Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2008)
M. Gross and D. James (Editors)

Accurate Viscous Free Surfaces for Buckling, Coiling, and Rotating Liquids

Christopher Batty Robert Bridson

Department of Computer Science, University of British Columbia

Abstract

We present a fully implicit Eulerian technique for simulating free surface viscous liquids which eliminates artifacts in previous approaches, efficiently supports variable viscosity, and allows the simulation of more compelling viscous behaviour than previously achieved in graphics. Our method exploits a variational principle which automatically enforces the complex boundary condition on the shear stress at the free surface, while giving rise to a simple discretization with a symmetric positive definite linear system. We demonstrate examples of our technique capturing realistic buckling, folding and coiling behavior. In addition, we explain how to handle domains whose boundary comprises both ghost fluid Dirichlet and variational Neumann parts, allowing correct behaviour at free surfaces and solid walls for both our viscous solve and the variational pressure projection of Batty et al. [BBB07].

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Physically based modeling

1. Introduction

Viscous liquids are a common feature of the world around us. Household examples include honey, syrup, paints, cake batter, and molasses; the unique behaviour exhibited by these liquids is therefore extremely familiar to most of us. Film and games often make use of increasingly exotic examples including wet mud, tar, lava, quicksand, or goo. The distinguishing characteristic of these liquids is their resistance to shearing flow, resulting in extremely slow, damped motion that, in the interior of the fluid, is not terribly compelling to watch. However, at the interface between air and liquid a host of complex and distinctive effects can arise. When viscous fluid is poured onto a surface it will often begin to coil or fold over upon itself, generating intricate surface details. The unwieldy technical names for such phenomena are cylindrical and planar viscous jet buckling, respectively; however, they can readily be understood by considering that liquid will prefer the path of least resistance. The falling fluid above and the viscous pile below apply opposing forces, but the surrounding air applies little to no resistance, causing the fluid to bend or bow out to one side. This and many more subtle behaviours are generated by the delicate coupling of air and liquid, and the resulting motion may provide important visual cues to a fluid's material properties. A recent example comes from the makers of *Bee Movie* [Rui07], who met with difficulties attempting to model honey with standard viscous fluid simulators. Although they resorted to a (non-physical) viscoelastic model, we postulate that the true root of the problem lies not in the constitutive law, but in the free surface boundary conditions. We present a new method that enforces these conditions easily and accurately for the first time, using a novel fully implicit time integration scheme. This new method allows for the efficient simulation

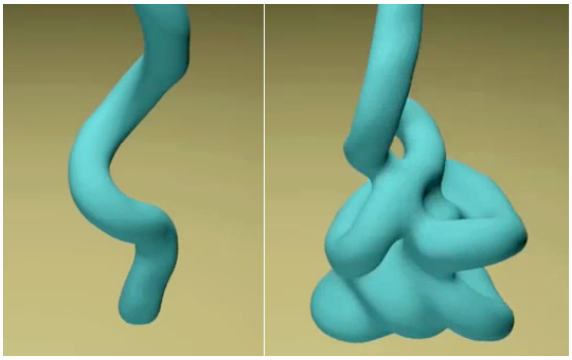

Figure 1: *An initially straight stream of viscous fluid buckles and coils as it falls.*

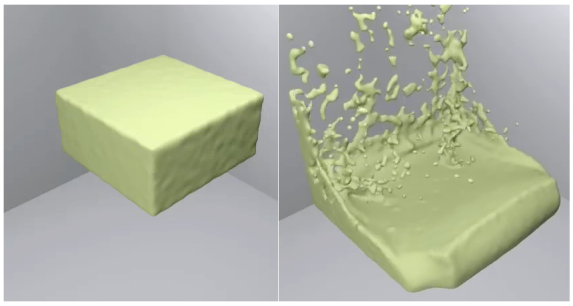

Figure 2: *A block of fluid whose viscosity varies smoothly along its length is dropped onto a flat plane; the far end splashes in an inviscid manner, while the near end deforms only slightly.*

of a variety of complex viscous liquid phenomena that were previously extremely difficult or impossible to reproduce.

1.1. Contributions

We now summarize our primary contributions. First, we point out that in order to achieve convincing viscous behaviour it is in fact vital to enforce the traction-free boundary conditions on the liquid free surface, which requires full coupling between the components of velocity. We then proceed to develop a fully implicit variational interpretation of the viscosity update which relates the total viscous dissipation to an energy term reflecting the change in fluid velocity. We prove its equivalence to the standard PDE form and note that since the minimization form is quadratic in velocity, the problem is automatically well-posed and its discretization is symmetric semi-definite, allowing efficient solution using conjugate gradient. Furthermore, it leads to a simple volume-weighting scheme on the MAC grid which implicitly enforces the difficult free surface boundary condition, greatly simplifying implementation. Finally, we illustrate how to combine this type of variational Neumann boundary condition with traditional Dirichlet boundary conditions, allowing us to handle both free surfaces and solid walls. This is useful for our viscous solve as well as the variational pressure projection introduced by Batty et al [BBB07]. We provide examples illustrating that this method is unconditionally stable, eliminates artifacts in rotation and bending, conserves angular momentum, supports variable viscosity without modification, and provides more accurate modeling of free surface viscous liquids than previously seen in graphics.

2. Related Work

We will focus on demonstrating that correct free surface boundary conditions are important for properly simulating viscous liquids, and will use viscous buckling and coiling as our key example. This phenomenon was first studied by physicist G. I. Taylor [Tay68], and a thorough experimental study was carried out by Cruikshank & Munson [CM81]. Bejan later penned a review article on the subject [Bej87], which also issued a rallying cry to the computational fluid mechanics community to tackle this "new frontier".

Viscous fluids were introduced to computer graphics by Miller & Pearce [MP89], who extended particle systems with inter-particle forces to approximate melting and flowing of viscous substances. Similarly, Terzopoulos et al. [TPF89] demonstrated the ability to melt finite element solids into collections of interacting particles.

The first work in computer graphics to simulate viscous fluids using the 3D Navier-Stokes equations was Foster & Metaxas [FM96], who adapted the classic MAC method of Harlow & Welch [HW65]. Though quite effective, it required small time steps due to the use of explicit integration. Stam [Sta99] introduced an implicit viscosity solve (along with semi-Lagrangian advection) which enabled much larger time steps, greatly improving simulation efficiency. By assuming constant viscosity, this method decouples the components of velocity allowing each to be solved independently. The resulting three linear systems are symmetric positive definite with a Poisson-like form and can be conveniently solved with a conjugate gradient method. We will refer to this method as the classic decoupled solve.

Carlson et al. [CMVT02] adapted this model to handle free surface liquids and variable viscosity; by further adding a heat diffusion model they generated an impressive animation of a wax bunny steadily melting due to a nearby heat source. However, their simplification of both the variable viscosity term and the free surface boundary condition introduced artifacts such as nonphysical damping of ballistic motion, which they partially rectified by directly adding back in the expected net translational motion (albeit choosing to neglect rotation). Falt & Roble [FR03] later corrected the translational error (though again, not the rotational error) by enforcing Neumann boundary conditions of the form $(\nabla \vec{u}) \cdot \vec{n} = 0$ at grid-aligned air-fluid interfaces.

Rasmussen et al. [REN*04] also studied the case of free surface variable viscosity, but rather than dropping terms they eliminated the coupling between velocity components by proposing a combined implicit-explicit (IMEX) integration scheme. Under this scheme the dimensionally coupled components are first integrated explicitly, and the remaining decoupled, symmetric components are integrated implicitly. For constant viscosity regions the explicit components exactly cancel (assuming the input velocities are incompressible) leaving behind the same three linear systems as before. This technique was used to creating a stunning melting robot sequence for the third *Terminator* film.

Hong et al. [HK05] demonstrated two-phase fluids with discontinuous jumps in viscosity across the interface between constant viscosity fluids, simplifying earlier work by Kang et al. [KFL00] and adapting it to the octree discretization of Losasso et al. [LGF04]. Losasso et al. [LSSF06] extended this approach to multiple *immiscible* liquids, but still

Figure 3: *Three different simulations of a long sheet of fluid falling under gravity demonstrating the influence of viscosity on buckling; from left to right viscosity values are 0.2, 1, and 5.*

used constant viscosity for a given fluid to avoid the time step restrictions of the IMEX integration scheme.

Several papers have examined non-Newtonian fluids, ie. fluids whose stress is non-linearly related to the strain rate, and whose behaviour lies on the continuum between fluid and solid. Zhu & Bridson [ZB05] added a simplified frictional plasticity model to a fluid simulator to animate the motion of sand. To simulate large viscoplastic flow Bargteil et al. [BWHT07] started instead from the Lagrangian finite element viewpoint, and added remeshing and basis updates to the invertible finite element method of Irving et al. [ITF04]. Wojtan & Turk subsequently extended this scheme with an embedded deformation method and an explicit surface tracker to retain thin features and speed up meshing [WT08].

Goktekin et al. introduced an explicit method for simulating *viscoelastic* liquids [GBO04], by adding an elasticity step to a fluid simulator based on an estimate of accumulated strain. They captured the complex elastic behavior of such fluids, including a small degree of buckling. However, our work differs from theirs in a few key points. First, our method is fully implicit and unconditionally stable, and properly handles rotation. Secondly, and more importantly, we demonstrate that by correctly capturing the true free surface boundary condition, we can capture the buckling of purely viscous Newtonian fluids. For example, our method can simulate honey or molasses without introducing spurious (nonphysical) elastic effects. In fact, it is complementary to their method and could be used as a drop-in replacement for their standard viscous step, which is entirely orthogonal to the elastoplastic components of the work.

There are also examples of SPH methods [CBP05], vorticity-based methods [ETK*07], and Lattice Boltzmann methods [Thu07] that support viscous fluids, though none in graphics have displayed viscous buckling. In computational physics, a few papers have successfully tackled this phenomenon including the SPH method of Rafiee et al. [RMH07] and the unstructured mesh finite element method of Bonito et al. [BPL06]. We will instead focus on Eulerian, Cartesian grid-based simulation.

In computational physics, the classic MAC scheme has been adapted to handle highly viscous (low Reynolds number) free surface fluids. A pair of papers by Hirt & Shannon [HS68] and Nichols & Hirt [NH71] looked at enforcing the full traction-free surface boundary conditions in 2D, the former examining the normal stress condition, the latter the tangential stress condition. They assume each cell is either full or empty, approximate the resulting surface normals as either grid-aligned or at 45 degrees, and derive discrete conditions for each case. Pracht used these same conditions in an implicit approach [Pra71] that solves a large linear system for pressure and velocity simultaneously.

The various incarnations of the GENSMAC method of Tomé, McKee, and co-workers [TM94, TM99, TFC*01] extended the general MAC framework to three dimensions including explicit traction-free surface boundary conditions. To our knowledge, GENSMAC is the first and only MAC-type scheme to successfully simulate viscous jet buckling. The free surface is again enforced using a case-based analysis, assuming incompressibility and a small set of possible surface normals. More recent work of de Sousa et al. [dSMN*04] used an accurate normal extracted from a surface mesh, but it is unclear how this is used in applying the boundary conditions. Noting difficulties with simulating low Reynolds flow, Oishi et al. [OCF*06] adapted GENSMAC to an implicit solve in 2D, but with decoupled pressure and velocity (in contrast to Pracht's work). They present results showing that to achieve reasonable time step sizes, it is necessary to solve both the equations of motion *and* the boundary conditions implicitly. They have since extended this method to 3D [OTCM08], enabling the simulation of 3D coiling for quite viscous fluids. However, this technique requires the solution of a large asymmetric linear system as

well as unwieldy derivation and implementation of 26 cases of discrete surface orientation arising in 3D.

There are also techniques that more accurately enforce the boundary conditions in an explicit manner. These approaches perform a least-squares estimate of the velocity gradient near the surface using several sample points and an SVD operation, and then apply an extrapolation while enforcing the boundary conditions [PZ02, HP04]. This contrasts with the simple constant extrapolation prevalent in computer graphics (eg. [EMF02]).

3. Variable Viscosity Flow

We wish to simulate highly viscous incompressible fluids, possibly with varying viscosity. In this setting, the Navier Stokes equations have the following form:

$$\vec{u}_t = -\vec{u}\cdot\nabla\vec{u} + \tfrac{1}{\rho}\nabla\cdot\tau - \tfrac{1}{\rho}\nabla p + \vec{g} \tag{1}$$

$$\nabla\cdot\vec{u} = 0 \tag{2}$$

$$\tau = \mu(\nabla u + (\nabla u)^T) \tag{3}$$

where, $\vec{u}$ is velocity, μ is dynamic viscosity, p is pressure, ρ is density, $\vec{g}$ is external accelerations (eg. gravity), and τ is the viscous shear stress tensor. We take the standard approach in graphics of using operator splitting to solve for viscous forces independently. In a given timestep we first apply advection and external forces, project the velocities to be divergence free, solve for viscosity, and finally project the velocities to be divergence free a second time (see eg. [LSSF06]). (Two pressure projections are needed because operator-split viscosity formulations typically assume an incompressible velocity field.) This leaves us with the following PDE for integrating viscosity alone:

$$\vec{u}_t = \frac{1}{\rho}\nabla\cdot\left(\mu(\nabla\vec{u} + (\nabla\vec{u})^T)\right) \tag{4}$$

Previous approaches discretized this PDE form directly, using explicit, IMEX [REN*04], or implicit schemes, giving the following:

$$\vec{u} = \vec{u}^{old} + \frac{\Delta t}{\rho}\nabla\cdot\left(\mu(\nabla\vec{u^*} + (\nabla\vec{u^\dagger})^T)\right) \tag{5}$$

For the sake of brevity, we use $\vec{u}$ to refer to the updated velocity, while $\vec{u}^{old}$ refers to the input velocity. To define a particular integration scheme, $\vec{u}^*$ and $\vec{u}^\dagger$ are chosen to be either $\vec{u}^{old}$ or $\vec{u}$. A fully explicit scheme sets $\vec{u}^* = \vec{u}^\dagger = \vec{u}^{old}$, a fully implicit scheme sets $\vec{u}^* = \vec{u}^\dagger = \vec{u}$, and the IMEX scheme can be arrived at by setting $\vec{u}^* = \vec{u}$ and $\vec{u}^\dagger = \vec{u}^{old}$. (For constant viscosity $\nabla\cdot(\nabla\vec{u})^T = 0$ due to incompressibility, decoupling the components of velocity and leaving the Poisson-like form usually given.)

The explicit scheme tends to require a small time step for stability; one can employ sub-cycling, taking many viscous sub-steps per overall time step, but for moderately viscous fluids this quickly becomes untenable. Rasmussen et al. partially addressed this with the IMEX scheme, whose implicit part somewhat lessens the time step restriction. It also decouples the three velocity components in the implicit part, giving the usual three systems of the classic decoupled solve. However, their primary reason for choosing an IMEX scheme over a fully implicit one which would eliminate the time step restriction entirely was that for their finite differencing method the implicit scheme generates an asymmetric linear system. Such a system cannot be solved with the usual conjugate gradient method, requiring instead a more expensive and potentially less robust solver such as GMRES. We will show in section 5 that we actually can solve this problem efficiently in a fully implicit way, by exploiting a variational principle that guarantees symmetry.

4. Viscous Free Surface Boundary Conditions

Neglecting the effects of surface tension, the true free surface boundary conditions for Navier-Stokes dictate that there is zero traction $\vec{t}$ applied on the plane of the surface. From the definition of Cauchy stress, this gives us:

$$\vec{t} = \sigma\vec{n} = 0 \tag{6}$$

where σ is the full Cauchy stress tensor and $\vec{n}$ is the normal to the free surface. Splitting σ into components of pressure p and shear stress τ, we have:

$$\sigma\vec{n} = (-p\mathbf{I} + \tau)\vec{n} = 0 \tag{7}$$

Since we have decoupled the velocity and pressure solves in our method, we do the same with the boundary conditions. If we assume as usual that the free surface pressure is zero during the pressure solve, we're left with the boundary condition $\tau\vec{n} = 0$. This implies:

$$\mu(\nabla u + (\nabla u)^T)\vec{n} = 0 \tag{8}$$

A key point to note is that this couples together the components of velocity *even for constant viscosity* [LIRO07]. To correctly enforce it we must solve the full system (4) even if decoupling occurs on the interior of the fluid.

Methods for enforcing this constraint fall into two categories: explicit and implicit. The explicit approach first extrapolates the current surface velocities into the nearby air region, possibly subject to the zero-traction constraint. During the viscous solve these air velocities are held fixed as Dirichlet boundary conditions. In graphics, simple constant extrapolation of velocity without constraint enforcement is typical. The complexity of this approach can increase almost arbitarily depending on the desired spatial accuracy for both the extrapolation and the zero-traction constraint. However, because it uses the input velocities as the starting point, its temporal accuracy and stability are ultimately still limited even in an otherwise fully implicit solve [OCF*06]. In practice, this means that if the viscosity would otherwise dictate a large change in surface velocity, it cannot occur because the old extrapolated boundary velocities remain unchanged.

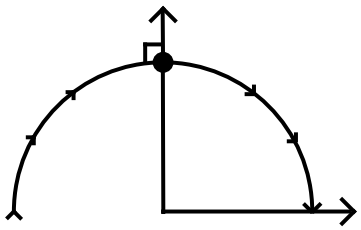

Figure 4: *A rotational velocity field $\vec{u} = (y, -x)$. The zero-traction boundary condition must be correctly enforced in order to preserve angular momentum.*

In contrast, the implicit approach uses a Neumann boundary condition, so that boundary velocities need not be known in advance. The key difficulty encountered with this approach is that the full complexity of the extrapolate/constrain process above must effectively be built into the linear system, greatly increasing implementation complexity.

Considered naïvely, either of these approaches requires estimating the normal direction, and determining exactly where and how to discretize the constraint onto the simulation grid. However, this boundary condition is in a sense a "natural" or homogeneous Neumann boundary condition, and finite element methods commonly exploit this property to circumvent the need to enforce such conditions explicitly. For example, Bonito et al. used this idea in their finite element simulations of viscous buckling [BPL06]. Our variational interpretation accomplishes the same goal within the finite difference scheme, with an approach closely related to that of Batty et al. [BBB07].

Before presenting the details, we emphasize that correctly enforcing this boundary condition is not merely an esoteric exercise, but crucial in animating the most attractive aspects of viscous flow. A common and seemingly reasonable boundary condition one might apply to the classic decoupled solve has the form $(\nabla\vec{u})\cdot\vec{n} = 0$. This is the Neumann boundary condition used by Falt & Roble (assuming grid-aligned surfaces), and also corresponds to the constant extrapolation of velocity used by Enright et al. [EMF02]. However, consider the simple 2D rigid rotational velocity field defined by $\vec{u} = (y, -x)$, as seen in Figure 4. At a location on the positive y-axis with surface normal $\vec{n} = (0, 1)$, equation (8) becomes $\frac{\partial u}{\partial y} = 0$ and $\frac{\partial v}{\partial y} = 0$. The true gradients of this rotational field are $\frac{\partial v}{\partial y} = 0$ and, crucially, $\frac{\partial u}{\partial y} = 1$. We see that the incorrect boundary condition directly works to halt rotational motion, and for moderately viscous fluids the effect is that angular velocity is rapidly damped out. Our technique will correctly give $\frac{\partial v}{\partial y} = 0$ and $\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = 1$, eliminating this artifact.

5. A Variational Interpretation of Viscosity

We now consider how to phrase an implicit viscosity step in terms of a minimization problem. One characterization of the true solution to a Stokes viscous flow problem (i.e. ignoring advection) is that it is the unique velocity field which minimizes the rate of viscous dissipation, subject to the constraint of incompressibility. This result, known as the minimum dissipation theorem, is originally due to Helmholtz [Bat67]. If we express the deformation rate tensor as

$$\dot{\varepsilon} = \frac{\nabla\vec{u} + (\nabla\vec{u})^T}{2} \tag{9}$$

then the rate of viscous dissipation is given by

$$\Phi = 2\mu\dot{\varepsilon} : \dot{\varepsilon} = 2\mu\|\dot{\varepsilon}\|_F^2 \tag{10}$$

Recall that the : operator refers to tensor double contraction (analogous to a matrix dot-product) and $\|\cdot\|_F$ indicates the Frobenius norm of a matrix. Unfortunately, simply minimizing this expression fails to produce the desired effect, because we have decoupled pressure and viscosity. We no longer have a classic Stokes problem and are not strictly enforcing incompressibility during this step. Instead what we can do is try to enforce that the velocity changes as little as possible, while simultaneously seeking a velocity field that minimizes dissipation over the timestep. Putting this together we get:

$$\min_{\vec{u}} \iiint \rho \left\|\vec{u} - \vec{u}^{old}\right\|^2 + 2\Delta t \iiint \mu \left\|\frac{\nabla\vec{u} + (\nabla\vec{u})^T}{2}\right\|_F^2 \tag{11}$$

Here the volume integrals are taken over the fluid; no boundary integrals are required. Calculus of variations can be used to show that minimizing this expression gives us back exactly the time-discretized PDE form for the viscous update, even for the variable viscosity case—see appendix A for the mathematical details. The integrals are quadratic in the new velocity and obviously bounded below by zero, so the minimization is automatically well-posed, and the discretized form will be symmetric semi-definite (as well as sparse), guaranteeing that conjugate gradient can be used to solve it efficiently. However, the most beneficial result of expressing the viscosity update in this manner is that we no longer need to handle the free surface with special cases: it is captured automatically by minimization of this volume integral.

5.1. Discretization of the Variational Principle

Rather than tackling the PDE form directly, which would include the complex free surface condition (8), we will discretize the variational principle (11), and then minimize this discrete form. We store the velocity components in the MAC grid configuration, so that the first integral has fractional volume weights centred on faces, exactly as in Batty et al. [BBB07]. Similarly, the viscous dissipation integral gives rise to volume terms associated with the various components of stress, which we locate on cell-centres and edges, as done by Goktekin et al. [GBO04]. Notice that centred finite differencing of adjacent MAC velocities places stress at these locations. As a result of this configuration, the volume weights

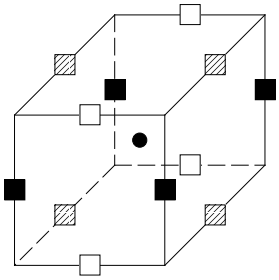

Figure 5: *The locations of stress samples on the MAC grid. $\tau_{11}, \tau_{22}, \tau_{33}$ all sit at the central black circle. τ_{23} samples are white squares, τ_{13} samples are black squares, and τ_{12} samples are the hatched squared.*

for the second integral are chosen by computing the volume fraction of fluid contained within the cube of volume Δx^3 surrounding each stress sample point. The actual method of computing these volumes is dependent on the choice of surface tracker. Estimates rather than exact volumes may be used, but the volume estimates for the different locations should be consistent. In our system we splat the union of spheres around the particles onto a grid to get an approximate signed distance field, and then estimate volumes with simple quadrature.

Discretizing and minimizing, we get a new discrete velocity update that closely mirrors the standard implicit solve:

$$u = u^{old} + \frac{\Delta t}{V_u \rho} \begin{pmatrix} (V_p 2\mu u_x)_x \\ +(V_{\tau_{12}} \mu (u_y + v_x))_y \\ +(V_{\tau_{13}} \mu (u_z + w_x))_z \end{pmatrix}$$

$$v = v^{old} + \frac{\Delta t}{V_v \rho} \begin{pmatrix} (V_{\tau_{12}} \mu (v_x + u_y))_x \\ +(V_p 2\mu v_y)_y \\ +(V_{\tau_{23}} \mu (v_z + w_y))_z \end{pmatrix}$$

$$w = w^{old} + \frac{\Delta t}{V_w \rho} \begin{pmatrix} (V_{\tau_{13}} \mu (w_x + u_z))_x \\ +(V_{\tau_{23}} \mu (w_y + v_z)_y) \\ +(V_p 2\mu w_z)_z) \end{pmatrix}$$

The V terms refer to cell-centered volumes (p, but note that $\tau_{11}, \tau_{22}, \tau_{33}$ all sit here), face-centred volumes (u, v, w), and edge-centred volumes ($\tau_{12}, \tau_{13}, \tau_{23}$). This is of course similar to the form given by Rasmussen et al. [REN*04]. The important differences are the addition of volume weights, and the use of the MAC grid so that centred differencing leaves the various discrete derivatives in the correct locations. Appendix B gives a detailed discretization for a u-velocity update in 2D for the sake of brevity, but the extension to higher dimensions is straightforward. In practice we note it is often more convenient to use dimensionless volume fractions rather than actual volumes.

5.2. Combining ghost fluid and variational boundaries

A natural question to ask is whether this type of variational Neumann boundary can co-exist with Dirichlet boundary conditions, especially of the second order accurate ghost

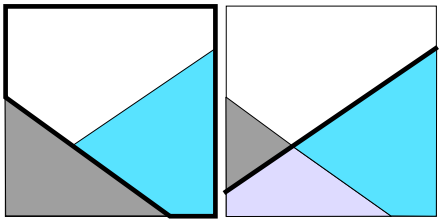

Figure 7: *Left: A liquid-air-solid triple point for the pressure projection case. Cyan indicates liquid, white indicates air, grey indicates solid wall. The combined volume used for the fluid weights is outlined by the bold line. Right: The liquid signed distance field is extrapolated into the wall for use in the ghost fluid Dirichlet boundary condition, with the ghost-fluid interface identified by the bold line.*

fluid-type employed by Enright et al. [ENGF03]. This is relevant not only to the current work, in which the air boundary is Neumann and the solid boundary is Dirichlet, but also to the work of Batty et al. [BBB07] who used a similar natural boundary condition to handle the Neumann pressure gradient constraint along non-grid-aligned solid walls. Their results were primarily restricted to examples lacking free surfaces, though they claimed that ghost fluid Dirichlet boundaries could straightforwardly be incorporated. This turns out to be the case, as we outline below. For concreteness we focus on the variational pressure problem, in which ghost fluid air-water interfaces must be handled carefully to avoid surface artifacts. The same general method is applied for boundary conditions in our viscous solve as well, by swapping air for solid in the following discussion.

The main uncertainty is whether the fluid volume estimates used in variational approaches ought to include the volume from a cell in which a Dirichlet condition is being applied. A "ghost fluid" point of view shows that the answer is yes. The ghost fluid method treats the air side of the interface as a smooth extension of the fluid domain, whose variables are chosen in such a way as to enforce the Dirichlet condition at the correct location. Therefore we assume the fluid volume is also extended smoothly into the air region, and so include its volume in our minimization (Figure 7, left). To enforce both variational solid and ghost-fluid air boundary conditions on different parts of the boundary, we simply compute the fluid volume weights by including air volume, but excluding solid volume. Then we apply the ghost fluid method on top using the liquid signed distance to determine the interface location, and ignoring the presence of weighting terms and solid walls. The discretization of equation (11) from [ENGF03] becomes (up to scaling):

$$\frac{(vol_{i+\frac{1}{2}}) \frac{p_{fs} - p_i}{\theta \Delta x} - (vol_{i-\frac{1}{2}}) \frac{p_i - p_{i-1}}{\Delta x}}{\Delta x} \tag{12}$$

The signed distance data used for determining the posi-

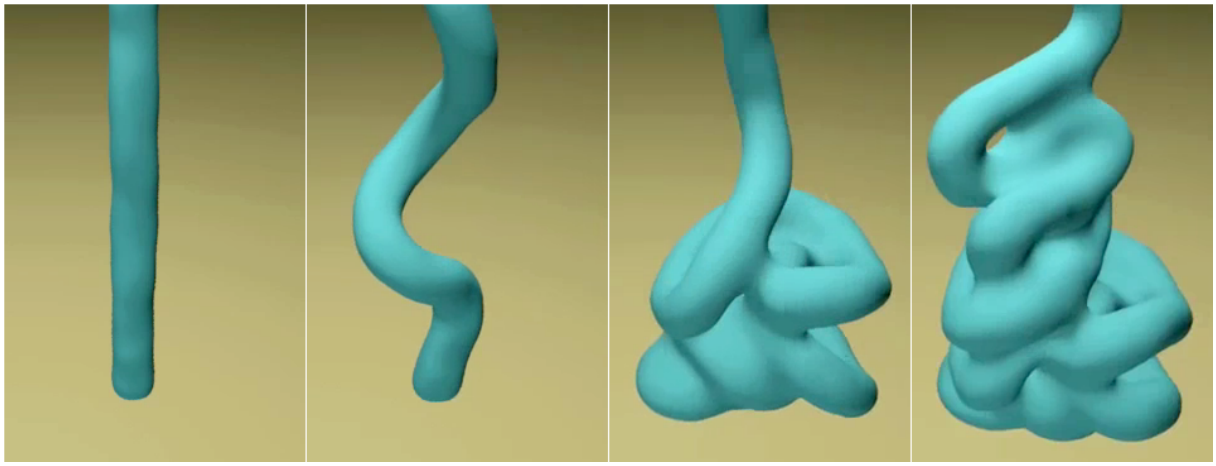

Figure 6: *A cylinder of viscous liquid falls under gravity, and spontaneously develops a coiling and folding motion.*

tion of the interface should be extrapolated smoothly into the wall, much like in the work of Rasmussen et al. [REN*04]. This ensures that the solver "sees" a smooth liquid surface right up to the (implicitly defined) solid wall, rather than one which erroneously bends away or terminates. This is illustrated in Figure 7, right.

6. Implementation

We augmented the basic FLIP approach of Bridson et al. [ZB05] with our new viscous solve. While we emphasize that our method plugs conveniently into any standard Cartesian grid-based graphics fluid simulator, an advantage of using FLIP with our method is that in combination they can seamlessly handle liquids that range continuously from almost purely inviscid to extremely viscous in a single simulation (Figure 2). We slightly reduced the memory footprint of FLIP by using one particle per cell with a larger radius, and transferring velocities from particles to the grid using a wider SPH-like kernel. The rendered surface is generated by wrapping a smoothed implicit surface around the underlying particles. Despite only minimal optimization, our examples typically required only a minute or two per frame for simulation. For example, the buckling sheet averaged one minute per frame on a 45x45x300 grid. Of that, about 50% is currently the viscosity step, which we solve with conjugate gradient and an incomplete Cholesky preconditioner. We note that while our method is inherently slower than that of Carlson et al. due to solving a unified system that is three times larger, we believe that the improved behaviour is worth this additional expense.

7. Examples

We now present a variety of examples demonstrating the validity of our approach and the range of behaviours that can be achieved. First, we illustrate the benefits of fully implicit viscosity integration. In a 2D simulation with a moderate coefficient of viscosity, we used the explicit, IMEX, and our fully implicit schemes to simulate a blob of initially motionless fluid falling under gravity. (Because these examples are 2D, for the explicit and IMEX schemes we implemented the tangential stress condition as proposed by Nichols & Hirt, setting $\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}$, either implicitly or explicitly to match the integration scheme. In 2D $\tau\vec{n} = 0$ implies $\tau = 0$, which is relatively easy to implement, but in 3D the normal becomes important, greatly increasing the complexity.) Our fully implicit approach is perfectly stable taking one large step, whereas the explicit and IMEX approaches require approximately 28 and 14 sub-steps, respectively, to avoid blow-up.

Next we examined rotational motion of a 2D circular disk of high viscosity fluid under zero gravity conditions. The Falt & Roble Neumann conditions result in rotational motion being lost instantly. Extrapolated explicit Dirichlet conditions fare slightly better, since the boundary conditions contain lagged velocities, but it still halts after a few timesteps. Our variational approach does a much better job at maintaining rotation without discernible artifacts, and rotates for hundreds of frames. (The remaining dissipation is primarily due to splitting errors related to the distinct advection and pressure phases of the simulator - advection alone partially transfers energy in rotational modes to divergent modes, which are then removed by pressure projection.)

A common test of elastic bending is a beam pinned at one end to a solid wall. We perform an analogous test on a chunk of constant viscosity (non-elastic) fluid, by applying no-slip boundary conditions at the wall (Figure 8). The implicit Neumann boundary conditions of Falt & Roble fail due to the loss of rotation at the surface. The fluid is far more damped than the viscosity would otherwise dictate, nearly halting motion altogether. Furthermore, rather than rotating, the fluid incorrectly shears and falls vertically instead of

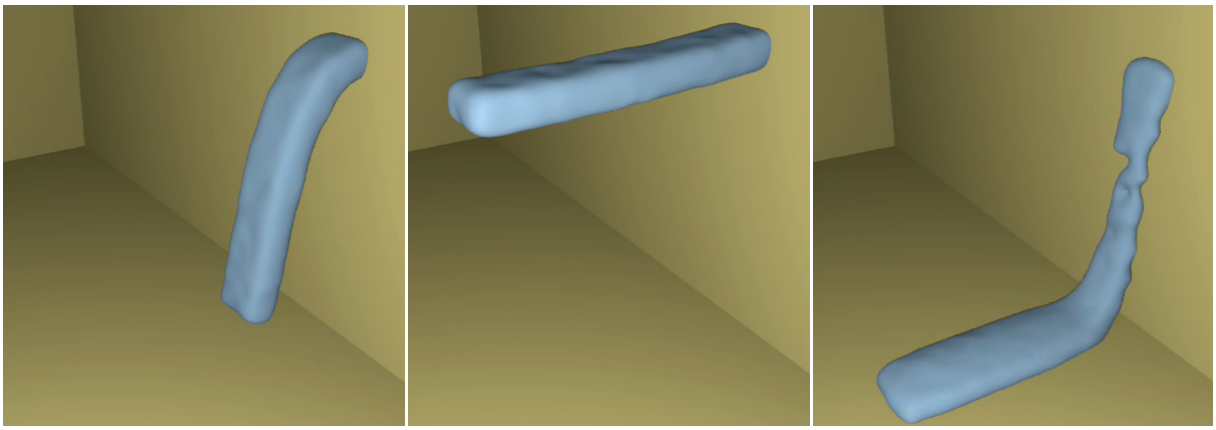

Figure 8: *A beam-shaped blob of constant viscosity fluid is attached to a wall, and simulated with three different boundary conditions. Left: Correct variational boundary conditions allow rotation, so viscous forces cause the fluid to bend in towards the wall. Middle: Incorrect Neumann boundary conditions cannot handle rotation, so the fluid can only shear and the motion is excessively damped. Right: Incorrect Dirichlet boundary conditions cannot change to reflect large changes in velocity, so the fluid falls as if unsupported.*

collapsing in towards the wall. The extrapolated Dirichlet boundary condition likewise results in large shearing. It has the additional problem that because the boundary velocities are set before the solve, they cannot change in response to the viscous forces propagating from the "pinned" end which ought to partially counterbalance gravity. The bulk of the fluid therefore falls under gravity as if it were not supported at all. Our technique results in the correct behaviour.

Next we drop a long thin cylinder of viscous fluid onto a plane (Figure 6). We successfully reproduce the strong buckling and coiling effect that is characteristic of many common purely viscous fluids, and has not been accomplished previously in graphics. To explore the effect of different coefficients of viscosity on the buckling behaviour, we drop a sheet of fluid perpendicular to the ground plane (Figure 3). For low viscosities no buckling occurs, while for higher viscosities the folds become much longer and more pronounced.

To demonstrate that we can handle variable viscosity, we drop a block of fluid whose viscosity varies continuously from one end to the other (Figure 2). Initially the block falls uniformly under gravity, illustrating that our method introduces no erroneous rotational or translational forces. Once it collides with the flat, featureless ground plane, the inviscid end collapses and splashes up against the far wall, while the viscous end sags slightly on impact. Waves and turbulent motion occurring at the inviscid end damp out as they pass towards the viscous end, so that when the simulation concludes the initial sharp edge of the fluid block is still visible.

Lastly, we illustrate that our approach to handling Dirichlet and Neumann variational boundaries together lets us easily incorporate free surfaces into the method of Batty et al. We drop a sphere of liquid inside a hollow Stanford bunny mesh, generating complex splashing and interaction with the bunny geometry (Figure 9).

8. Conclusions and Future Work

We have shown that by considering a variational principle for the viscosity solve, we can achieve complex viscous fluid effects that have been lacking in the graphics literature to date. Nonetheless, there are several avenues for future work. First, the complete free surface boundary condition couples pressure to velocity, so a unified pressure-viscosity solve is likely needed to handle this tighter coupling. Unfortunately this requires solving a larger and more complex symmetric indefinite system, and it is unclear if this would benefit graphics applications. Similarly, we did not support surface tension, although it can play a vital role in the surface behaviour. We could easily add it to the pressure solve (see eg. [ENGF03]) or use another method from the graphics literature, but a fully unified implicit approach would be interesting to consider. Finally, although the new linear system of our method is symmetric positive definite, it is no longer an M-matrix. This is the class of matrices with positive eigenvalues and non-positive off-diagonal entries, and for which the modified incomplete Cholesky preconditioner is expected to perform well. Research into alternative preconditioners could therefore further accelerate our solver.

Acknowledgements

This work was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada.

References

[Bat67] BATCHELOR G. K.: *An Introduction to Fluid Dynamics*. Cambridge University Press, 1967.

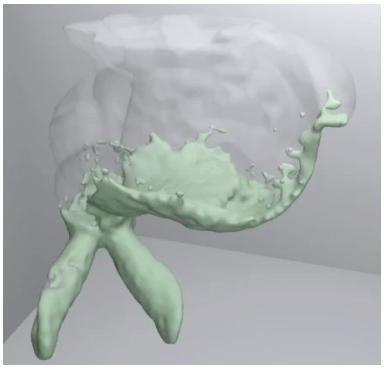

Figure 9: *A low viscosity liquid splashes inside the Stanford bunny.*

[BBB07] BATTY C., BERTAILS F., BRIDSON R.: A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph. 26*, 3 (2007), 100.

[Bej87] BEJAN A.: Buckling flows:a new frontier in fluid mechanics. *Annual Reviews of Heat Transfer 1* (1987), 262–304.

[BPL06] BONITO A., PICASSO M., LASO M.: Numerical simulation of 3d viscoelastic flows with free surfaces. *Journal of Computational Physics 215* (2006), 691–716.

[BWHT07] BARGTEIL A. W., WOJTAN C., HODGINS J. K., TURK G.: A finite element method for animating large viscoplastic flow. *ACM Trans. Graph. 26*, 3 (2007), 16.

[CBP05] CLAVET S., BEAUDOIN P., POULIN P.: Particle-based viscoelastic fluid simulation. In *Symposium on Computer Animation 2005* (2005), pp. 219–228.

[CM81] CRUIKSHANK J. O., MUNSON B. R.: Viscous-fluid buckling of plane and axisymmetric jets. *Journal of Fluid Mechanics 113* (1981), 221–239.

[CMVT02] CARLSON M., MUCHA P. J., VAN HORN R., TURK G.: Melting and flowing. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2002), pp. 167–174.

[dSMN*04] DE SOUSA F. S., MANGIAVACCHI N., NONATO L. G., CASTELO A., TOMÉ M. F., MCKEE S.: A front-tracking/front-capturing method for the simulation of 3d multi-fluid flows with free surfaces. *Journal of Computational Physics 198* (2004).

[EMF02] ENRIGHT D., MARSCHNER S., FEDKIW R.: Animation and rendering of complex water surfaces. *ACM Trans. Graph. 21*, 3 (2002), 736–744.

[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 *4th ASME JSME Joint Fluids Engineering Conference* (2003).

[ETK*07] ELCOTT S., TONG Y., KANSO E., SCHRÖDER P., DESBRUN M.: Stable, circulation-preserving, simplicial fluids. *ACM Trans. Graph. 26*, 1 (2007), 4.

[FM96] FOSTER N., METAXAS D.: Realistic animation of liquids. *Graph. Models Image Process. 58*, 5 (1996), 471–483.

[FR03] FÄLT H., ROBLE D.: Fluids with extreme viscosity. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Sketches & Applications* (2003), pp. 1–1.

[GBO04] GOKTEKIN T. G., BARGTEIL A. W., O'BRIEN J. F.: A method for animating viscoelastic fluids. *ACM Trans. Graph. 23*, 3 (2004), 463–468.

[HK05] HONG J.-M., KIM C.-H.: Discontinuous fluids. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers* (2005), pp. 915–920.

[HP04] HAO Y., PROSPERETTI A.: A numerical method for three-dimensional gas-liquid flow computations. *Journal of Computational Physics 196* (2004), 126–144.

[HS68] HIRT C. W., SHANNON J. P.: Free surface stress conditions for incompressible-flow calculations. *Journal of Computational Physics 2* (1968), 403–411.

[HW65] HARLOW F., WELCH J.: Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys Fluids 8* (1965), 2182–2189.

[ITF04] IRVING G., TERAN J., FEDKIW R.: Invertible finite elements for robust simulation of large deformation. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 131–140.

[KFL00] KANG M., FEDKIW R. P., LIU X.-D.: A boundary condition capturing method for multiphase incompressible flow. *J. Sci. Comput. 15*, 3 (2000), 323–360.

[LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (2004), pp. 457–462.

[LIRO07] LIMACHE A., IDELSOHN S., ROSSI R., ONATE E.: The violation of objectivity in laplace formulations of the navier-stokes equations. *International Journal for Numerical Methods in Fluids 54*, 6–8 (2007), 639–664.

[LSSF06] LOSASSO F., SHINAR T., SELLE A., FEDKIW R.: Multiple interacting liquids. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (2006), pp. 812–819.

[MP89] MILLER G., PEARCE A.: Globular dynamics: A connected particle system for animating viscous fluids. *Computers and Graphics 13*, 3 (1989), 305–309.

[NH71] NICHOLS B. D., HIRT C. W.: Improved free surface boundary conditions for numerical incompressible-flow calculations. *Journal of Computational Physics 8* (1971), 434–448.

[OCF*06] OISHI C. M., CUMINATO J. A., FERREIRA V. G., TOMÉ M. F., CASTELO A., MANGIAVACCHI N., MCKEE S.: A stable semi-implicit method for free surface flows. *Journal of Applied Mechanics 73* (2006), 940–947.

[OTCM08] OISHI C. M., TOMÉ M. F., CUMINATO J. A., MCKEE S.: An implicit technique for solving 3d low reynolds number moving free surface flows. *Journal of Computational Physics (in press)* (2008).

[Pra71] PRACHT W. E.: A numerical method for calculating transient creep flows. *Journal of Computational Physics 7* (1971), 46–60.

[PZ02] POPINET S., ZALESKI S.: Bubble collapse near a solid boundary: a numerical study of the influence of viscosity. *Journal of Fluid Mechanics 464* (2002), 137–163.

[REN*04] RASMUSSEN N., ENRIGHT D., NGUYEN D., MARINO S., SUMNER N., GEIGER W., HOON S., FEDKIW R.: Directable photorealistic liquids. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 193–202.

[RMH07] RAFIEE A., MANZARI M. T., HOSSEINI M.: An incompressible sph method for simulation of unsteady viscoelastic free surface flows. *International Journal of Non-Linear Mechanics 42* (2007), 1210–1223.

[Rui07] RUILOVA A.: Creating realistic cg honey. In *SIGGRAPH '07: ACM SIGGRAPH 2007 posters* (2007), p. 58.

[Sta99] STAM J.: Stable fluids. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), pp. 121–128.

[Tay68] TAYLOR G. I.: Instability of jets, threads, and sheets of viscous fluids. In *Proc. Int. Cong. Appl. Mech.* (1968).

[TFC*01] TOMÉ M., FILHO A. C., CUMINATO J. A., MANGIAVACCHI N., MCKEE S.: Gensmac3d: a numerical method for solving unsteady three-dimensional free surface flows. *International Journal for Numerical Methods in Fluids 37* (2001), 747–796.

[Thu07] THUEREY N.: *Physically Based Animation of Free Surface Flows with the Lattice Boltzmann Method*. PhD thesis, University of Erlangen-Nuremberg, 2007.

[TM94] TOMÉ M., MCKEE S.: Gensmac: A computational marker and cell method for free surface flows in general domains. *Journal of Computational Physics 110* (1994), 171–186.

[TM99] TOMÉ M., MCKEE S.: Numerical simulation of viscous flow: Buckling of planar jets. *International Journal for Numerical Methods in Fluids 29* (1999), 705–718.

[TPF89] TERZOPOULOS D., PLATT J., FLEISCHER K.: Heating and melting deformable models (from goop to glop). In *Graphics Interface 1989* (1989), pp. 219–226.

[WT08] WOJTAN C., TURK G.: Fast viscoelastic behavior with thin features. *ACM Transactions on Graphics (Proc. SIGGRAPH) 27*, 3 (2008).

[ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. *ACM Trans. Graph. 24*, 3 (2005), 965–972.

Appendix A: Equivalence of the Minimization Form

Let $\mathbf{D}$ be the rate of deformation operator defined such that $\mathbf{D}(\vec{u}) = (\nabla\vec{u} + (\nabla\vec{u})^T)/2$. Suppose $\vec{u}$ is the minimizer of (11). We introduce an arbitrary vector $\vec{q}$, a scalar ε, and a scalar function $g(\varepsilon)$ such that:

$$g(\varepsilon) = \iiint_{\text{fluid}} \rho\|\vec{u}+\varepsilon\vec{q}-\vec{u}^{old}\|^2 + 2\Delta t \iiint_{\text{fluid}} \mu\mathbf{D}(\vec{u}+\varepsilon\vec{q}) : \mathbf{D}(\vec{u}+\varepsilon\vec{q})$$

This function is quadratic in ε. Since $\vec{u}$ is the minimizer, we know that $\varepsilon = 0$ is the minimizer of g, and thus $g'(0) = 0$. Thus the coefficient of the linear terms of g must be 0, so we have:

$$0 = \iiint_{\text{fluid}} \rho\vec{q}^T(\vec{u}-\vec{u}^{old}) + 2\Delta t \iiint_{\text{fluid}} \mu\mathbf{D}(\vec{u}) : \mathbf{D}(\vec{q})$$

We now require a generalized integration by parts formula. For a symmetric rank-two tensor A and a vector q the following can easily be verified:

$$\iiint_{\omega} \mathbf{D}(\vec{q}) : \mathbf{A} = \iint_{\partial\omega} \vec{q}^T\mathbf{A}\vec{n} - \iiint_{\omega} \vec{q}^T\nabla\cdot\mathbf{A}$$

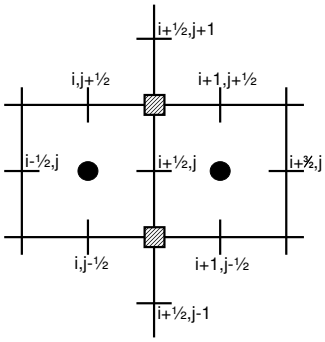

Figure 10: *The 2D stencil for the u-velocity update.*

This allows us to eliminate the $\mathbf{D}(\vec{q})$ term giving:

$$0 = \iiint_{\text{fluid}} \rho\vec{q}^T(\vec{u}-\vec{u}^{old}) - 2\Delta t \iiint_{\text{fluid}} \vec{q}^T\nabla\cdot\mu\mathbf{D}(\vec{u}) + 2\Delta t \iint_{\text{surface}} \mu\vec{q}^T\mathbf{D}(\vec{u})\vec{n}$$

Since $\vec{q}$ is arbitrary, the terms multiplying it must be zero. Hence in the fluid domain we have:

$$0 = \rho(\vec{u}-\vec{u}^{old}) - 2\Delta t\nabla\cdot\mu\mathbf{D}(\vec{u})$$

therefore

$$\vec{u} = \vec{u}^{old} + \frac{\Delta t}{\rho}\nabla\cdot\mu(\nabla\vec{u} + (\nabla\vec{u})^T)$$

which is the evolution equation for viscosity (5). On the surface of the fluid we have

$$0 = 2\Delta t\mu\mathbf{D}(\vec{u})\vec{n}$$

or equivalently

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

which is the boundary condition on stress (8). Thus minimizing this integral is equivalent to solving the PDE form.

Appendix B: Detailed 2D Discretization

The discretization of the implicit u-velocity update in 2D is:

$$u_{i+\frac{1}{2},j} = u^{old}_{i+\frac{1}{2},j} + \frac{\Delta t}{\rho V_{i+\frac{1}{2},j}}(A+B+C)$$

where

$$A = 2\left(\frac{(V_p\mu)_{i+1,j}\frac{u_{i+\frac{3}{2},j}-u_{i+\frac{1}{2},j}}{\Delta x} - (V_p\mu)_{i,j}\frac{u_{i+\frac{1}{2},j}-u_{i-\frac{1}{2},j}}{\Delta x}}{\Delta x}\right)$$

$$B = \left(\frac{(V_{\tau_{12}}\mu)_{i+\frac{1}{2},j+\frac{1}{2}}\frac{u_{i+\frac{1}{2},j+1}-u_{i+\frac{1}{2},j}}{\Delta x} - (V_{\tau_{12}}\mu)_{i+\frac{1}{2},j-\frac{1}{2}}\frac{u_{i+\frac{1}{2},j}-u_{i+\frac{1}{2},j-1}}{\Delta x}}{\Delta x}\right)$$

$$C = \left(\frac{(V_{\tau_{12}}\mu)_{i+\frac{1}{2},j+\frac{1}{2}}\frac{v_{i+1,j+\frac{1}{2}}-v_{i,j+\frac{1}{2}}}{\Delta x} - (V_{\tau_{12}}\mu)_{i+\frac{1}{2},j-\frac{1}{2}}\frac{v_{i+1,j-\frac{1}{2}}-v_{i,j-\frac{1}{2}}}{\Delta x}}{\Delta x}\right)$$

Figure 10 shows the corresponding stencil.