



A Method for Animating Viscoelastic Fluids

Tolga G. Goktekin

Adam W. Bargteil

James F. O'Brien

University of California, Berkeley

Abstract

This paper describes a technique for animating the behavior of viscoelastic fluids, such as mucus, liquid soap, pudding, toothpaste, or clay, that exhibit a combination of both fluid and solid characteristics. The technique builds upon prior Eulerian methods for animating incompressible fluids with free surfaces by including additional elastic terms in the basic Navier-Stokes equations. The elastic terms are computed by integrating and advecting strain-rate throughout the fluid. Transition from elastic resistance to viscous flow is controlled by von Mises's yield condition, and subsequent behavior is then governed by a quasi-linear plasticity model.

Keywords: Natural phenomena, physically based animation, computational fluid dynamics, viscoelastic fluids, elastoplastic, viscous fluids, goop, glop.

CR Categories: 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.

1 Introduction

In this paper we describe a technique for animating the behavior of materials, such as the one shown in Figure 1, that exhibit a combination of both fluid and solid characteristics. Often referred to as *viscoelastic fluids*, these materials initially respond to strain elastically like a solid, but when subjected to increasingly large stresses they flow like a fluid. A tremendous variety of materials exhibit this type of behavior, and a few common examples include: mucus, egg white, dough, gelatin, unset cement, liquid acrylic, toothpaste, gels, clay, and liquid soap. Like a solid, these materials can bounce and jiggle, but they will also flow like a fluid. For some of these materials, such as egg white or clay, the combination of elastic and fluid behavior is quite apparent. For others, such as liquid soap, the elastic behavior manifests less obviously as predominately fluid behavior that differs subtly from that of a simply viscous fluid.

The technique we present builds on prior Eulerian methods for animating incompressible fluids with free surfaces. As evidenced by their widespread use, these methods can efficiently produce results that are realistic enough for applications in the demanding visual effects industry.

Our method computes viscoelastic fluid behavior by supplementing the basic Navier-Stokes equations with additional terms for elastic body forces. These terms can be

E-mail: {goktekin,adamb,job}@eecs.berkeley.edu.

From the ACM SIGGRAPH 2004 conference proceedings.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM SIGGRAPH 2004, Los Angeles, CA

© Copyright ACM 2004

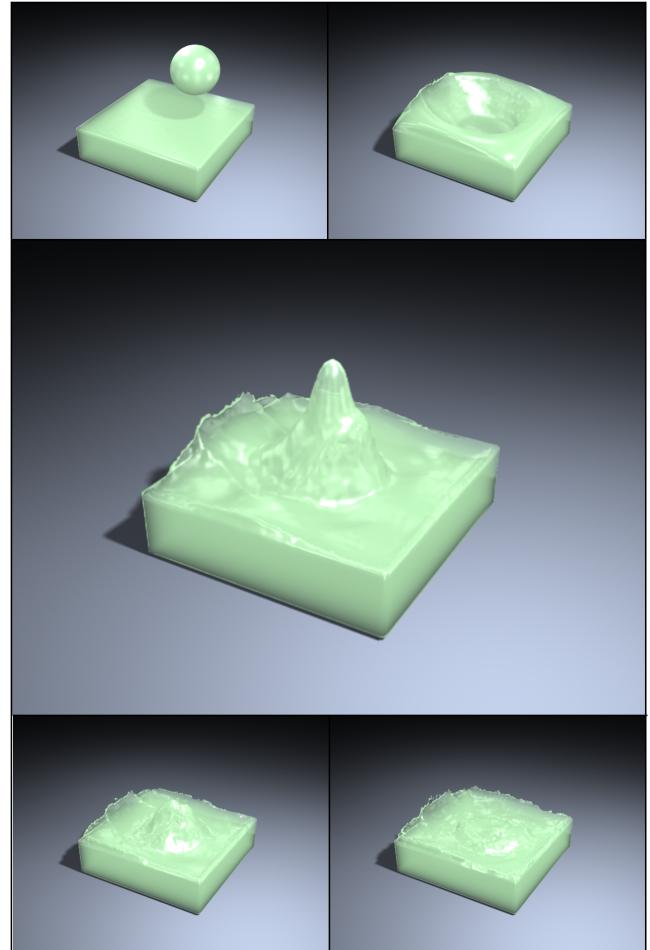


Figure 1: A sequence of images showing a splash in a viscoelastic fluid.

readily computed on rectilinear grids using a staggered discretization scheme, and the use of an Eulerian formulation easily accommodates modeling large flows. These elastic terms require computing the material strain throughout the fluid. Because the fluid simulations do not make use of an explicit reference configuration, strain is computed by integrating strain-rate and advecting the results. The transition from elastic resistance to viscous flow is controlled by von Mises's yield condition, and subsequent behavior is then governed by a quasi-linear plasticity model.

2 Fluids, Solids, and the In-between

While the mechanics describing the behaviors exhibited by solids and fluids may seem distinct, they are actually quite similar. First, both resist changes to their volume. The physical reasons for why they conserve volume may differ,

but the mathematical expressions capturing the behaviors are essentially the same. Furthermore, many fluid methods assume incompressibility and most solid methods assume that volume changes will be negligible. Second, the internal damping force for a solid and the viscous force for a fluid are not just similar, they are identical. One can easily start with the expression for strain-rate-proportional damping and derive the viscous terms of the Navier-Stokes equations. The key difference between an ideal solid and an ideal fluid is the presence or absence of an elastic term that attempts to restore the material to its original shape.

Many materials cannot be classified clearly as a solid or fluid. These materials elastically resist deformation up to a certain point after which they begin to flow. In fact, even materials that are clearly solids behave this way: they act elastically over some regime and then begin to flow plastically (or perhaps fracture). Similarly, some materials that are clearly fluids can withstand very small strains elastically. The continuous variable that spans the space between solid and fluid materials is this limit on how much stress can be tolerated before flow occurs. Other properties such as damping/viscosity, density, and elastic stiffness are largely orthogonal. When the elastic limit is set to a high value, the material behaves like a solid, when it is zero the material behaves like a fluid, and intermediate values correspond to materials like mucus, liquid soap, toothpaste, or clay. These intermediate materials are often referred to as viscoelastic fluids or as elastoplastic solids, depending on whether their behavior is closer to that of an ideal fluid or ideal solid.

3 Background

In the field of computer graphics, the technique described in [Carlson et al., 2002] is perhaps closest in intent to the method we describe here. Like us, they were interested in modeling materials with properties intermediate between solids and fluids using an Eulerian grid-based fluid simulation method. However, they opt to map the continuum between fluids and solids to varying viscosity. In their system a solid is simply a fluid with very high viscosity. This approach ignores the elastic behavior demonstrated by many materials. Nevertheless, they do generate nice results for highly viscous fluids, and they describe an implicit integration method for coping with stability issues arising from very high viscosities.

Other graphics researchers have used particle-based methods for modeling highly viscous fluids and for modeling fluids with some form of elasticity. In [Terzopoulos et al., 1989] the authors modeled melting thermoelastic materials. The particles exerted cohesive, viscous, and volume-preserving forces on their neighbors. While solid, each particle was connected to a fixed set of neighbors using elastic springs. As the material would become more fluid-like, the springs would weaken, and eventually disappear. By varying the elastic properties of the materials, this method could model a range of behaviors, but without plasticity, it could not model materials, like clay, that flow into a new configuration and then resist changes from that configuration. Similar approaches using different particle formulations have appeared in [Desbrun and Gascuel, 1995], [Desbrun and Cani, 1996], [Cani and Desbrun, 1997], and [Stora et al., 1999]. The method appearing in [Desbrun and Gascuel, 1995] used elastic forces with dynamically determined neighbors to allow behavior that is similar to plastic flow.

Perhaps the greatest limitation on the level of realism achievable by these particle methods was the relatively small number of particles used. However, as processor speeds have increased, particle-based methods have been able to achieve increasingly impressive results. Compelling real-time results

for modestly sized systems appear in [Müller et al., 2003], and [Premož et al., 2003] demonstrates off-line results that are comparable to the current best grid-based methods. Although both of these recent methods focus on strictly liquid behavior, they could be extended along lines similar to what we propose here.

Some methods for modeling solids have dealt with limited amounts of plastic flow. Both [Terzopoulos and Fleischer, 1988a] and [Terzopoulos and Fleischer, 1988b] describe transition to plastic flow based on von Mises's yield condition, and [O'Brien et al., 2002] used a similar plasticity model for ductile fracture behavior. We use the same yield condition of von Mises, but we do not assume that plastic flow occurs instantaneously. Instead we use a more complex model that accommodates phenomena such as creep. Additionally, these prior methods used Lagrangian meshes with largely fixed topology, and so they would have encountered "tangling" difficulties, such as inverting elements, for large amounts of plastic flow.

Another, rather interesting, approach to combining solid and fluid behaviors appears in [Nixon and Lobb, 2002]. They surround a fluid simulation with an elastic membrane. The result is an object that behaves somewhat like a water balloon.

Our work builds directly on previous grid-based, Eulerian methods for animating fluids with free surfaces. Details on these methods can be found in [Foster and Metaxas, 1996], [Stam, 1999], [Foster and Fedkiw, 2001], and [Enright et al., 2002]. In particular, our work essentially extends [Enright et al., 2002] to include the behavior of viscoelastic fluids.

Outside the graphics field, viscoelastic materials have been studied extensively. We refer the reader to the texts [Fung, 1965], [Han and Reddy, 1999], and [Bird et al., 1987] for detailed descriptions of several different models for viscoelastic and elastoplastic materials.

The general approaches we use for introducing elastic forces into the Navier-Stokes equations, and integrating and advecting strain are not completely novel. Some recent examples of fluid simulation outside the graphics literature that involve elastic forces include [Gerritsma, 1996], [Tomé et al., 2002], and [Bonito et al., 2003]. A detailed analysis of two-dimensional simulations of viscoelastic fluids on staggered rectilinear grids appears in [Gerritsma, 1996]. The three-dimensional method we use for storing rank-two tensor quantities on a staggered grid is a generalization of their two-dimensional method. In [Bonito et al., 2003] a combination of rectilinear grids and finite elements are used with a volume-of-fluid method to model three-dimensional fluids with elastic properties. They do not include plasticity and they store all quantities at cell centers. The marker-and-cell based method in [Tomé et al., 2002] is another example solving viscoelastic free-surface flows. They address issues relating to elastic-stress based boundary conditions at rigid-body and free-surfaces. Although they use a staggered grid for the velocity field, they still store their tensor values at cell centers.

4 Simulation Methods

The framework we use for fluid simulation is based on the method described in [Enright et al., 2002]. This framework consists of two primary components which work together to produce useful results. The first is a rectilinear grid that stores the values that define the fluid's state. The values on the grid change as forces act on the fluid, and they also change as the fluid moves through the space delineated by the grid. The second component is a function whose level-set at zero locates the boundaries of the fluid. The function is represented using a combination of particles and values

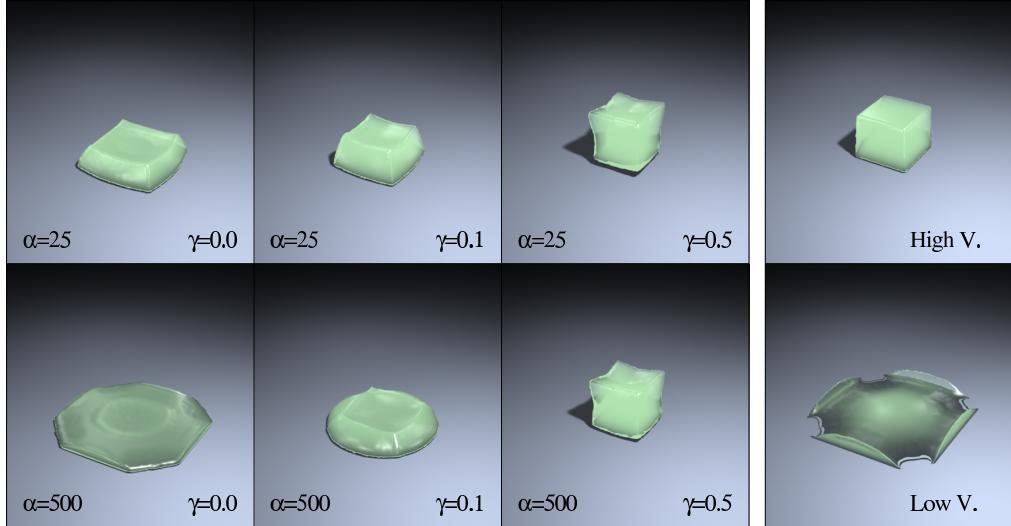


Figure 2: This figure demonstrates some effects achieved by varying α and γ . Each image shows a cube of material that has been dropped onto a hard surface inside an invisible box. The rightmost images show a low viscosity simple fluid and an extremely viscous simple fluid for comparison. The viscoelastic examples have the same viscosity as the low viscosity fluid. Observe that for appropriate values of α and γ the viscoelastic fluid actually bounces.

defined on a second rectilinear grid. The particles and grid values evolve based on the motion of the fluid. A full description of this type of system is beyond the scope of this paper, so we focus on the changes we make to accommodate viscoelastic behavior. We suggest the following references for a more complete description of this simulation methodology: [Foster and Metaxas, 1996], [Stam, 1999], [Foster and Fedkiw, 2001], and [Enright et al., 2002].

4.1 Governing Equations

Behavior of the viscoelastic fluid is governed by a modified version of the Navier-Stokes equations that includes an additional term for elastic stress. At any point in the fluid the fluid's acceleration is determined by

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}^\top \nabla) \mathbf{u} - \frac{\nabla p}{\rho} + \frac{\mu_v}{\rho} (\nabla^\top \nabla) \mathbf{u} + \frac{\mu_e}{\rho} \nabla^\top \boldsymbol{\epsilon} + \frac{\mathbf{f}}{\rho} \quad (1)$$

where \mathbf{u} is the fluid's velocity at a point in space, p its pressure, ρ its density, μ_v its coefficient of viscosity, μ_e its elastic modulus, $\boldsymbol{\epsilon}$ the elastic strain tensor at that point in space, and \mathbf{f} includes any other forces such as gravity or vorticity confinement. The symbol ∇ denotes the vector of differential operators $\nabla = [\partial/\partial x, \partial/\partial y, \partial/\partial z]^\top$, and we have implicitly assumed that μ_v and μ_e are constant throughout the material. By omitting elastic and viscous terms relating to dilation, we have also assumed that the fluid is incompressible.

Because the fluid is incompressible, the velocity field within the fluid volume is subject to the constraint that

$$\nabla^\top \mathbf{u} = 0 \quad (2)$$

This condition is enforced by adjusting the pressure field at each integration step. Additionally, we do not use the first term of Equation (1) (the advection term) directly. Instead, we use a semi-Lagrangian method to advect field values. We refer the reader to [Stam, 1999] and [Foster and Fedkiw, 2001] for a discussion on efficiently modeling the above equations.

4.2 Elastic, Plastic, and Total Strain

The fourth term of Equation (1) computes acceleration due to elastic forces and it requires knowing the elastic strain throughout the fluid. If we had an explicit deformation function then we could use its spatial derivatives to compute strain. However, with the Eulerian formulation we are using there is no deformation function available. Furthermore, the large deformation and flow experienced by the material makes tracking deformation impractical. Instead we compute strain by integrating strain-rate. Because we do not wish to model a perfectly elastic material, we also require rules concerning how the elastic strain changes due to plastic yielding.

We first separate the total strain into an elastic and a plastic component so that

$$\boldsymbol{\epsilon}^{\text{Tot}} = \boldsymbol{\epsilon}^{\text{Elc}} + \boldsymbol{\epsilon}^{\text{Plc}} \quad . \quad (3)$$

Where $\boldsymbol{\epsilon}^{\text{Tot}}$, $\boldsymbol{\epsilon}^{\text{Elc}}$, and $\boldsymbol{\epsilon}^{\text{Plc}}$, are respectively the total, elastic, and plastic strains. (Outside this section we denote elastic strain as simply $\boldsymbol{\epsilon}$.) The strain rate is the time derivative of the total strain, so the total strain at some time t is

$$\boldsymbol{\epsilon}^{\text{Tot}} = \boldsymbol{\epsilon}_0^{\text{Tot}} + \int_0^t \dot{\boldsymbol{\epsilon}}^{\text{Tot}} dt \quad (4)$$

where $\boldsymbol{\epsilon}_0^{\text{Tot}}$ is the total strain at time $t = 0$ and $\dot{\boldsymbol{\epsilon}}^{\text{Tot}}$ is the strain rate given by

$$\dot{\boldsymbol{\epsilon}}^{\text{Tot}} = \left(\nabla \mathbf{u}^\top + (\nabla \mathbf{u}^\top)^\top \right) / 2 \quad . \quad (5)$$

Similarly, the plastic strain is given by integrating plastic flow.

$$\boldsymbol{\epsilon}^{\text{Plc}} = \boldsymbol{\epsilon}_0^{\text{Plc}} + \int_0^t \dot{\boldsymbol{\epsilon}}^{\text{Plc}} dt \quad (6)$$

We use von Mises's criterion for determining when plastic flow should occur. This criterion depends on the elastic

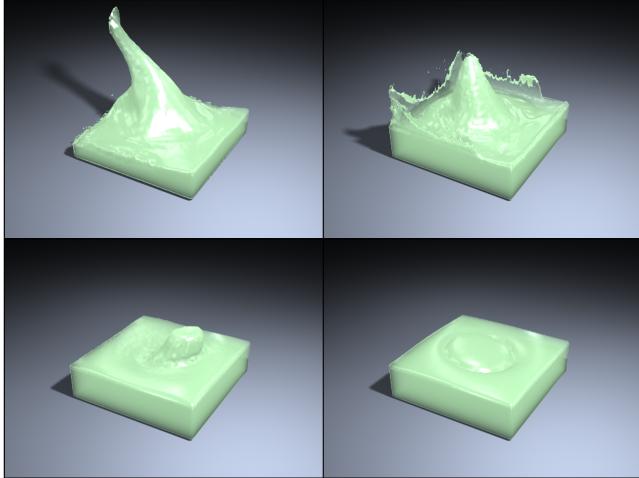


Figure 3: These images, along with the ones shown in Figure 1, show how splash behavior is affected by elasticity. The upper-left image shows a fluid example. As expected, the Worthington column moves in the direction of the impact. In the viscoelastic examples, the column rises vertically (upper-right) or actually rises back toward the impact direction (lower-left). The lower-right image shows a fluid with both elasticity and high viscosity.

strain deviation, ϵ' , which is the elastic strain with any dilation removed¹

$$\epsilon' = \epsilon^{\text{Elc}} - \frac{\text{Tr}(\epsilon^{\text{Elc}})}{3} I . \quad (7)$$

So long as the magnitude (Frobenius norm) of the strain deviation remains below the yield point, γ , no plastic flow occurs. When the limit is exceeded, flow occurs at a rate proportional to the amount the limit has been exceeded by. So the flow rate for plastic strain is

$$\dot{\epsilon}^{\text{Plc}} = \alpha \frac{\epsilon'}{\|\epsilon'\|} \max(0, \|\epsilon'\| - \gamma) , \quad (8)$$

where α is the material's elastic decay rate, which determines the rate of plastic flow.

Assuming that the initial total and plastic strains are both zero, we can combine the above to compute the time derivative of the elastic strain which takes into account changes to both total and plastic strains

$$\dot{\epsilon}^{\text{Elc}} = (\nabla u^\top + (\nabla u^\top)^\top)/2 - \alpha \frac{\epsilon'}{\|\epsilon'\|} \max(0, \|\epsilon'\| - \gamma) \quad (9)$$

The images shown in Figure 2 illustrate some effects generated by varying α and γ .

Note that Equation (9) does not take into account the movement of the material through space. Like velocity or any other fluid property, the elastic strain must be advected with the fluid. We use the same semi-Lagrangian advection scheme that we use for the fluid velocities, and we update the elastic strain using Equation (9) after our advection step.

4.3 Grid Issues

The well known staggered grid method, originally described by [Harlow and Welch, 1965], elegantly avoids problems that

¹This distinction is a bit pedantic here, because the fluid is incompressible and so the dilation should always be zero.

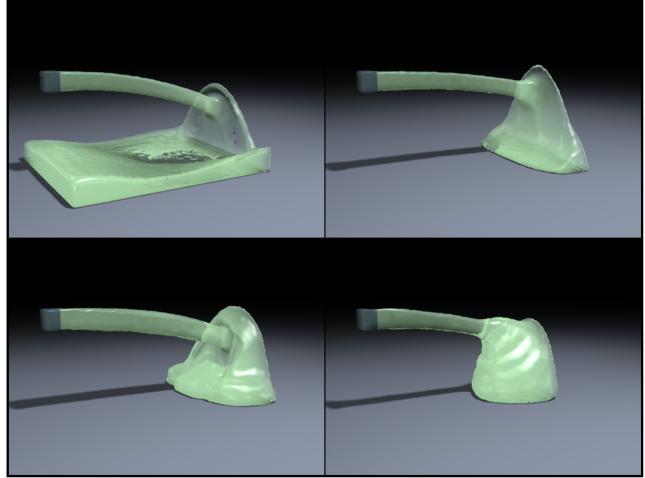


Figure 4: Examples of fluid being sprayed into a container. The way different fluids flow or pile in the container varies significantly.

plague methods that store collocated pressure and velocity values. However in addition to scalars (*e.g.* pressure) and rank-one tensors (*e.g.* velocity), we also need to store the elastic strain, a rank-two tensor, on the simulation grid.

Just as velocity components are stored separately on faces, the different components of the strain tensor are stored at different locations. The diagonal entries are stored at the cell centers. The off-diagonal entries are stored at the center of edges perpendicular to the component directions. For example, the xy components are stored on the edges parallel to the z axis. This approach is a generalization of the 2D method described in [Gerritsma, 1996], and they describe its merits in detail.

We use a particle-level-set method for tracking the fluid's free surface as described in [Enright et al., 2002], but with the substantially faster, though less accurate, method detailed in [Enright et al., 2004]. The authors note that the method is susceptible to volume loss, and we found this behavior to be problematic for some of our examples that involve fixed, small amounts of fluid. We were able to ameliorate this problem somewhat by using a level-set grid with twice the fluid grid's resolution, and that is staggered with respect to the fluid grid. This scheme places level-set grid centers on the cell centers, face centers, edge centers, and nodes of the fluid grid. In addition to helping to prevent volume loss by locating level-set values where velocity boundary constraints are enforced, the higher resolution also benefits the rendered surfaces.

5 Results and Discussion

We have implemented this method for modeling viscoelastic behavior and used it to generate several example animations. Most of these examples were selected to illustrate some interesting aspect of viscoelastic fluid behavior. All of the examples shown in this paper also appear on the accompanying video, which also contains additional examples.

Figures 1 and 3 show several splashes that are generated when a fluid sphere is hurled into a tank containing the same material. The motion of the pure fluid example differs substantially from the viscoelastic examples. Additionally, the surfaces of the viscoelastic examples retain evidence of the impact even after motion has stopped. Figure 4 shows jets of different fluids sprayed into a closed container. Again, the behavior of simple and viscoelastic fluids differ substantially.

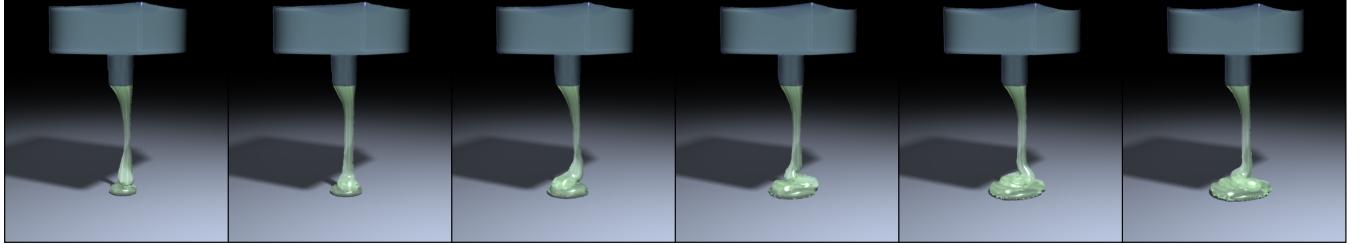


Figure 5: A sequence of images showing a viscoelastic fluid draining from a tank. The stream spirals in a fashion characteristic of viscoelastic fluids.

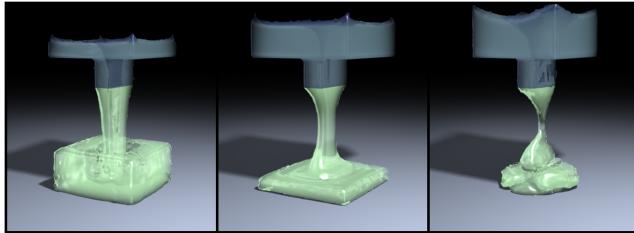


Figure 6: These additional examples of viscoelastic fluids draining from a tank show a range of different behaviors.

Figures 5 and 6 illustrate some of the variation in viscoelastic fluid behaviors. In Figure 5, fluid draining from a tank forms a thin stream that forms a spiral pattern as it piles. The examples in Figure 6 billow up around the downward stream, create a folding pattern, and break apart into gobs. A simply viscous fluid would merely flow out to fill the container.

The drip examples in Figures 7 and 8 show the behavior of a gob of material that has been stuck to the underside of a horizontal surface. The effect of shape memory created by the elastic forces can be seen in these images. Some of the images in Figure 2 show similar behavior that occurs when cubes of different materials are dropped onto a hard surface. The examples with high yield strain, *i.e.* large γ , behave like deformable solids and bounce.

As shown in Figure 9, we can also model highly deformable, sticky objects that interact with each other. When the spheres collide, their level-set surfaces merge so that they adhere. The fluid retains its momentum, generating the resulting spinning and stretching motion. Close examination shows that the spheres slightly anticipate their collision. This error occurs because the surfaces begin to interact through shared ghost cells.

The images in Figure 10 are still images from an animation we produced using this simulation method. These examples demonstrate viscoelastic fluids interacting with interesting geometry.

All of the images were rendered with a standard Newton-iteration based ray marching algorithm implemented in the open source renderer Pixie developed by Okan Arikan. While ray marching produces nice results we think they might be improved using the method described in [Heckbert, 1987].

Some of our examples suffer from noticeable volume loss. This occurs because, while the particle level-set method does a nice job modeling moderately thick volumes of fluids, very thin surfaces, or strands, still have a tendency to vanish. These effects are particularly noticeable visually when the fluid is moving in orderly fashion, as opposed to splashing about chaotically. It is difficult to say if this behavior is a deficiency in our implementation or a limitation of the surface tracking method.

The speed of this simulation method is approximately the same with and without the addition of elastic forces. For example, one of the falling cube examples on a 40^3 grid requires about half an hour of computation per second of animation on a single 3 GHz Pentium 4 processor. We are using an explicit integration method for the viscous and elastic forces, so very high viscous or elastic coefficients would probably cause stability problems and force smaller time steps. If it became an issue, that difficulty could be ameliorated somewhat with an implicit integration scheme.

The material can be made to adhere to or slip off of boundary surfaces by adjusting the velocity or pressure constraints enforced along closed boundaries. However, in our current implementation all fluids will stick to each other because different surface components merge when they collide. For the fluids we show in our examples, this behavior is a desirable feature. However, for non-sticky materials, like cold gelatin, it would be undesirable.

To a large extent, our method for incorporating elastoplastic terms does not depend on the underlying fluid simulation method, and one could easily adapt the method to other fluid simulation techniques such as smoothed-particle hydrodynamics. Furthermore, we found that once we already had a working fluid simulation, adding the elastoplastic terms was fairly easy.

Finally, we note that while the method we present can model a wide range of phenomena, many real materials can demonstrate behaviors not captured by this model. Biological fluids, such as blood, can exhibit many interesting effects that arise from their microscopic structure. Even relatively simple polymer suspensions can demonstrate behavior that can only be roughly captured with this model. Quoting from [Bird et al., 1987]:

A fluid that's macromolecular
Is really quite weird — in particular
The abnormal stresses
The fluid possesses
Give rise to effects quite spectacular.

Acknowledgments

We thank the other members of the Berkeley Graphics Group for their helpful criticism and comments. We especially thank Okan Arikan for his help with rendering our images. This work was supported in part by NSF CCR-02-04377, California MICRO 02-055, and by generous support from Pixar Animation Studios, Intel Corporation, Sony Computer Entertainment America, the Okawa Foundation, and the Alfred P. Sloan Foundation.

References

- BIRD, R. B., ET AL. 1987. *Dynamics of Polymeric Liquids*. John Wiley & Sons, New York.
- BONITO, A., PICASSO, M., AND LASO, M. 2003. Numerical simulation of 3D viscoelastic flows with free surfaces. Submitted to Journal Non-Newtonian Fluid Mechanics. Preprint.



Figure 7: A sequence of images showing a viscoelastic fluid dripping off of a surface to which it has adhered.

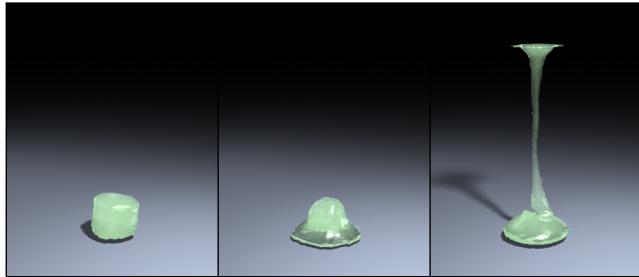


Figure 8: Other drip examples. The first is highly elastic and falls off as a single blob. The second does as well, but splats because it is less rigid. The final example demonstrates a characteristic swirling behavior.

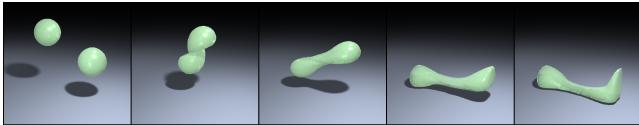


Figure 9: Two sticky spheres that have been thrown into each other so that they adhere together.

- CANI, M.-P., AND DESBRUN, M. 1997. Animation of deformable models using implicit surfaces. *IEEE Transactions on Visualization and Computer Graphics* 3, 1 (Jan.), 39–50.
- CARLSON, M., MUCHA, P. J., III, R. B. V. H., AND TURK, G. 2002. Melting and flowing. In the *ACM SIGGRAPH 2002 Symposium on Computer Animation*, 167–174.
- DESBRUN, M., AND CANI, M.-P. 1996. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Computer Animation and Simulation 1996*, 61–76.
- DESBRUN, M., AND GASCUEL, M.-P. 1995. Animating soft substances with implicit surfaces. In the *Proceedings of SIGGRAPH 95*, 287–290.
- ENRIGHT, D. P., MARSCHNER, S. R., AND FEDKIW, R. P. 2002. Animation and rendering of complex water surfaces. In the *Proceedings of ACM SIGGRAPH 2002*, 736–744.
- ENRIGHT, D., LOSASSO, F., AND FEDKIW, R. 2004. A fast and accurate semi-Lagrangian particle level set method. *Computers and Structures*.
- FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. In the *Proceedings of ACM SIGGRAPH 2001*, 23–30.
- FOSTER, N., AND METAXAS, D. 1996. Realistic animation of liquids. In *Graphics Interface 1996*, 204–212.
- FUNG, Y. C. 1965. *Foundations of Solid Mechanics*. Prentice-Hall, Englewood Cliffs, N.J.
- GERRITSMA, M. I. 1996. *Time dependent numerical simulations of a viscoelastic fluid on a staggered grid*. PhD thesis, Rijksuniversiteit Groningen.
- HAN, W., AND REDDY, B. D. 1999. *Plasticity: Mathematical Theory and Numerical Analysis*. Interdisciplinary Applied Mathematics. Springer-Verlag, New York.

- HARLOW, F., AND WELCH, J. 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. *The Physics of Fluids* 8, 2182–2189.
- HECKBERT, P. S. 1987. Ray tracing Jell-O brand gelatin. In the *Proceedings of ACM SIGGRAPH 87*, 73–74.
- MÜLLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications. In the *ACM SIGGRAPH 2003 Symposium on Computer Animation*, 154–159.
- NIXON, D., AND LOBB, R. 2002. A fluid-based soft-object model. *IEEE Computer Graphics & Applications* 22, 4 (July), 68–75.
- O'BRIEN, J. F., BARGTEIL, A. W., AND HODGINS, J. K. 2002. Graphical modeling and animation of ductile fracture. In the *Proceedings of ACM SIGGRAPH 2002*, 291–294.
- PREMÖZE, S., TASDIZEN, T., BIGLER, J., LEFOHN, A., AND WHITAKER, R. 2003. Particle-based simulation of fluids. *Computer Graphics Forum* 22, 3 (Sept.), 401–410.
- STAM, J. 1999. Stable fluids. In the *Proceedings of ACM SIGGRAPH 99*, 121–128.
- STORA, D., AGLIATI, P.-O., CANI, M.-P., NEYRET, F., AND GASCUEL, J.-D. 1999. Animating lava flows. In *Graphics Interface 99*, 203–210.
- TERZOPoulos, D., AND FLEISCHER, K. 1988. Deformable models. *The Visual Computer* 4, 306–331.
- TERZOPoulos, D., AND FLEISCHER, K. 1988. Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. In the *Proceedings of ACM SIGGRAPH 88*, 269–278.
- TERZOPoulos, D., PLATT, J., AND FLEISCHER, K. 1989. Heating and melting deformable models (from goop to glop). In *Graphics Interface 1989*, 219–226.
- TOMÉ, M., MANGIAVACCHI, N., CUMINATO, J., CASTELO, A., AND MCKEE, S. 2002. A finite difference technique for simulating unsteady viscoelastic free surface flows. *Journal of Non-Newtonian Fluid Mechanics* 106, 61–106.

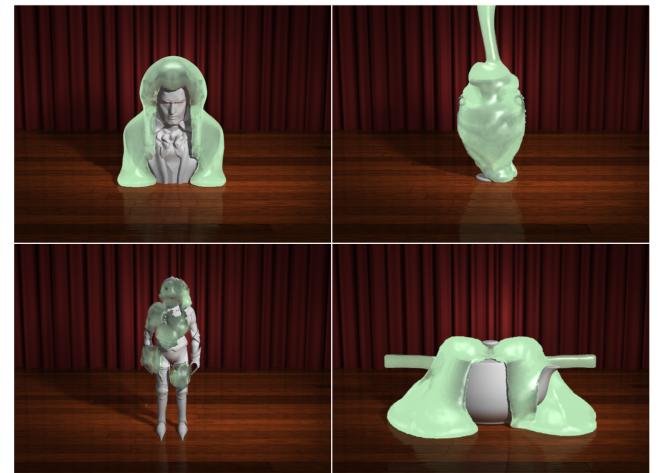


Figure 10: Images from the animation *Gratuitous Goop*, which was produced using the methods described in this paper and appeared in the SIGGRAPH 2004 Electronic Theater.