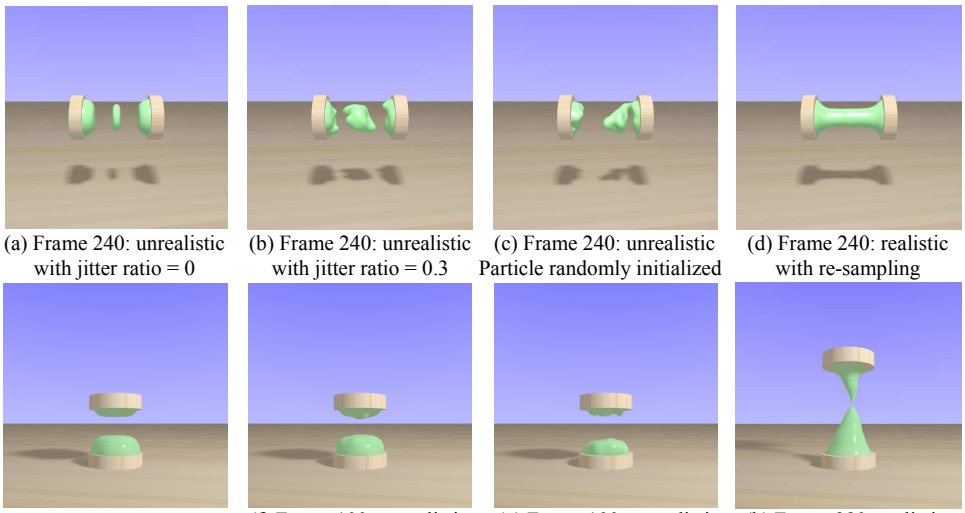
Fig. 3. Comparison of fluid rotations with and without rotational tensor Ω. (b) and (c) are in one animation

The second term in Equation (9) is the traceless strain tensor while, in [Goktekin et al. 2004], the trace is taken out from the stress tensor T. Since the tensor-rate is continuously integrated to compute the stress tensor, using either the traceless strain tensor or the traceless stress tensor causes insignificant difference in the visual results. In addition, the von Mises plastic yield condition used in [Goktekin et al. 2004] can be trivially embedded into Equation (9) by replacing T with T':

$$T' = \frac{T}{\| T \|} \max(0, \| T \| - \gamma) \tag{17}$$

where $\|T\|$ is the Frobenius norm of the stress tensor T, and γ the plastic yield value. It is noted that, if γ=0, then T=T'. For the properties of the von Mises plastic yield condition and its impact to a non-Newtonian fluid, the reader is referred to [Goktekin et al. 2004] for detailed discussions.

5. PARTICLE RE-SAMPLING



(a) Frame 240: unrealistic with jitter ratio = 0
(b) Frame 240: unrealistic with jitter ratio = 0.3
(c) Frame 240: unrealistic Particle randomly initialized
(d) Frame 240: realistic with re-sampling
(e) Frame 100: unrealistic with jitter ratio = 0
(f) Frame 100: unrealistic with jitter ratio = 0.3
(g) Frame 100: unrealistic Particle randomly initialized
(h) Frame 280: realistic with re-sampling

Fig. 4. Comparison of unrealistic and realistic fluid stretching: eight example animations. Let d as the grid cell size. The jitter ratio is the maximum jittered distance over d. (a), (b), and (c) are unrealistic stretching without the particle re-sampling. (a) Particles initialized on grid; (b) Particles initialized on grid with jitter ratio = 0.3; (c) Particle randomly initialized; (d) Particles initialized on grid and realistic stretching with the particle re-sampling. (e), (f), and (g) are unrealistic stretching without the particle re-sampling. (e) Particles initialized on grid; (f) Particles initialized on grid with jitter ratio = 0.3; (g) Particle randomly initialized; (h) Particles initialized on grid and realistic stretching with the particle re-sampling. No gravity in (a), (b), (c), and (d), while gravity is enabled in (e), (f), (g), and (h).

As mentioned in the introduction, the irregular particle distribution causes inaccurate fluid modeling in the sparse-particle regions. As a result, particles may form unrealistic clusters. This particle clustering problem is tackled in the previous particle-based models [Desbrun and Cani 1996 and Premoze et al. 2003]. Desbrun and Cani [1996] propose using a repulsive pressure force and Premoze et al. [2003] propose random particle positions. In our efforts to animate non-Newtonian fluids, we observe that the previous approaches work well for the animations of violent fluid motions, such as splashing and spraying, because the randomness in the motion appears to suppress the problem. However, those approaches do not work well for the animations of fluid stretching, which is a common motion involved in many non-Newtonian fluid phenomena. Figure 4 shows eight examples of fluid stretching. In all the examples, the repulsive pressure force in [Desbrun and Cani 1996] is used. In [Premoze et al. 2003], no detail was given on how to update particle positions randomly at each time step. To capture the idea of particle randomness, we initialize particle positions randomly in four of the examples. It can be seen that fluid stretching is not realistic in the six examples without particle re-sampling while it is realistic in the two examples with particle re-sampling.

The particle re-sampling method consists of a down-sampling method and an up-sampling method. The major contribution lies in the up-sampling method. The down-sampling method is trivial. When two particles are found closer than a threshold distance ε_d, they are deleted. Meanwhile, a new particle is created at the center of mass of the two deleted particles and inherits their total mass. The new particle's velocity and stress tensor are linearly interpolated from those of the two deleted particles. The down-sampling ensures that particles are maintained at the minimum distance ε_d. At each re-sampling step, the down-sampling method is executed before the up-sampling method. This execution order is made because the up-sampling operation does not result in particle distance less than ε_d, which can be seen later in Section 5.1.

The up-sampling method is based on two well-known computational techniques: Delaunay triangulation and convex hull. The basic idea of the method is very simple, and consists of four operations: (1) Detect sparse-particle region using Delaunay triangulation; (2) Insert a new particle into the sparse-particle region; (3) Detect fluid boundary using the convex hull such that new particle is not inserted outside of the fluid; (4) Interpolate particle attributes on the new particles. The operations (1) and (2) are explained in detail in Section 5.1 since both are involved with the Delaunay triangulation, and the operations (3) and (4) are in Section 5.2 and 5.3, respectively. For clarity, unless as stated explicitly, the geometric concepts and operations are described and illustrated only in 2D, but they can be easily applied to 3D.

Delaunay triangulation has been utilized to reconstruct surface from point cloud in [Alexa et al. 2003]. In that application, 2D Delaunay triangulation is used to up-sample points on a 2D projection plane for the reconstructed surface. To our best knowledge, we are the first to combine Delaunay triangulation and convex hull in 3D for the particle up-sampling in a fluid model.

5.1 DELAUNAY TRIANGULATION

Delaunay triangulation is an aggregate of space-filling disjoint triangles in 2D (or tetrahedra in 3D) that are constructed from the given points. The most distinguished concept in the Delaunay triangulation is the Delaunay condition that no given point falls inside the circum-circles of any triangles in 2D (or circum-spheres of any tetrahedra in 3D). As a brief introduction, a simple example is illustrated in Figure 5, where two Delaunay triangles are constructed from 4 points, *A*, *B*, *C*, and *D* which are denoted by black dots. Point *A* is not inside the circum-circle of triangle *BCD* and point *D* not inside that of triangle *ABC*. The circum-circles are denoted by the dashed circles. The construction of the Delaunay triangulation has been extensively studied in computational

geometry. In our implementation, we follow the classical incremental construction algorithm in [Bowyer 1981 and Watson 1981].

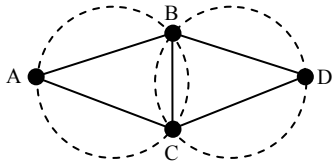


Fig. 5. Example of Delaunay triangulation.

If the Delaunay triangulation is constructed on the fluid particles, the sparse-particle regions can be easily detected according to the Delaunay condition. In particular, if the circum-circle of a Delaunay triangle is larger than the radius R_{min}, then a sparse-particle region is found because no particle is inside that circum-circle. To up-sample in this region, a new particle is inserted at the center of the circum-circle. This Delaunay-based up-sampling method guarantees minimum particle concentration that there exists at least one particle in any region of radius larger than R_{min}. The smaller the R_{min} is, the higher the particle concentration. The overall re-sampling method thus maintains a well distribution of particles: the threshold distance ε_d in the down-sampling method specifies the minimum particle separation while the threshold radius R_{min} the maximum separation. ε_d must be less than R_{min} such that the up-sampling operation does not result in particle distance less than or equal to ε_d; otherwise, there would be unnecessary system oscillation between particle insertion and deletion. After some trials, ε_d is chosen as 0.6**D*, and R_{min} as 0.95**D* in our model, where *D* is the initial distance between particles, which are uniformly distributed in a grid.

5.2 CONVEX HULL

Like Delaunay triangulation, convex hull is also a well-known computational geometric concept. The convex hull of a set of points is the smallest convex set that encloses the points. Usually, the convex hull is a closed series of line segments in 2D (a closed triangular mesh in 3D). A simple example is illustrated in Figure 6, where a convex hull is the closed series of line segments *AB*, *BC*, *CD*, *DE*, and *EA*, and encloses eight points, three inside and five on the hull, which are denoted by black dots. The convex hull construction is also well studied. In our implementation, we use the QuickHull algorithm [Barber et al. 1996].

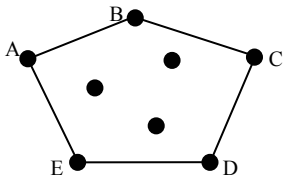


Fig. 6. Example of convex hull.

If a new particle is inserted during the Delaunay-based up-sampling, it may fall outside the fluid boundary. The up-sampling outside the fluid boundary must be detected and prevented; otherwise, the fluid may unrealistically expand. The detection is based on the convex hull of the neighboring particles to the new particle. If the new particle is outside the convex hull, then it is outside the fluid boundary and is not inserted. An

example is illustrated in Figure 7 (a), where the hollow dot *D* denotes the new particle, the black dots the neighboring particles, the big dashed circle the local neighborhood, the dashed small circle the circum-circle of 3 existing particles *A*, *B*, and *C* that make a Delaunay triangle, and the closed series of line segments the convex hull. In this example, particle *D* is outside the convex hull and thus its insertion for up-sampling is cancelled. If particle *D* is on or inside the convex hull as in Figure 7 (b), then it is inserted.

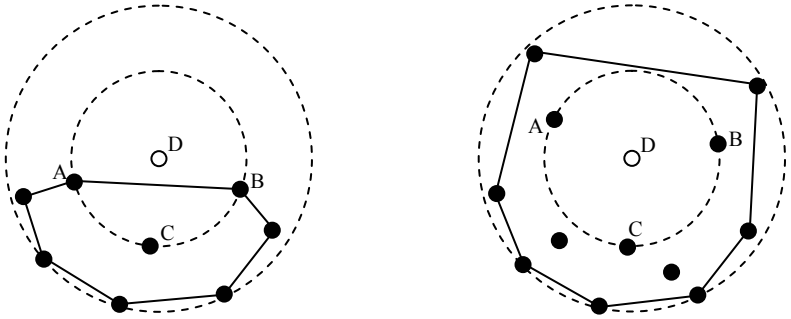


Fig. 7. (a) D (the new particle) outside of the convex hull. It is not inserted. (b) D inside of the convex hull. It is inserted.

5.3 INTERPOLATION AT NEW PARTICLES

When a new particle is inserted, its attributes are interpolated from its neighboring particles. Since SPH is based on the interpolation theory, the attributes can be easily computed at the new particle locations using the classical SPH method, namely, Equation (2). Even though the kernel is already normalized, the interpolated attributes are more accurate if they are each divided by the total weight. Thus, the interpolation equation is:

$$A(r) = \frac{1}{W_{total}} \sum_{j=1}^{n} A_j W_j \tag{18}$$

where A represents the attribute to be interpolated,

$$W_j = \frac{m_j}{\rho_j} W(r - r_j, h) \ , \ W_{total} = \sum_{j=1}^{n} W_j \ , \tag{19}$$

and r the new particle location. In the implementation, the density ρ, velocity v and stress tensor T are substituted into Equation (18).

In the down-sampling method, the mass is conserved since the new particle inherits the total mass of the two deleted particles. In the up-sampling method, the mass of the new particle is computed using Equation (18). In order to conserve mass, the mass contributed by each neighboring particle in Equation (18) is deducted from the corresponding particle, that is,

$$m_j = m_j - m_j \frac{W_j}{W_{total}} \tag{20}$$

where m_j is the mass of the jth neighboring particle.

In the up-sampling method, the mass of the new particle may be very small after the computation of Equation (18). This usually occurs at the place where the fluid splits apart and the mass concentration gets lower than inside the fluid. If the mass is close to zero, the particle system may become unstable. To prevent this instability, the new particle is not inserted if its mass is lower than a threshold value ε_m. In our model, $\varepsilon_m = 0.01\ m_o$, where m_o is the mass of the initialized particles.

6. RESULTS

The proposed model has been implemented and several example animations are produced. Most of the examples are selected to illustrate some interesting motions of non-Newtonian fluids, including fluid-fluid interaction and fluid interaction with rigid-body. Gravity is enabled; unless indicated otherwise explicitly. Some of the key frames in the animations have been presented in the previous sections, and the others are shown at the end of this paper. The accompanied video files of all the animations are also submitted.

In Figure 8, three non-Newtonian fluids with different μ_e and λ are pulled upward. The two of the high elastic fluids are even pulled up off the ground. This behavior is quite different from that of the Newtonian fluid behavior. In Figure 9, two or more non-Newtonian fluid balls, mixable or immiscible, collide with each other. In Figure 10, two immiscible fluids in the shape of long box run into each other. With different μ_e and λ, their behaviors range from fluid to solid. In Figure 11(a) and (b), immiscible fluids run into each other. If the speed is fast enough, one fluid penetrates the other, which causes drastic topological change, as in Figure 11(b); otherwise, they stick together as in Figure 11(a). In Figure 11(c), even though the speed is as fast as the penetration example in Figure 11(b), there is no penetration for mixable fluids. In Figure 12, fluid interaction with rigid-ball is demonstrated. The rigid ball shoots at a fluid tower at different positions and with different sizes. In Figure 12(f) is an immiscible fluid ball shooting at the fluid tower.

Animations in figures	Particle number	Time per frame(s)	Up-sampling time per frame(s)	Sub-steps per frame
Figure 2	4500	8.03	2.35	8
Figure 3	3100	5.65	1.63	8
Figure 4(a)-(c), (e)-(g)	1200	1.46	No up-sampling	8
Figure 4(d), (h)	1200	2.12	0.60	8
Figure 8	1200	2.09	0.60	8
Figure 9(a),(b)	1500	2.63	0.76	8
Figure 9(c)	3000	5.21	1.59	8
Figure 10	2000	3.54	1.08	8
Figure 11	1200	2.11	0.60	8
Figure 12(a)-(e)	1100	1.26	0.55	4
Figure 12(f)	1400	1.57	0.70	4

Table 1: Statistics of the example animations.

All the animations are produced on a 3.0 GHz Pentium 4 PC with 1 GB of memory. By default, different animations in each figure have the same initial conditions except those indicated explicitly. The statistics of the animations are summarized in Table 1. For the animations with the re-sampling method, the actual particle numbers fluctuate around the listed numbers. The initial particle number is chosen such that it is as small as possible but still able to demonstrate interesting fluid motions with smooth fluid surface. Because of the high elasticity constants, we have to divide each frame time step into smaller sub-steps in order to maintain animation stability. The sub-step number is chosen by trials. Irrespective of the sub-step number, the animation time per frame increases linearly in proportion to the particle number. The re-sampling method is called once per frame. About 90% of the re-sampling time is spent on the up-sampling method, in particularly, on the Delaunay construction. The time complexity of the Delaunay construction is $O(N^{4/3})$, where N is the number of particles [Bowyer 1981]. In the example animations, the Delaunay construction time varies approximately linearly with

the particle number. The fluid surface construction time and the rendering time are not included in Table 1. The former is about the same as the time per frame in Table 1, and the latter about 10 seconds per frame.

7. CONCLUSION AND FUTURE WORK

In this paper, a particle-based non-Newtonian fluid model is proposed. After combining with the previous fluid-fluid interaction model [Muller et al. 2005], our model is able to animate the repulsive interactions of immiscible non-Newtonian fluids, which is not presented in the previous fluid models according to our best knowledge. The grid-based Newtonian fluid-fluid interaction model proposed by [Hong and Kim 2003] only accounts for the interfacial tension and cannot handle repulsive fluid interactions. Thus, combining it with the grid-based non-Newtonian model [Goktekin et al. 2004] still can not animate the repulsive interactions of immiscible non-Newtonian fluids.

At present, the example animations are produced offline. Since only a few thousands of particles are used at most, it is promising for interactive animations in the future when computer hardware improves in performance. Two possible efficiency improvements can be made in the near future. In the current implementation, the up-sampling method is called for every frame and the Delaunay triangulation is constructed over all the fluid particles. In our experiments, we observe that the actual up-sampling, i.e. particle insertion, takes place only at some of the frames and at parts of the fluid. Therefore, the animation efficiency can be improved by an adaptive up-sampling method such that the up-sampling including the Delaunay construction is called when and where needed or most likely needed. Another possible efficiency improvement is to reduce the number of the sub-steps for each frame. The current implementation uses explicit Euler integration, and the high elasticity μ_e forces the smaller sub-steps for animation stability. This limitation can be alleviated by an implicit integration scheme such that less or no sub-step is necessary.

Our model can be used as the ground work for future non-Newtonian fluid modeling and animation. One possible avenue to explore is to add a heating model such that the elasticity constant μ_e and the relaxation time λ are not uniform for a non-Newtonian fluid, while they are in the current model. As well, it would be more interesting to animate a non-Newtonian fluid which exhibits fluid and solid behaviors at the same time, but at different parts of its body. It would also be interesting to animate one non-Newtonian fluid to transmit heat to another high elastic non-Newtonian fluid, which behaves more like solid and then is melting during the heat transmission. An application of such a model for real phenomenon would be the animation of glass blowing, which we are currently investigating.

ACKNOWLEDGMENTS

The authors would like to thank financial supports provided by NSERC and the University of Alberta.

REFERENCES

ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. 2003. Computing and Rendering Point Set Surfaces. *IEEE Transactions on Visualization and Computer Graphics, Vol. 9, No. 1*, pp. 3-15, 2003.

BARBER, C.B., DOBKIN, D.P. AND HUHDANPAA, H.T. 1996. The Quickhull Algorithm for Convex Hulls. *ACM Transactions on Mathematical Software, 22(4)*:469-483, 1996.

BOWYER, A. 1981. Computing Dirichlet Tessellations. *The Computer Journal, Heyden & Sons Ltd, Vol 2, Num 24*, pp.162-166, 1981.

CARLSON, M., MUCHA, P. J., VAN HORN, III, R. B., AND TURK, G. 2002. Melting and flowing. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2002*, 167–174.

CARLSON, M., MUCHA, P. J., AND TURK, G. 2004. Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid. In *Proceedings of ACM SIGGRAPH 2004*, Los Angeles, California, August 8-12.

CHANIOTIS, A. K., POULIKAKOS, D. AND KOUMOUTSAKOS, P. 2002. Remeshed Smoothed Particle Hydrodynamics for the Simulation of Viscous and Heat Conducting Flows. *Journal of Computational Physics 182*, 67–90 (2002).

CLAVET, S., BEAUDOIN P., AND POULIN P. 2005. Particle-based Viscoelastic Fluid Simulation. To appear in *ACM SIGGRAPH/Eurographics 2005 Symposium on Computer Animation*.

DESBRUN, M., AND CANI, M.-P. 1996. Smoothed particles: A new paradigm for animating highly deformable bodies. *Computer Animation and Simulation 1996*, 61–76.

DESBRUN, M., AND CANI, M.-P. 1999. Space-Time Adaptive Simulation of Highly Deformable Substances. *Tech. Rep. 3829, INRIA*, 1999.

ELLERO, M., KROEGER, M. AND HESS, S. 2002. Viscoelastic Flows studied by Smoothed Particle Dynamics, *J. Non-Newtonian Fluid. Mech. 105 (2002)*, 35-51.

ENRIGHT, D. P., MARSCHNER, S. R., AND FEDKIW, R. P. 2002. Animation and rendering of complex water surfaces. In *Proceedings of ACM SIGGRAPH 2002*, 736–744.

FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. In *Proceedings of ACM SIGGRAPH 2001*, 23-30.

FOSTER, N., AND METAXAS, D. 1996. Realistic animation of liquids. *Graphics Interface 1996*, 204–212.

GENEVAUX, O., HABIBI, A., AND DISCHLER, J.-M. 2003. Simulating fluid-solid interaction. *Graphics Interface 2003*, 31-38.

GOKTEKIN, T. G., BARGTEIL, A. W., AND O'BRIEN, J. F. 2004. A Method for Animating Viscoelastic Fluids. In *Proceedings of ACM SIGGRAPH 2004*, Los Angeles, California, August 8-12.

HONG, J.-M. AND KIM, C.-H. 2003. Animation of Bubbles in Liquid. *Computer Graphics Forum 22*, 3 (September 2003).

KEUNINGS, R. 2000. A survey of computational rheology, In *Proceedings of 13th international congress on rheology, (2000), vol. 1*, pp.1.7-1.14.

LORENSEN, W. E. AND CLINE, H. E. 1987. Marching cubes: A high resolution 3D surface reconstruction algorithm. In *Computer Graphics (Proceedings of ACM SIGGRAPH 87), Vol. 21, No. 4*, 163-169, 1987.

LOSASSO, F., GIBOU, F., AND FEDKIW, R. 2004. Simulating water and smoke with an octree data structure. In *Proceedings of ACM SIGGRAPH 2004*, 457-462.

MONAGHAN, J. J. 1992. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics, 30*:543–574, 1992.

MÜLLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications. *ACM SIGGRAPH/ Eurographics 2003 Symposium on Computer Animation*, 154–159.

MÜLLER, M., KEISER, R., NEALEN, A., PAULY, M., GROSS, M. AND ALEXA, M. 2004. Point based animation of elastic, plastic and melting objects. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, p141-151, 2004.

MÜLLER, M., SOLENTHALER, B. KEISER, R. AND GROSS, M. 2005. Particle-Based Fluid-Fluid Interaction. To appear in *ACM SIGGRAPH 2005 Symposium on Computer Animation*.

MÜLLER, M., SCHIRM, S., TESCHNER, M., HEIDELBERGER, B., AND GROSS, M. 2004. Interaction of Fluids with Deformable Solids. In *Proceedings of Computer Animation & Social Agents CASA'04*, Geneva, Switzerland, pp. 159-171, July 7-9, 2004.

OWENS, R. G. AND PHILLIPS, T. N. 2002. Computational Rheology. Imperial College Press, 2002.

PREMOZE, S., TASDIZEN, T., BIGLER, J., LEFOHN, A., AND WHITAKER, R. 2003. Particle-based simulation of fluids. *Computer Graphics Forum 22, 3 (September 2003)*, 401–410.

PRESS, W.H. ET AL. 1992. *Numerical Recipes in C: The Art of Scientific Computing*. 2nd ed., Cambridge, New York, Cambridge University Press, 1992.

RENARDY, M. 2000. *Mathematical Analysis of Viscoelastic Flows*. Philadelphia : Society for Industrial and Applied Mathematics, c2000.

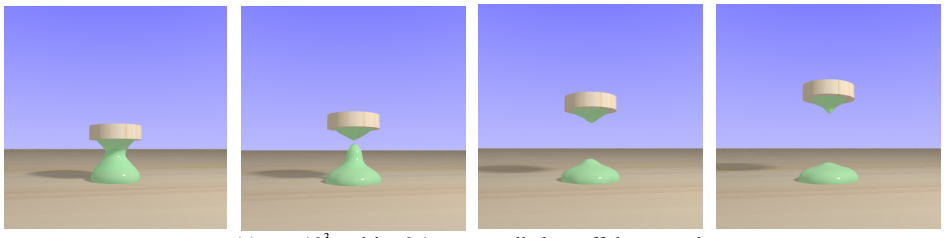
STAM, J. 1999. Stable fluids. In *Proceedings of ACM SIGGRAPH 1999*, 121–128.

STORA, D., AGLIATI, P.-O., CANI, M.-P., NEYRET, F., AND GASCUEL, J.- D. 1999. Animating lava flows. *Graphics Interface 99*, 203–210.

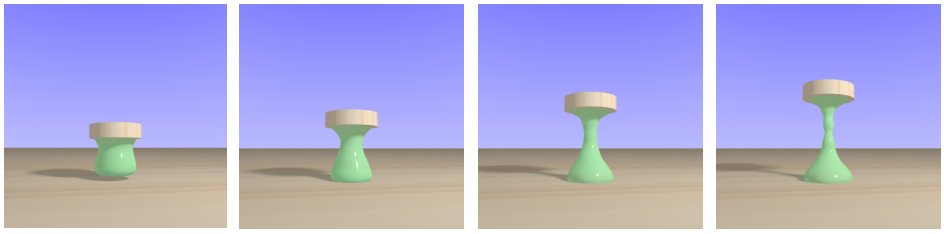
TAKAHASHI, T., FUJII, H., KUNIMATSU, A., HIWADA, K., SAITO, T., TANAKA, K., AND UEKI, H. 2003. Realistic Animation of Fluid with Splash and Foam. *Computer Graphics Forum 22, 3 (September 2003)*.

TERZOPOULOS, D., AND FLEISCHER, K. 1988. Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. In *Proceedings of ACM SIGGRAPH 88*, 269–278.

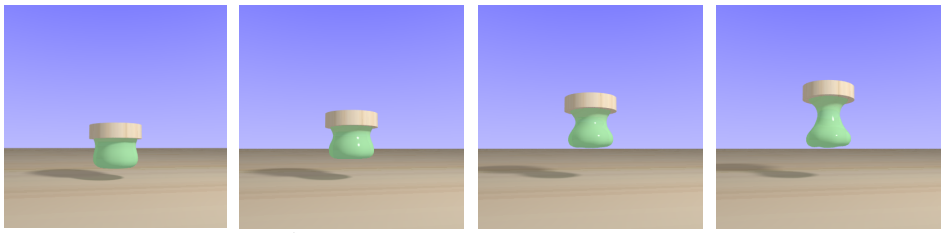
WATSON D.F. 1981. Computing the N-Dimensional Delaunay Tessellation with Apllication to Voronoi Polytopes. *The Computer Journal*, Heyden & Sons Ltd, Vol 2, Num 24, pp.167-172, 1981.



(a) $\mu_e = 10^3$ and $\lambda = 0.1$, never pulled up off the ground.

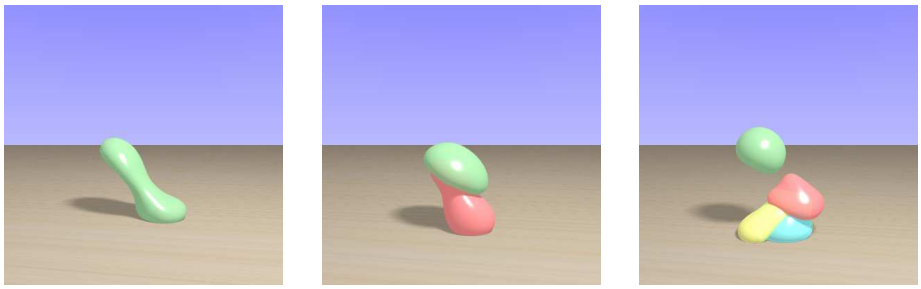


(b) $\mu_e = 10^4$ and $\lambda = 0.1$, pulled up off the ground a little (see the most left frame)



(c) $\mu_e = 10^4$ and $\lambda = 1$, pulled up off the ground higher than (b).

Fig. 8. Vertically stretch fluids of different μ_e and λ. Frame order is from left to right.

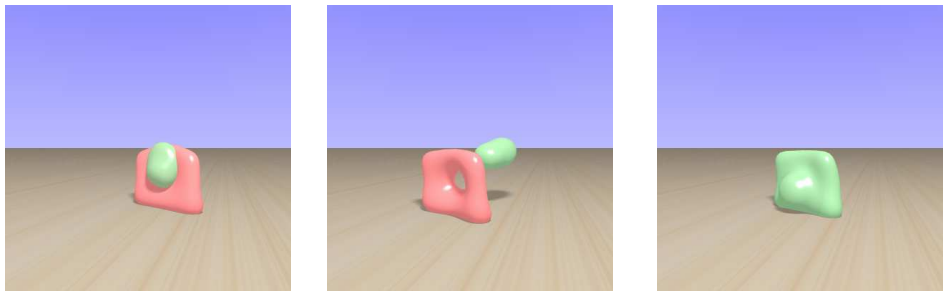


(a) Two mixable fluid balls (b) Two immiscible fluid balls (c) Four immiscible fluid balls

Fig. 9. Multiple balls collide.

(a) $\mu_e = 10^3$ and $\lambda = 0.1$ (b) $\mu_e = 10^4$ and $\lambda = 1$ (c) $\mu_e = 10^5$ and $\lambda = 0.1$

Fig. 10. Long fluids of different μ_e and λ collide.

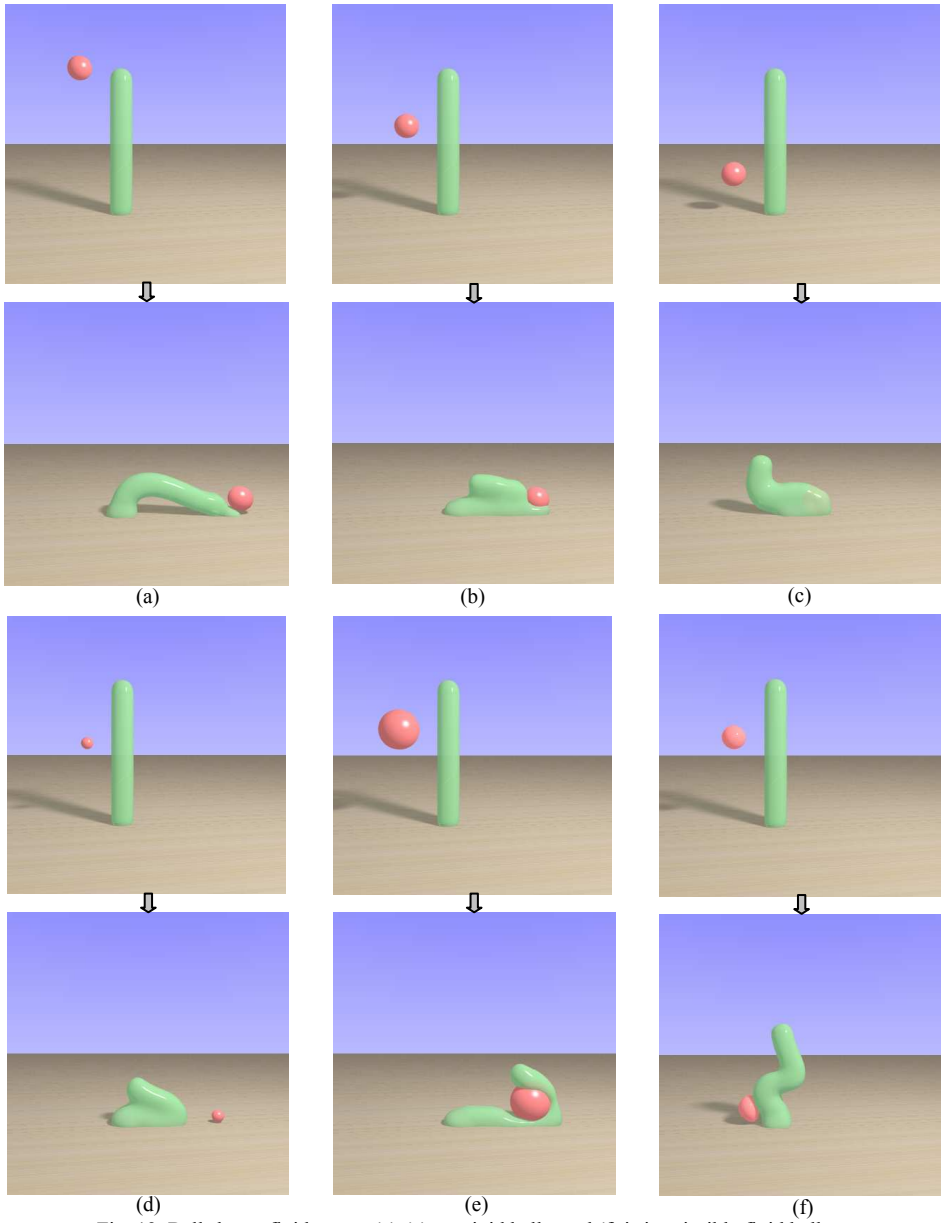


(a) Immiscible fluids. Speed = 75. Too slow to penetrate.

(b) Immiscible fluids. Speed = 150

(c) Mixable fluids. Speed = 150. Mixed without penetration.

Fig. 11. Cylinder fluid runs into moving fluid wall.



(a) (b) (c)

(d) (e) (f)

Fig. 12. Ball shoots fluid tower. (a)-(e) are rigid balls, and (f) is immiscible fluid ball.