

A Simple Finite Volume Method for Adaptive Viscous Liquids

Christopher Batty^{1,2} and Ben Houston²

¹Columbia University, New York, USA

²Exocortex Technologies Inc., Ottawa, Canada

Abstract

We present the first spatially adaptive Eulerian fluid animation method to support challenging viscous liquid effects such as folding, coiling, and variable viscosity. We propose a tetrahedral node-based embedded finite volume method for fluid viscosity, adapted from popular techniques for Lagrangian deformable objects. Applied in an Eulerian fashion with implicit integration, this scheme stably and efficiently supports high viscosity fluids while yielding symmetric positive definite linear systems. To integrate this scheme into standard tetrahedral mesh-based fluid simulators, which store normal velocities on faces rather than velocity vectors at nodes, we offer two methods to reconcile these representations. The first incorporates a mapping between different degrees of freedom into the viscosity solve itself. The second uses a FLIP-like approach to transfer velocity data between nodes and faces before and after the linear solve. The former offers tighter coupling by enabling the linear solver to act directly on the face velocities of the staggered mesh, while the latter provides a sparser linear system and a simpler implementation. We demonstrate the effectiveness of our approach with animations of spatially varying viscosity, realistic rotational motion, and viscous liquid buckling and coiling.

1. Introduction

Viscosity in fluids acts to damp out relative or shearing motions, resulting in highly damped motion and the characteristic coiling and folding of "thick" liquids such as honey and syrup. Despite the ubiquity of such liquids, challenges remain in our ability to animate them accurately and efficiently. For example, viscous forces are commonly presented in terms of Laplacian smoothing operations applied independently to each component of the velocity field, a simplification that fails near liquid surfaces [BB08] or when viscosity varies spatially [REN*04]. In practice, this breaks even rigid rotation, and prevents more complex coiling and buckling effects. These shortcomings were addressed only recently for the uniform grid case [BB08]. However, this leaves unresolved the important case of spatially *adaptive* liquid animation methods which provide enhanced scalability and flexibility using mesh elements of differing sizes and configurations. Every adaptive Eulerian method for fluid animation presented to date, whether based on octrees or tetrahedra, has either neglected viscosity entirely [LGF04, CFL*07, SBH09, BXH10, BBB10] or applied the simplified Laplacian model [HK05, ETK*07, MCP*09].

The goal of this paper is to introduce an adaptive Eulerian liquid simulation technique based on tetrahedral meshes that

supports the full set of purely Newtonian viscous behaviours, including rotation, buckling, coiling, and variable viscosity. This enables the creation of scenes featuring both large bodies of viscous liquid as well as important smaller details. To achieve this, we make two contributions:

- **Nodal Finite Volume Method for Eulerian Viscosity:** We develop a simple, node-based Eulerian discretization of variable-coefficient viscosity on tetrahedral meshes by adapting concepts from the animation of deformable solids.
- **Mapping between Node- and Face-based Velocities:** To incorporate this nodal scheme into standard staggered velocity fluid simulators, we present two approaches to map between these distinct sets of degrees of freedom.

2. Related Work

Methods to animate viscous materials can be grouped into Eulerian and Lagrangian approaches. The earliest Eulerian treatment of viscosity in graphics was by Foster and Metaxas [FM96], who used an explicit finite difference approach based on the classic marker-and-cell scheme [HW65]. However, the stability restriction $\Delta t < O(\rho\Delta x^2/\mu)$ for explicit integration of viscosity necessitates extremely small time steps, making this method inefficient for even moderate vis-

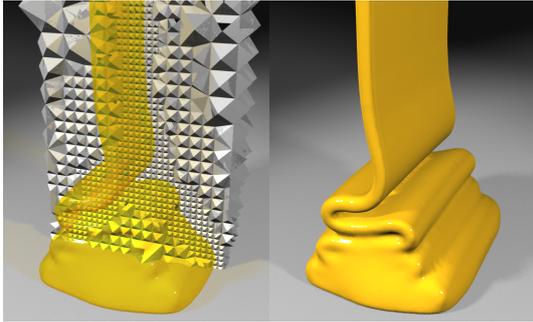


Figure 1: A buckling viscous sheet, alongside a cutaway visualization of the adaptive tetrahedral simulation mesh.

cosities. Stam addressed this by adopting an unconditionally stable fully implicit scheme for viscosity which (along with semi-Lagrangian advection) proved crucial in making the method practical for graphics applications [Sta99]. Carlson et al. extended Stam’s implicit viscosity method to free surface liquids [CMVT02], and Falt et al. improved the boundary conditions to support rigid translation [FR03]. Rasmussen et al. noted that the preceding approaches applied a simplified Laplacian form of viscosity which is inaccurate when viscosity varies spatially [REN*04]. They presented a semi-implicit discretization of the variable-coefficient viscosity equations that nonetheless avoids coupling the different components of velocity. Batty et al. showed that the fully coupled equations can instead be discretized with an unconditionally stable fully implicit scheme that yields a symmetric positive-definite linear system, and demonstrated that proper treatment of the free surface is vital to handling rotational motions such as buckling [BB08]. It is worth noting that elastic forces can also give rise to buckling in solids and non-Newtonian fluids (eg. [GBO04]), however we will focus on the purely viscous effects of common Newtonian fluids.

While the preceding methods use uniform grids, unstructured mesh methods for Eulerian viscous fluids have also been proposed, often using the simplified Laplacian form of viscosity noted above. Wendt et al. [WBOL07] applied a finite volume discretization storing both normal and tangential components of velocity on faces of tetrahedra, in contrast to the more common staggered approach that stores only the normal components. Elcott et al. [ETK*07] treated the vorticity form of the Navier-Stokes equations on a staggered mesh using discrete exterior calculus, again using the Laplacian form of viscosity. Mullen et al. used the same discretization on top of an energy-preserving fluid integrator [MCP*09].

In computational physics there are a vast variety of low and high order finite element and finite volume discretizations of the Stokes problem (i.e., tightly coupled viscosity and pressure) on unstructured meshes; however, to our knowledge Bonito et al. [BPL06] are the only group to ap-

ply them to three-dimensional, high viscosity, free surface flows. They simulated coiling viscoelastic liquids with a node-based stabilized tetrahedral finite element method for the Stokes problem. While their tetrahedral Stokes solver is potentially adaptive, the use of regular grid advection limited them to uniformly sized elements. Our approach is also distinguished by a focus on simplicity and efficiency motivated by the needs of animation. For example, we treat viscosity and pressure separately for efficiency, and use only basic (non-stabilized) linear elements. Our method avoids pressure instabilities and locking by integrating conveniently with the common staggered velocity representation for Eulerian fluids.

Lagrangian treatments of viscoelastic deformable objects are also highly relevant to our work. O’Brien et al. applied the finite element method (FEM) to deforming and fracturing viscoelastic and viscoplastic materials [OH99, OBH02] using explicit time integration. This included a proper model for viscosity, often referred to as damping in the context of solids. Teran et al. presented an equivalent and geometrically intuitive finite volume method (FVM) for viscoelastic solids, and noted that since damping forces are typically linear in velocity they can be treated implicitly to allow larger time steps [TBFH03]. Irving et al. extended this method to handle inverted elements [ITF04] and incompressible materials [ISF07]. Bargteil et al. added remeshing to support large plastic flows [BHWT07], and Wojtan and Turk showed that using an embedded approach can enable faster remeshing by avoiding the need to generate surface-conforming elements [WT08]. As a result, this last method is able to handle highly viscoplastic deformations that approach the behaviour of mildly compressible fluids. Despite its Lagrangian perspective, this is the closest relative of our method.

Recently, Bergou et al. derived a viscous thread model by applying the Rayleigh analogy to elastic rods [BAV*10]. This one-dimensional model efficiently captures extremely slender materials that are impossible to achieve with full volumetric models. On the other hand, it cannot realistically handle the bulk viscous flows supported by grid-based methods [BB08]. We therefore pursue adaptive tetrahedral methods in hopes of better bridging the gap between the extremes of small details and large volumes.

3. Newtonian Viscous Liquids

Our simulation framework builds on the method of Batty et al. [BXH10], which uses an adaptive BCC lattice tetrahedral mesh to solve the incompressible Euler equations. This scheme uses an embedded boundary approach to handle non-conforming geometry within the pressure solve, a semi-Lagrangian method for advection, and a marker level set method for surface tracking [MMS07]. (Note that while we use this framework for the underlying simulator, our discretization is directly extensible to any unstructured mesh-based method and choice of surface tracker.) We extend this

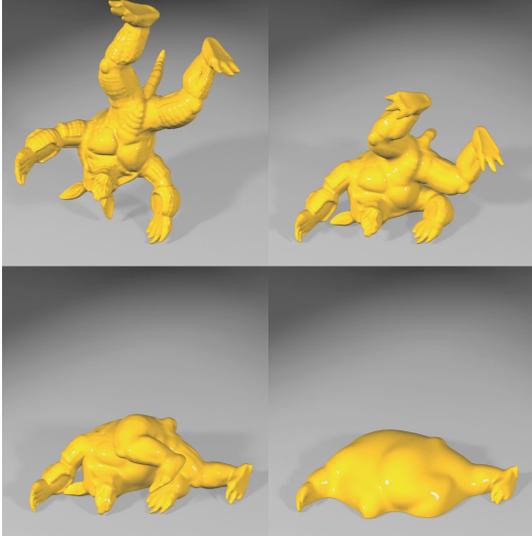


Figure 2: A viscous liquid armadillo is dropped on its head. Adaptivity enables efficient simulation of both the volume of the body and details such as the tail and claws.

framework with a viscosity model to arrive at the Navier-Stokes equations. To do so, the viscous step described below is added immediately prior to the pressure projection step, with an optional additional pressure projection applied beforehand [LSSF06, BB08].

The partial differential equations describing the evolution of velocity due to viscous forces are:

$$\rho \frac{\partial \vec{u}}{\partial t} = \nabla \cdot \tau \quad (1)$$

$$\tau = \mu(\nabla \vec{u} + (\nabla \vec{u})^T) \quad (2)$$

where \vec{u} is velocity, ρ is density, μ is viscosity, τ is deviatoric stress, and t is time. Viscosity and density are allowed to vary spatially. The free surface condition is zero traction \vec{t} on the surface:

$$\vec{t} = \tau \vec{n} = \mu(\nabla \vec{u} + (\nabla \vec{u})^T) \vec{n} = 0 \quad (3)$$

where \vec{n} is the surface normal. The no-slip solid condition requires that the relative velocity be zero along solid boundaries:

$$\vec{u} = \vec{u}_{solid} \quad (4)$$

4. Finite Volume Discretization

In approaching this problem, our first thought was to extend the finite difference method of Batty et al. [BB08] from regular grids to unstructured meshes. However, on a grid staggered face velocity components are neatly aligned so that velocity gradients (which define the components of strain rate) can be easily computed with centered finite differences. This

is not generally true for a staggered tetrahedral mesh: nearby face normals often point in unrelated directions making basic finite differences impossible.

Instead, we begin by discretizing the viscosity equations in a node-based finite volume manner, adhering to the geometric approach of Teran et al. [TBFH03] for deformable elastic objects. A full velocity vector is stored at each mesh node and a linear velocity profile is assumed within each element, leading to a constant strain rate per element.

We first outline how to compute the velocity gradient, focusing on a triangular element in 2D for simplicity. We compute the relative edge velocities $\vec{d}_{v_1} = \vec{v}_1 - \vec{v}_0$ and $\vec{d}_{v_2} = \vec{v}_2 - \vec{v}_0$, where v_i refer to node velocities of the simplex in question, and construct a 2×2 matrix D_v with columns \vec{d}_{v_1} and \vec{d}_{v_2} . Similarly we construct a 2×2 matrix D_x from columns $\vec{d}_{x_1} = \vec{X}_1 - \vec{X}_0$ and $\vec{d}_{x_2} = \vec{X}_2 - \vec{X}_0$, where \vec{X}_i refer to node positions. From this we can compute the velocity gradient: $\vec{F} = D_v D_x^{-1}$. This extends straightforwardly to three dimensions, yielding 3×3 matrices instead.

The element's discrete strain rate tensor $\dot{\epsilon}$ can then be computed as the symmetric part of the velocity gradient:

$$\dot{\epsilon} = \frac{1}{2}(\vec{F} + \vec{F}^T) = \frac{1}{2}(D_v D_x^{-1} + D_x^{-T} D_v^T) \quad (5)$$

In Newtonian fluids, deviatoric stress τ is linearly related to strain rate $\dot{\epsilon}$ through the viscosity coefficient μ :

$$\tau = 2\mu\dot{\epsilon} \quad (6)$$

This gives a discrete approximation of (2). To approximate (1) and thereby compute viscous forces on the mesh nodes, it remains only to discretize the divergence operator applied to the per-element stresses. In three dimensions, Teran et al. describe how this leads to force contributions to each node of a tetrahedron with the form:

$$\vec{f}_i = \frac{\tau}{3}(A_1 N_1 + A_2 N_2 + A_3 N_3) \quad (7)$$

where $A_j N_j$ are the area weighted normals of the faces of the tetrahedron bordering node i . A large sparse viscosity matrix D can then be assembled in the standard way from the contributions of all the elements.

An implicit (backward Euler) formulation of this problem has the form:

$$M_V^{new} = M_V^{old} + \Delta t D_V^{new} \quad (8)$$

where M is a diagonal matrix of lumped nodal masses, \vec{v} are the nodal velocities, and Δt is the timestep. This time integration scheme is unconditionally stable, allowing arbitrarily large time steps in the presence of high viscosities. Our liquid surface does not necessarily align with the tetrahedral mesh, so we take an embedded approach [WT08]. Level set values at nodes are used to determine the liquid fraction for an element (eg. as in marching tetrahedra), and the resulting mass is divided evenly amongst the incident

nodes. (We did not pursue the more careful mass distribution approach used by Wojtan and Turk.) Similar embedded ideas are common amongst Eulerian fluid schemes as well [ENGF03, BBB07, BB08, BXH10].

Boundary conditions: As with most Eulerian approaches, our simulation mesh extends beyond the liquid surface. We use our liquid surface tracker to determine which mesh tetrahedra lie outside the surface, and simply ignore them when constructing the viscosity matrix. This results in no forces being applied from outside the liquid, which is precisely the free surface condition of zero traction. Solid boundary no-slip conditions are applied as Dirichlet conditions that set vertices on or inside objects to match the object velocity. These simple first order solid and free surface conditions were effective for the scenarios we tested; if desired, enhanced boundary enforcement could likely be achieved using more elaborate cut cell or XFEM approaches.

Relation to Lagrangian Methods: In general, this is equivalent to a standard damping model for isotropic linear elastic deformable objects [OH99, TBFH03], with several convenient simplifications. First, since our Eulerian mesh is fixed in space there is no distinction between deformed and undeformed (or reference) coordinates; the deformation gradient (which Teran denotes as F) reduces to an identity matrix, so the Cauchy and Piola-Kirchhoff stresses are equivalent. Secondly, since we enforce incompressibility as a constraint via pressure projection, the damping coefficient associated with the first Lamé parameter λ in the linear elastic constitutive model (i.e., the second coefficient of viscosity) can be dropped, giving a stress of $\sigma = \tau = 2\mu\dot{\epsilon}$ as in equation 6. Finally, the Eulerian perspective implies that elements never deform, so we avoid the machinery needed to robustly handle ill-conditioning and inversion [ITF04, BHWT07]. This connection to Lagrangian schemes also illustrates another potential advantage: an existing unstructured mesh Eulerian liquid simulator can be readily retrofitted with an appropriately modified damping routine from a deformable object simulator to support realistic viscous flows.

Solving the System: By analogy to the damping matrix for solids, our viscosity matrix D will be symmetric negative definite, as noted by Teran et al. [TBFH03] and outlined in detail by Robinson-Mosher et al. [RMSF10]. We solve the resulting symmetric positive definite system using conjugate gradients, although designing a multigrid method for viscosity would be a useful enhancement.

5. Mapping Between Face and Node Velocities

The above model could be readily inserted into the recent Lagrangian nodal FEM liquid simulator of Misztal et al. [MBE*10]. However, our framework and many related approaches ([FOK05, KFCO06, ETK*07, CFL*07, MCP*09, SBH09, BBB10]) use staggered meshes in which only the normal component of velocity is stored on each face, as in

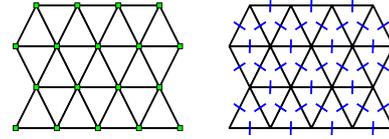


Figure 3: Different Degrees of Freedom: Left: In the nodal finite volume method, full velocity vectors are stored at mesh nodes indicated by green squares. Right: In a staggered mesh velocity representation, velocity normal components are stored on faces indicated by blue dashes.

figure 3, right. This arrangement avoids pressure instabilities and the need to explicitly treat locking [ISF07, MBE*10].

Unfortunately, a single element's face normal components turn out to be insufficient to estimate strain rate. Three normal components allow reconstruction of a single 3D velocity vector, and since a tetrahedron has four faces, the fourth normal component would ordinarily enable construction of a simple linear profile within the tetrahedron (e.g., construct one velocity vector per node from its three incident face components, and interpolate barycentrically). However, the discrete incompressibility constraint discards a degree of freedom [PVW06, KFCO06, Vid09]; any three of the four face components of a tetrahedron actually give the same velocity vector. This yields a constant velocity over the tetrahedron, with a zero gradient and thus a zero discrete strain rate.

Clearly a method based on face normal velocities will require a larger stencil to support viscosity, but choosing an appropriate stencil and discretizing the velocity gradients initially appears non-trivial. We will present two solutions based on exploiting the node-based discretization above.

5.1. An Implicit Coupling Approach

The first approach we propose is to apply the node-based discretization in combination with a translation from face degrees of freedom to nodal degrees of freedom. That is, given a method to interpolate from face velocities to node velocities, we can incorporate this translation into the linear system itself. We describe below how this can be done, extending the framework of Schroeder et al. [SZF11] to tetrahedra. (As they note, this idea is closely related to the notion of "hard bound particles" in the context of Lagrangian deformable objects [SSIF07].)

The following small linear system performs a least-squares reconstruction of a nodal vector velocity from incident faces:

$$v = (N^T N)^{-1} N^T u \quad (9)$$

The rows of matrix N are the normals of the faces surrounding a node, u is a vector containing the velocity normal components of those faces, and v is the resulting full velocity

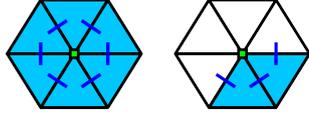


Figure 4: Reconstruction Stencil: Left: The face velocity components (blue) surrounding a node (green) are used in the least squares reconstruction of its velocity. Right: At the surface only faces bordering liquid cells (cyan) are used.

vector for the node. Figure 4, left, illustrates the face components used for a given node. Up to relabelling the vectors, this is the same problem $Nu = z$ described by Klingner et al. [KFCO06], except that we use faces incident on a node rather than the faces of a tetrahedron. Since there may be many more than three faces incident on a node, the system is over-determined; equation 9 applies the pseudo-inverse to solve it in a least-squares sense. Note that only faces bordering liquid tetrahedra are used in this reconstruction (figure 4, right). We are nevertheless guaranteed that each relevant node will have sufficiently many face components for a valid reconstruction (two in 2D, three in 3D).

After computing a matrix $(N^T N)^{-1} N^T$ for each node, we can construct a large sparse matrix H that takes the vector of all face normal velocities, and yields a vector of full velocities for all nodes. Its transpose, H^T , will be a conservative force distribution operator [SZF11], which transfers nodal forces onto face normal components of tetrahedra.

This lets us construct a new viscosity matrix \hat{D} that acts on face components as:

$$\hat{D} = H^T D H \quad (10)$$

In effect, velocities are reconstructed from faces to nodes by H , the node-based viscous forces are computed by D , and these forces are mapped back to mesh faces by H^T . This gives a linear system where the unknowns are the face normal components u :

$$Mu^{new} = Mu^{old} + \Delta t H^T D H u^{new} \quad (11)$$

where the M mass matrices refer to lumped masses for each face, rather than for each node. Clearly this system will also be symmetric positive-definite.

An advantage of this method is that it acts directly on the face components, and can therefore enable tight coupling to other face-based components of the fluid solver, such as the pressure projection [RMSF10]. Similarly, this mapping approach could be used to handle surface tension or solid-fluid coupling, as described for regular grids by Schroeder et al. [SZF11].

5.2. A FLIP-like Approach

While the above method has some advantages, the introduction of the H matrix leads to fairly large stencils, particularly

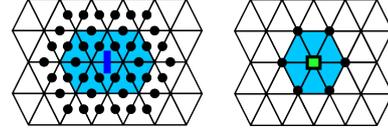


Figure 5: Sparsity: Face-based solve, left: A single face (blue segment) affects stress computations for each shaded triangle (cyan) which links it to every incident face (i.e., 41 components). Node-based solve, right: A central vertex (green) is coupled to vertices in its one-ring, (i.e., 2 components \times 7 nodes = 14 components).

in three dimensions, since the reconstruction of a nodal velocity depends on the potentially large number of incident faces (see figure 5 for a 2D example.)

Since many animation scenarios do not rely heavily on tight coupling of viscosity, we now outline a simpler strategy inspired by the popular PIC/FLIP methods. Originally presented for fluid advection [ZB05], and recently applied to collisions in hair [MSW*09], these methods use particles to represent velocity degrees of freedom, but efficiently compute the pressure forces for incompressibility on an auxiliary grid. The PIC approach simply interpolates velocities back and forth between the grid and the particles, adding significant numerical dissipation. The FLIP approach instead increments the particle velocities with just the *change* in velocities that occurred on the grid. This avoids entirely overwriting particle velocities with smoothed out grid velocities, and dramatically reduces dissipation. Interestingly, this idea of reconciling distinct velocity representations by incrementing rather than overwriting was also suggested by Guendelman et al. to avoid dissipation when switching between node- and face-based velocities on octrees [GSLF05]. (In the subsequent discussion we use the terms "FLIP" and "PIC" to describe how data is transferred between degrees of freedom; we do not use FLIP/PIC for *advection*, and emphasize that this choice is orthogonal to our method.)

To apply this idea to our problem, we effectively segregate the FVM linear solve from the two interpolation steps as follows. First reconstruct nodal velocities from face components using either the direct least-squares approach described above, or the two-step method of Klingner et al. [KFCO06]. Next, solve the original FVM formulation of Section 4 on the nodes alone. Finally, increment the tetrahedral face velocities with the normal component of the interpolated *change* in nodal velocities, in the spirit of FLIP. For highly viscous liquids the additional smoothing suffered by a PIC variant is qualitatively small, but for scenes with low or widely varying viscosity coefficients the FLIP approach is preferable. This simpler method produces results comparable to the implicit coupling approach, and we use it for all of our 3D animations.

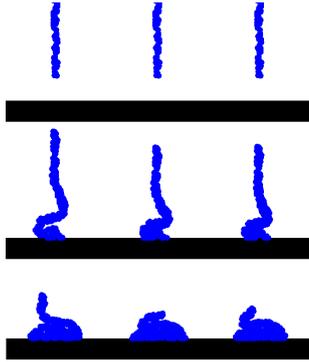


Figure 6: A 2D example comparing the different methods on a collapsing viscous liquid column. Left: Face-based (implicit coupling) solve. Centre: Node-based solve with FLIP interpolation. Right: Node-based solve with PIC interpolation. All the results are qualitatively similar, with FLIP appearing slightly less stiff due to reduced dissipation.

6. Results

We provide several examples of viscous fluid effects that cannot be captured with existing adaptive Eulerian fluid solvers.

Method Comparisons: To illustrate that the approaches outlined above produce comparable results, we ran a two-dimensional example of a column of viscous liquid collapsing. We compare the node-based scheme using FLIP interpolation against the face-based (implicitly coupled) scheme, and against the node-based scheme using PIC interpolation (figure 6). The results for all three are highly similar, though the FLIP variant exhibits slightly less damping.

Coiling Column: To test coiling, we emit a stream of viscous liquid ($\mu=150$) at a constant rate onto a flat surface. Upon impacting the ground the liquid quickly begins to bend, meander, and coil on itself (refer to the accompanying video).

Buckling Sheet: We modify the above example to emit a sheet of liquid ($\mu=100$), which buckles symmetrically back and forth (figure 1). The figure also shows a cutaway of the adaptive tetrahedral mesh.

Variable Viscosity Tori: In this example we drop three liquid tori onto the ground, with a continuous gradient on the viscosity coefficient, from $\mu = 125$ at one end of the domain to $\mu = 0$ at the other (figure 7).

Viscous Armadillo: We drop a high resolution viscous liquid Stanford Armadillo ($\mu=40$) on its head, demonstrating that our method is effective for bulk volumes with smaller details such as the tail and claws (figure 2). Realistic rotational motion is visible in the sagging of the tail and legs. Tetrahedra counts ranged between 0.9M-1.9M, while the equivalent regularly sampled domain would consist of 40M



Figure 7: Three tori collapse under gravity with decreasing viscosity coefficient from left to right.

tetrahedra. The simulation averaged 4.3 minutes/frame for the first 300 frames (during which most of the challenging dynamics occur) on a 6-core Intel Core i7 980X. The same simulation performed using the Laplacian model for viscosity fails to rotate properly and exhibits unnatural shearing. However, as noted by Batty & Bridson [BB08], the Laplacian form is more efficient, occasionally by as much as a factor of three depending on the liquid configuration. Thus it may be preferable in scenes where rotation and bending are negligible.

7. Discussion & Conclusions

We presented an extension to Eulerian unstructured tetrahedral mesh liquid simulators to support previously difficult viscous effects, including buckling, coiling, and variable viscosity. In doing so, we highlighted the connection to Lagrangian methods and described two ways to couple differing sets of degrees of freedom on tetrahedral meshes; we hope this proves useful as methods for solids and fluids increasingly converge (e.g., [GBO04, ISF07, WT08, MBE*10]). For example, it suggests an alternative approach to incompressibility in node-based FEM solids: map between node- and face-velocities, and perform the standard staggered pressure projection. A drawback of our system is the use of a structured lattice mesh and a marker level set surface, which both limit the level of detail that can be tracked. (The marker level set surface also exhibits slight temporal incoherence particularly in merging regions. This visual artifact of particle level set methods is accentuated by the slow movement of the liquid, but it is unrelated to our viscosity model.) A valuable direction would therefore be to integrate our method with the framework of Brochu et al. [BBB10] to animate substantially thinner viscous sheets and threads.

References

- [BAV*10] BERGOU M., AUDOLY B., VOUGA E., WARDETZKY M., GRINSPUN E.: Discrete viscous threads. *ACM Trans. Graph. (SIGGRAPH)* 29, 4 (2010), 116. 2
- [BB08] BATTY C., BRIDSON R.: Accurate viscous free surfaces for buckling, coiling, and rotating liquids. In *Symposium on Computer Animation* (2008), pp. 219–228. 1, 2, 3, 4, 6
- [BBB07] BATTY C., BERTAILS F., BRIDSON R.: A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph. (SIGGRAPH)* 26, 3 (2007), 100. 4
- [BBB10] BROCHU T., BATTY C., BRIDSON R.: Matching fluid

- simulation elements to surface geometry and topology. *ACM Trans. Graph. (SIGGRAPH)* 29, 3 (2010). 1, 4, 6
- [BHW07] BARGTEIL A. W., HODGINS J. K., WOJTAN C., TURK G.: A finite element method for animating large viscoplastic flow. *ACM Trans. Graph. (SIGGRAPH)* 26, 3 (2007), 16. 2, 4
- [BPL06] BONITO A., PICASSO M., LASO M.: Numerical simulation of 3D viscoelastic flows with free surfaces. *J. Comp. Phys.* 215, 2 (2006), 691–716. 2
- [BXH10] BATTY C., XENOS S., HOUSTON B.: Tetrahedral embedded boundary methods for accurate and flexible adaptive fluids. *Computer Graphics Forum (Eurographics)* 29, 2 (2010), 695–704. 1, 2, 4
- [CFL*07] CHENTANEZ N., FELDMAN B. E., LABELLE F., O'BRIEN J. F., SHEWCHUK J. R.: Liquid simulation on lattice-based tetrahedral meshes. In *Symposium on Computer Animation* (2007), pp. 219–228. 1, 4
- [CMVT02] CARLSON M., MUCHA P. J., VAN HORN R., TURK G.: Melting and flowing. In *Symposium on Computer Animation* (2002), pp. 167–174. 2
- [ENGF03] ENRIGHT D., NGUYEN D., GIBOU F., FEDKIW R.: Using the particle level set method and a second order accurate pressure boundary condition for free surface flows. In *Proceedings of the 4th ASME-JSME Joint Fluids Engineering Conference* (2003), pp. 337–342. 4
- [ETK*07] ELCOTT S., TONG Y., KANSO E., SCHRÖDER P., DESBRUN M.: Stable, circulation-preserving, simplicial fluids. *ACM Trans. Graph.* 26, 1 (2007), 4. 1, 2, 4
- [FM96] FOSTER N., METAXAS D.: Realistic animation of liquids. *Graphical Models and Image Processing* 58, 5 (1996), 471–483. 1
- [FOK05] FELDMAN B. E., O'BRIEN J. F., KLINGNER B. M.: Animating gases with hybrid meshes. *ACM Trans. Graph. (SIGGRAPH)* 24, 3 (July 2005), 904–909. 4
- [FR03] FÄLT H., ROBLE D.: Fluids with extreme viscosity. In *SIGGRAPH Sketches* (2003), p. 1. 2
- [GBO04] GOKTEKIN T. G., BARGTEIL A. W., O'BRIEN J. F.: A method for animating viscoelastic fluids. *ACM Trans. Graph. (SIGGRAPH)* 23, 3 (Aug. 2004), 463–468. 2, 6
- [GSLF05] GUENDELMAN E., SELLE A., LOSASSO F., FEDKIW R.: Coupling water and smoke to thin deformable and rigid shells. *ACM Trans. Graph. (SIGGRAPH)* 24, 3 (2005), 973–981. 5
- [HK05] HONG J.-M., KIM C.-H.: Discontinuous fluids. *ACM Trans. Graph. (SIGGRAPH)* 24, 3 (July 2005), 915–920. 1
- [HW65] HARLOW F. H., WELCH J. E.: Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids* 8, 12 (1965), 2182–2189. 1
- [ISF07] IRVING G., SCHROEDER C., FEDKIW R.: Volume conserving finite element simulations of deformable models. *ACM Trans. Graph. (SIGGRAPH)* 26, 3 (2007), 13. 2, 4, 6
- [ITF04] IRVING G., TERAN J., FEDKIW R.: Invertible finite elements for robust simulation of large deformation. In *Symposium on Computer Animation* (2004), ACM Press, pp. 131–140. 2, 4
- [KFCO06] KLINGNER B. M., FELDMAN B. E., CHENTANEZ N., O'BRIEN J. F.: Fluid animation with dynamic meshes. *ACM Trans. Graph. (SIGGRAPH)* 25, 3 (2006), 820–825. 4, 5
- [LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. *ACM Trans. Graph. (SIGGRAPH)* 23, 3 (Aug. 2004), 457–462. 1
- [LSSF06] LOSASSO F., SHINAR T., SELLE A., FEDKIW R.: Multiple interacting liquids. *ACM Trans. Graph. (SIGGRAPH)* 25, 3 (2006), 812–819. 3
- [MBE*10] MISZTAL M., BRIDSON R., ERLEBEN K., BAERENTZEN A., ANTON F.: Optimization-based fluid simulation on unstructured meshes. In *VRIPHYS* (2010). 4, 6
- [MCP*09] MULLEN P., CRANE K., PAVLOV D., TONG Y., DESBRUN M.: Energy-preserving integrators for fluid animation. *ACM Trans. Graph. (SIGGRAPH)* 28, 3 (2009), 38. 1, 2, 4
- [MMS07] MIHALEF V., METAXAS D., SUSSMAN M.: Textured Liquids Based on the Marker Level Set. *Computer Graphics Forum (Eurographics)* 26, 3 (2007), 457–466. 2
- [MSW*09] MCADAMS A., SELLE A., WARD K., SIFAKIS E., TERAN J.: Detail-Preserving Continuum Simulation of Straight Hair. *ACM Trans. Graph. (SIGGRAPH)* 28, 3 (2009), 62. 5
- [OBH02] O'BRIEN J. F., BARGTEIL A. W., HODGINS J. K.: Graphical modeling and animation of ductile fracture. *ACM Trans. Graph. (SIGGRAPH)* 21, 3 (2002), 291–294. 2
- [OH99] O'BRIEN J. F., HODGINS J. K.: Graphical modeling and animation of brittle fracture. In *SIGGRAPH* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 137–146. 2, 4
- [PVW06] PEROT B., VIDOVIC D., WESSELING P.: Mimetic reconstruction of vectors. In *Compatible Spatial Discretizations*, Arnold D. N., Bochev P. B., Lehoucq R. B., Nicolaides R. A., Shashkov M., (Eds.). Springer, 2006, pp. 173–188. 4
- [REN*04] RASMUSSEN N., ENRIGHT D., NGUYEN D., MARINO S., SUMNER N., GEIGER W., HOON S., FEDKIW R.: Directable photorealistic liquids. In *Symposium on Computer Animation* (2004), pp. 193–202. 1, 2
- [RMSF10] ROBINSON-MOSHER A., SCHROEDER C., FEDKIW R.: A symmetric positive definite formulation for monolithic fluid structure interaction. In *submission* (2010). 4, 5
- [SBH09] SIN F., BARGTEIL A. W., HODGINS J. K.: A point-based method for animating incompressible flow. In *Symposium on Computer Animation* (2009), pp. 247–255. 1, 4
- [SSIF07] SIFAKIS E., SHINAR T., IRVING G., FEDKIW R.: Hybrid Simulation of Deformable Solids. In *Symposium on Computer Animation* (2007). 4
- [Sta99] STAM J.: Stable fluids. In *SIGGRAPH* (1999), pp. 121–128. 2
- [SZF11] SCHROEDER C., ZHENG W., FEDKIW R.: Implicit surface tension formulation with a Lagrangian surface mesh on an Eulerian simulation grid. In *submission* (2011). 4, 5
- [TBFH03] TERAN J., BLEMKER S. S., FEDKIW R., HING V. N. T.: Finite volume methods for the simulation of skeletal muscle. In *Symposium on Computer Animation* (2003), Eurographics Association, pp. 68–74. 2, 3, 4
- [Vid09] VIDOVIC D.: Polynomial reconstruction of staggered unstructured vector fields. *Theoret. Appl. Mech.* 26, 2 (2009), 85–99. 4
- [WBOL07] WENDT J. D., BAXTER W., OGUZ I., LIN M. C.: Finite volume flow simulations on arbitrary domains. *Graphical Models* 69, 1 (2007), 19–32. 2
- [WT08] WOJTAN C., TURK G.: Fast viscoelastic behavior with thin features. *ACM Trans. Graph. (SIGGRAPH)* 27, 3 (2008), 47. 2, 3, 6
- [ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. *ACM Trans. Graph. (SIGGRAPH)* 24, 3 (July 2005), 965–972. 5